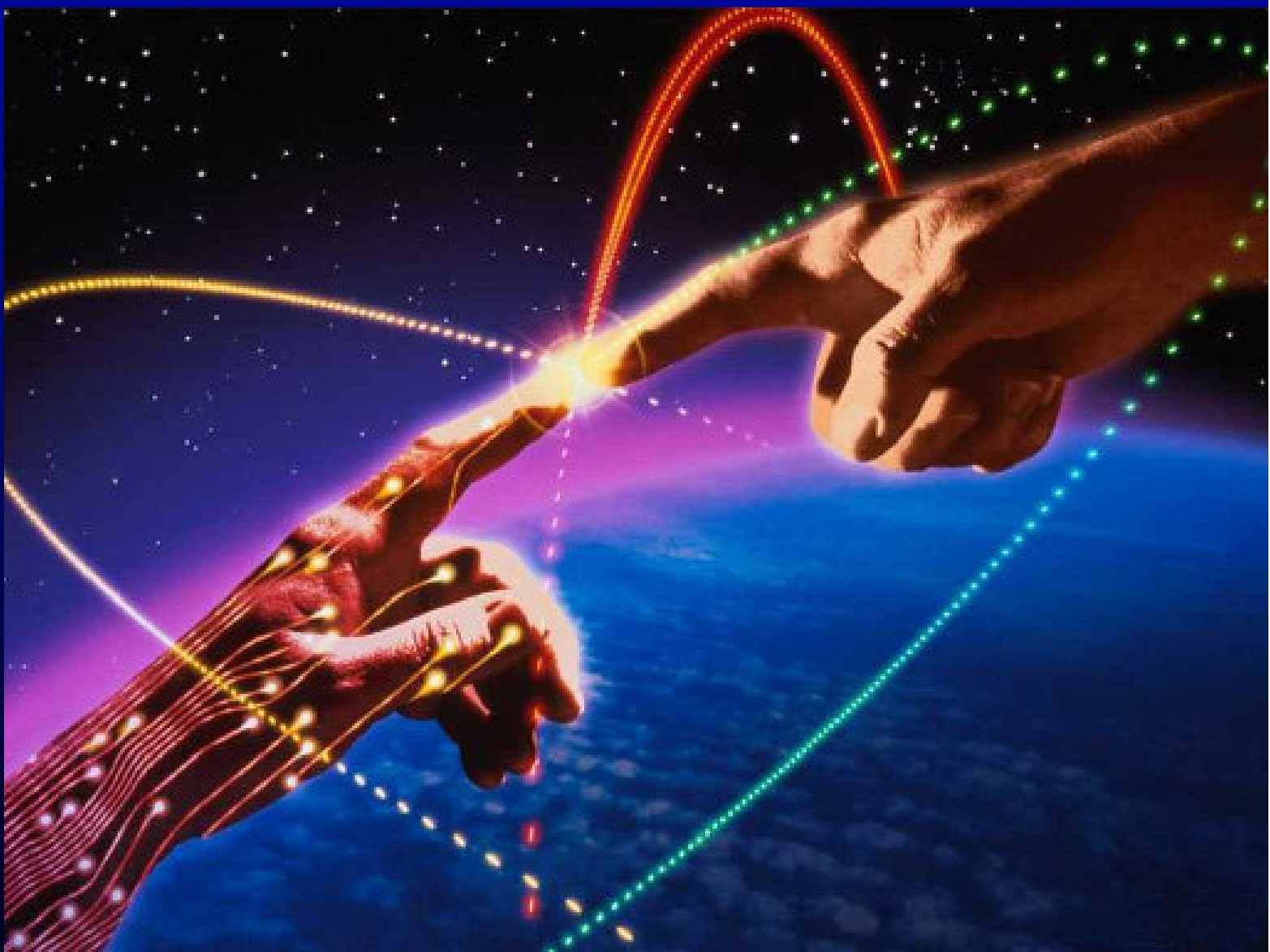


ZX Magazin

časopis pro uživatele počítačů ZX Spectrum a kompatibilních
číslo: 4/04-2/05

cena: 34 Kč





Prý jsem s úvodníkem na řadě já, +GAMA. Ani to nestíhám sledovat. Aspoň se s vámi podělím o to, co jsem při tvorbě tohoto ZXM zjistil. Word je ještě větší pytel sraček, než jsme doufali (viz skoro jakékoliv Intro od roku 2000). S vypětím všech sil jsem v něm vytvořil jakýs-takýs ZXM, jenom proto, abych ho pak musel přenášet do Adobe InDesignu. Wordovská verze byla nepoužitelná. Ztráta času. Adobe se taky pomicrosoftšťuje. Ze softwarových kouzelníků, jejichž komplexní a dobře ovladatelné DTP běhalo i na starých šrotech (černobilý Macintosh, Pentium 133) rychleji než běžné textové editory, se stala softwarová prasata, jejichž výtvar se neovládá snadno, je nechutně pomalý a vyloženě splácáný, má potíže s importem svého vlastního formátu a hroutí se (z mnoha verzí PageMakeru jsem pod OS X neuchodil jedinou - krom staříčké verze 2). Nemám teď tiskárnu na tisk náhledů, a protože na obrazovce se přeci jenom needituje tak dobře, jako na starém dobrém číňanském vynálezu, až těsně před release ZXM jsem zjistil, že jsem ho (díky neschopnosti Adobe importovat vlastní styly) vysázel zbytečně obrovským písmem. Berte to tak, že ušetříte za brýle a kontaktní čočky. Ze stejného důvodu může být vyšší výskyt chyb, na papíře se dobře kontrolují, na obrazovce se snadno přehlédnou. Zdržení ve vydávání ZXM je už půl roku (čtyřka měla vyjít na Vánoce), bohužel obsah neodpovídá časovému skluзу, při neustálém střídání zaměstnání (*výzkumník, neurochirurg, sanitář, plicař*) jsem neměl moc času na tvorbu opravdu validních článků. Jsem proto jenom rád, že se pořád najdou dobrovolníci, kteří i bez odměny časné či věčné do ZXM přispívají a obhacují jeho obsah. Díky za pomoc. Přes všechny těžkosti se ale zdá, že je tu ZX Spectrum stále s námi, že dokáže ustát i to, jak se během dospívání mění náš životní styl a množství volného času. Prostě tu je a čeká, až budeme kódovat, tvořit hudbu nebo grafiku, případně si jen tak hrát... Nebo je to jinak? Svět jde do sraček. A my s ním.

Obsah

Úvodní blábol mým národům!	2
Life is Life znáte to z úvodní kazety ke Spectru, ale znáte to opravdu?.....	3
Sinclair ZX 81, génius strýčka Cliva a zobrazování my si nedáme pokoj, ZX 81 rox!	6
Didaktik Gama a RGB výstup génius strýčka CSS a zobrazování	6
Porty a programování DivIDE užitečná tabulka.....	10
Emulátor Apple 1 na ZX Spectru ...no a co?	11
ZX Spectrum zachraňuje data z poškozeného HDD! kdy vy jste naposledy zálohovali?	16
K - mouse skutečné události, opravdoví hrdinové, tady a teď	18
Více o paralelním rozhraní i8255 vedle Z80 a ULA nejpoužívanější čip	30
Lords of time další truhla textovek.....	38
Rally driver schválně, jestli umíte závodit	38
Star raiders schválně, jestli umíte střílet	41
Meteoroids schválně, jestli umíte programovat Atari	42
FatWare co si na vás Baze přichystal	43
Kaplicon partylife rulez.....	44
Intro ...správný spectrista čte ZXM odzadu	45

ZX Magazin – časopis pro uživatele počítačů ZX Spectrum a komp

Vydavatel a šéfredaktor: Jaroslav Směták

Redakční rada: Jaroslav Směták, Martin Tobola, Jiří Doležal

Sazba: volně podle Lubomíra Bláhy

Grafická úprava: Jan Hanousek, Lubomír Bláha

Příprava obálky: volně podle Lubomíra Bláhy

Tisk předloh: Jaroslav Směták

Adresa redakce: Jaroslav Směták, Velká Lhota 111, 756 27

<http://zxm.speccy.cz>

Jakékoli reprodukce a přetisk materiálů z tohoto časopisu jsou možné pouze s písemným svolením vydavatele.

Neprochází jazykovou korekturou. Za obsah příspěvku a jeho původnost ručí autor.

Inzerce přijímá redakce. Za její obsah ručí inzerent. Cena inzerce dle dohody.

Distribuce ve formátu PDF na internetu, papírové vydání je dostupné u soukromých prodejců.

Vychází nepravidelně. Doporučená cena: 34 Kč

©2004-2005 ZX Magazin, Zbyněk Vanžura

ISSN 1210-4833

Life is Life

buněčné automaty

Milan Bok & dex

Každý spectrista asi ví, co to je, je-li řeč o hře Life. Pokud neví, tak zřejmě nedával pozor, když poprvé nahrával programy z úvodní kazety, kterou pro Sinclaira vytvořila firma Psion (její slovenská verze se objevila i u Didaktiku Gama).

Ani majitelé Didaktiků M a Kompakt nejsou omluveni, barevná varianta hry Life je jakýmsi bonusem Cookovy nezapomenutelné pařby Zolyx.

Přesto se o Life ve spectristické literatuře moc nedočtete. Nebo možná právě proto, že všichni někdy viděli vysvětlující text v basicovém prográmku na své úvodní kazetě a myslí si, že tím Life před nimi ztratila svá tajemství. Zato se o Life dozvíte ze zpravodaje Atari klubu 1,2/1993. Tak se přeci nenecháme od Ataristů zahanbit.

Hra Life, vymyšlená americkým matematikem Johnem Conwayem, je jen jedním, a poměrně jednoduchým, z velké skupiny buněčných automatů.

Teorie buněčných automatů je relativně mladá matematická disciplína a její vznik je úzce spojen právě s rozvojem počítačů.

Asi je jasné proč, a pokud ne, zkuste si na papíře o rozměru sto krát sto buněk nasimulovat pár tisíc generací!

Slovem automat se míní mechanismus zpracování informace, buňkami pak prvky určitým způsobem organizované sítě v libovolném počtu rozměrů.

Potřeba vzniku teorie buněčných automatů souvisí s výzkumem

emergentních vlastností labilních systémů, jako jsou celoplanetární počasí, ekosystémy nebo aerodynamické či hydrodynamické výpočty.

Ty se řídí sadou pravidel, velmi složitých a často ani ne přesně známých, a jejich chování vykazuje překvapivé vlastnosti, které nejsou součástí jejich definice, ale objevují se zcela nově.

Hledání emergujících vlastností v tak obtížně simulovatelných systémech si pak přímo říká o jednodušší model, na kterém mohou být principy fungování snáze odhalovány.

Jak jsme si řekli, jde o síť buněk, pro zobrazení simulace na obrazovce nejčastěji dvojrozměrné, ale může jít i o systémy třírozměrné nebo dokonce jednorozměrné, pochopitelně i vícerozměrné.

Ve dvourozměrném zobrazení mohou mít buňky tvar trojúhelníků (každá buňka pak má tři stranové sousedy) či šestiúhelníků, ovšem obě tyto sítě se při převodu na maticový tvar v podstatě podobají nejpoužívanější síti čtverečkové.

Počáteční stav systému - počáteční generace - se mění na další generace pomocí generačního kroku, který je buď paralelní, nebo sekvenční.

Máme-li řádku buněk délky D, kde některé buňky jsou plné (1) a ostatní prázdné (0), a v generačním kroku zaplníme buňku právě tehdy, když aspoň jedna z jejích sousedních buněk je plná, pak, je-li na konci i začátku řady plná buňka, stane se následující:

Při paralelním zpracování (všechny buňky souběžně), pokud byla aspoň jedna buňka plná, se řádka po N generacích zaplní.

Z čísla N lze určit, že v počáteční generaci obsahovala řádka díry (souvislé řetězce prázdných buněk) o největší délce $2 \times N$.

Při zpracování sekvenčním (postupným) se celá řádka zaplní hned v první generaci.

Složitější buněčné automaty mají pak, pochopitelně, i víc než jen dva stavy v jedné buňce. Při jejich kódování do barevných map pak vznikají obrazce, používané v netradičním oboru - počítačovém výtvarnictví. Taková je i mutace Life ve hře Zolyx.

Na Spectru o moc víc, než je vidět v Zolyxu, nedosáhneme, na to mají atributy malé rozlišení (nicméně známý demoefekt fire je taky buněčný automat).

Velká výtvarná díla vytvořená buněčnými automaty asi necháme jiným platformám.

Pro barevné mapování je třeba, aby bylo možné definovat každý pixel jednotlivě.

Jen pro zajímavost, na Commodoru bychom dosáhli maximálně pěti barev (to je maximální omezení jejich atributu).

Osmibitové Atari by zvládlo 16 různých barev se stejným jasnem (mód 11), 16 odstínů jedné barvy (mód 9) nebo 9 různě odstínovaných barev (mód 10).

Spectristická platforma, Sam Coupé,

těch 16 barev zvládne v pohodě taky, a mimochodem, Atari ST by taky víc nezvládlo, stejně jako Apple II GS. Jen Amiga 500 by ukázala 32 barev.

Conwayovo Life vyniká jednoduchostí.

Síť buněk je dvojrozměrná, stejná, jako na čtverečkovém papíře, a buňky jsou buď mrtvé (prázdné) nebo živé (obsazené). Okolí každé z nich je tvořeno osmi buňkami, sousedícími stranou nebo rohem.

Life pracuje paralelně a pravidla pro další generace, jejichž stanovení je stěžejním bodem Conwayovy práce, jsou tato:

1) Do další generace přežijí ty buňky, které mají dva nebo tři živé sousedy.

2) Umírá každá buňka, která má více než tři nebo méně než dva živé sousedy.

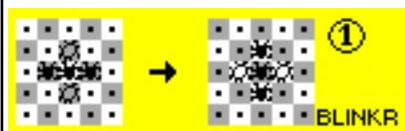
3) Pokud má mrtvá buňka právě tři živé sousedy, narodí se zde nová buňka.

To jsou všechna pravidla, vše o přežití, zániku i zrození.

Co se stane s jednotlivými útvary?

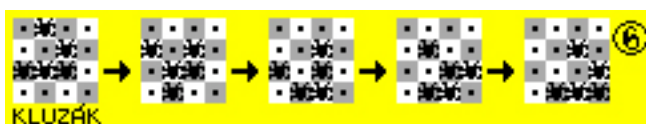
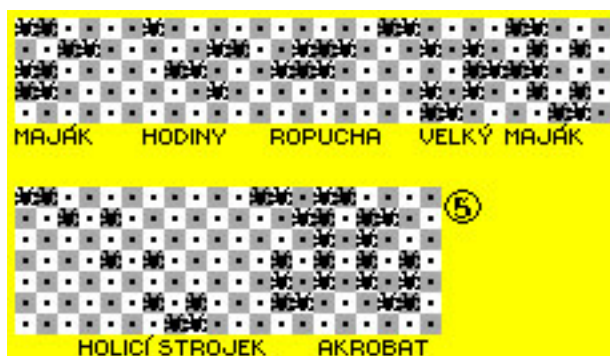
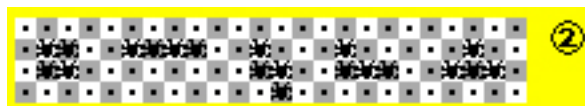
Každý obrazec o dvou buňkách během jedné generace zmizí (a pak, že ve dvou se to lépe táhne). Jediná buňka zahyne pochopitelně taky.

Z obrazců o třech buňkách je zajímavý pouze blinkr (viz obrázek 1),



tři buňky umístěné vodorovně se v další generaci změni na tři buňky umístěné svisle, a pak zase do původního tvaru. Je to zároveň nejjednodušší cyklický obrazec s nejkratší periodou.

Z obrazců o čtyřech buňkách spojených hranami (viz obrázek 2) se po několika generacích vyvinou stabilní útvary, které se už nemění (stabilní útvary najdete na obrázku 3). Z posledního útvaru se po devíti generacích stane semafor (soustava



čtyř blinkrů).

Z obrazců o více buňkách spojených hranami stojí za pozornost takzvané R-pentomino (viz obrázek 4),



jehož životopis dosud není přesně znám a je schopné vyrůst i z pole 100x100.

Mezi další zajímavé útvary patří cyklické obrazce, které po určitém počtu generací (perioda) vypadají stejně, jako na začátku. Stabilní útvary jsou vlastně cyklické obrazce o periodě 1, pak známe blinkr o periodě 2.

Další cyklické obrazce jsou maják, hodiny, ropucha, velký maják, holicí strojek a akrobat (viz obrázek 5).

Ostatní cyklické obrazce jsou už složitější, ale mohou se z jednoduchých obrazců vyvinout (viz pentadecathlon, popsáný dále).

Jedním z nejzajímavějších původních Conwayových objevů je kluzák (viz obrázek 6). Je to cyklický obrazec, který po čtyřech generacích získá zase původní tvar, ale posune se o jedno políčko doprava a dolů.

Kdybychom měli dostatečně velkou síť, je kluzák schopen odletět do libovolné vzdálenosti.

Jiné takto se pohybující útvary je

velice těžké najít. Conway je nazývá kosmické lodě a sám našel další tři. V literatuře jejich podobu ovšem nepopsal, aby si je čtenáři našli sami. Hledání kosmických lodí se tak stalo doménou dalších nadšenců.

Conway vyzkoušel i všechny vodorovné řady buněk do délky 20.

Řada tří buněk je blinkr. Z pěti buněk vznikne semafor a z řady o sedmi buňkách útvar zvaný medová farma, složený ze čtyř úlů. Nejzajímavější je řada deseti buněk, která dá vzniknout cyklickému útvaru s periodou 15, řečenému pentadecathlon.

Jedním z dosud v podstatě otevřených problémů je existence takzvané rajske zahrady.

Rajská zahrada je takové uskupení buněk, které nemůže vzniknout generačním krokem z žádného možného uskupení buněk (všimněte si, že u žádného obrazce nemůžeme jednoznačně určit tvar předchozí generace).

Americký matematik R.A. Smith dokázal, že v Life rajská zahrada existuje a alespoň jedna se nachází ve čtverci o straně deset miliard políček.

Tento poznatek ale při hledání rajske zahrady příliš nepomáhá.

V dnešní době se ale podařilo najít rajske zahrady i v mnohem menším poli. Dokonce se roku 2004 podařilo dokázat, že zmenšením sirotka,

popsaného už roku 1971, o 5 sloupců, vznikne útvar, který je zároveň sirotkem i rajskou zahradou, a to v poli 27x9. Rovněž roku 2004 byla posána rajská zahrada v poli 13x12 a 12x11. Bylo však už prokázáno, že v poli 6x5 rajská zahrada neexistuje.

Při určování pravidel pro hru Life věnoval Conway velké úsilí jejich vyvážení tak, aby bylo možné provádět sledování simulace systému dlouhodobě, tedy aby příliš mnoho obrazců nerostlo nade všechny meze nebo jich naopak příliš mnoho nemizelo.

Současná pravidla jsou stanovena na základě statistického zpracování asi dvou set tisíc různých počátečních generací buněk. Výsledkem je existence tak úžasných obrazců, jako kosmické lodě a cyklické útvary.

Conway se dokonce domníval, že žádný jednotlivý útvar nemůže nekonečně růst a nabídl padesát dolarů tomu, kdo donměnku vyvrátí nebo potvrdí.

Studiem hry se tak začala zabývat skupina hackerů, tehdy studujících na MIT. Roku 1971 získali Conwayových padesát dolarů za objev kluzákového děla.

Mimochodem, Life je dodnes oblíbenou zábavou UNIXových nadšenců a tak není divu, že simulační programy jsou většinou určeny právě pro UNIX a kluzák (glider) je symbolem hackingu.

Kluzákové dělo, které vzniká z konfigurace na obrázku 7, vystřelí každých 30 generací kluzák, který odletí pryč. První kluzák odlétá v 60. generaci.

Jelikož každý kluzák přidává k obrazci 5 buněk, opravdu roste nad všechny meze.

Skupina z MIT ale nezůstala jen u tohoto objevu.

(Jejich publikace:

Gardner M., *The fantastic combination of John Conways new solitaire game Life*, Scientific

American 1970/10,

Gardner M., *On cellular automata, self-reproduction, the Garden of Eden and the game Life*, Scientific American 1971/2)

Nalezli například způsob, jak rozestavit 13 kluzáků tak, aby se srazily a vytvořily kluzákové dělo.

Taky se jim podařilo umístit pentadecathlon tak, že požíral kluzáky, které střílelo kluzákové dělo. Pentadecathlon má periodu 15 a kluzák přilétá každých 30 generací, takže tato soustava je perfektně synchronní.

Pentadecathlon samozřejmě nedokáže eliminovat kluzák přilétající v libovolném okamžiku z libovolného směru, zatím se ale podařilo najít 24 způsobů eliminace kluzáku pentadecathlonem.

Princip spočívá v tom, že pentadecathlon v určitých fázích svého cyklu odštěpuje výběžky o dvou buňkách, které v další generaci mizí a proto je už pentadecathlon ke své další existenci nepotřebuje.

Je tedy nutné, aby přilétnuvší kluzák zasáhl výběžek v okamžiku odštěpení, a to tak, aby obrazec vzniklý z výběžku a kluzáku rychle zmizel a nepoškodil zbytek pentadecathlonu.

Poslední objev skupiny z MIT, publikovaný v únoru 1971, je způsob umístění několika kluzákových děl tak, že do sebe narážející proudy kluzáků postaví továrnu, která každých 300 generací vyrobí a vypustí jednu velkou kosmickou loď.

Na vyzkoušení těchto útvarů na Spectru už bude stará basicová simulace od Psionu horko těžko stačit především rozlišením. Tady už nezbyde, než se atributů vzdát a realizovat Life v pixelech.

Nejlepší by byl pohodlný editor i s knihovnou běžných útvarů, aby se nemusely pořád ručně kreslit.

Samotná rychlost simulace není až tak kritická, aby bylo vidět jednotlivé generace, Spectrum je výpočetně silný stroj a bude určitě stíhat i víc, než operátor stačí postřehnout.

Díky poměrně velkému grafickému rozlišení Spectra se tak otevírá velký prostor dalšímu zkoušení a kdo ví, třeba i dalším fantastickým nálezům.

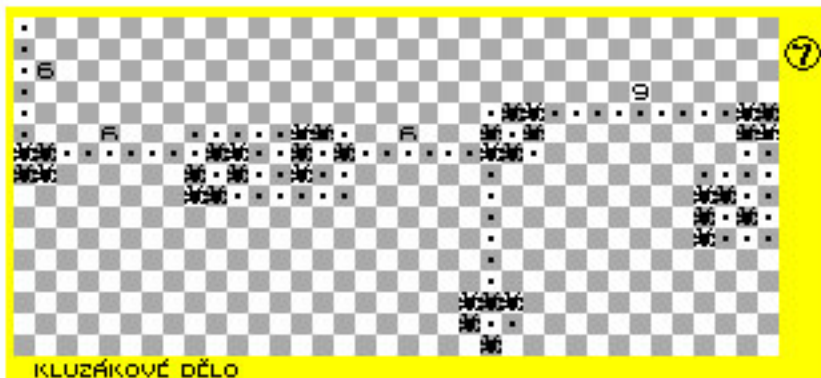
Další možností je použít UNIXový stroj (nebo fantastický program *LifeLab pro Macintosh*), toto řešení však se Spectrem může souviset pouze tehdy, bude-li ZX použito jako terminál, a zatím neznám řešení (kromě sériového propojení přes ZX Interface 1), které by něco takového jednoduchým způsobem umožňovalo.

Protože mnoho programů pro zkoumání Life používá X11 (i když existují i textové varianty), bylo by třeba pro Spectrum vytvořit X11 server.

Možná by šlo zkompileovat některý program pomocí HiSoft C.

Spectristické řešení v této oblasti na svou realizaci tedy teprve čeká.

Svět hry Life je ale rozhodně zajímavá a i v dnešní době ne zcela probádaná oblast.



Sinclair ZX 81, géníus strýčka Cliva a zobrazování

dex

Na otázku, jak je vlastně zapojeno zobrazování v počítači ZX 81, čili Total Cotel Computer, lze odpovědět velmi stručně. Prostě geniálně. Například modifikaci pro televizní systémy PAL, NTSC a SECAM získáte přemístěním pár malých součástek (dioda a odpory).

Počet snímků za sekundu (50/60 Hz) řídí pouhý jeden odpor (na nožičce 22 SCL). A takhle koncipovaný je celý stroj.

Uvnitř vlastně nic není.

Kilo RAMky (v jednom nebo dvou švábech, podle toho, jaké provedení bylo zrovna levnější). ROMka. Procesor. A pak už jenom SCL (Sinclair Computer Logic, přímý předchůdce spectráckého obvodu ULA). Jinak už v počítači není vlastně nic (modulátor, stabilizátor a bižuterie tvořená několika odpory).

Tedy maximální jednoduchost.

Nezapomínejte, jaké tehdy vládly poměry, cena osmibitových Apple II tehdy neklesala pod 1000 dolarů, Atari 800 stálo také tolik, "levný" PET od firmy Commodore přišel v Británii na 700 liber. A do toho přišlo ZX 80 za 100 liber a později ZX 81 jen za 70 liber (jako stavebnice pak ještě levněji)!

Přitom nešlo o žádnou hračku s číslicovým displejem (jako byly Commodore KIM 1 nebo Sinclairův Science of Cambridge Mk14), ale počítač s alfanumerickou klávesnicí a televizním terminálem. ZX 81 nabízelo rychlý a silný procesor s šestnáctibitovou aritmetikou, kvalitní Basic s

diskrétních součástek.

Naprostá většina obsluhy zobrazování je zajišťována softwarově podprogramy v ROMce s malou pomocí šikovně zapojeného hardwaru. Procesor vlastně většinu času jen zobrazuje, proto existuje možnost přepnout

do režimu FAST s vypnutým zobrazováním (to se automaticky dočasně zapíná při čekání na klávesu).

Adresový vodič A6 je zapojen zdánlivě nesmyslně na INT, ani RFSH není použit pro refresh paměti (ta si ho musí zajišťovat sama) a je zapojen do tvorby obrazu.

Jen při refresh

cyklu může totiž A6 přerušení vyvolat. V dolním bajtu adresové sběrnice SCL přetlačuje procesor (ten je oddělen odpory).

Mezi HALT, WAIT a NMI je zapojen tranzistor.

Výstup na televizi je společný s výstupem na magnetofon (takže při ukládání a nahrávání dat z magnetofonu se objevují v obrazovce pruhy, komunikace s kazetou je možná jen v režimu FAST). OUT (#FE), a se používá nejen pro výstup na magnetofon, ale i k tvorbě synchronizačního



plovoucí desetinnou čárkou a vědeckými funkcemi, a i v textovém módu možnost kreslení bodů a čar! Třeba graficky zdatný Commodore 64 v Basicu takové příkazy vůbec nemá (kreslení se tam provádí pomocí POKE do VideoRAM).

Způsob, jakým Sinclair dosahoval výsledků, je přímo zázračný.

A nejzázračnější je tvorba obrazu na ZX 81.

Popis platí vlastně i pro ZX 80, kde je místo SCL obvodu několik

impulzu.

Zobrazuje se, jak asi víte, v textovém režimu 32x24 znaků (jako u Spectra).

Jeden řádek může zabrat 33 bajtů. 32 jsou znaky, poslední znak značí konec řádku. To je 792 bajtů. Zdá se vám to snad neohospodárné? 32 bajtů na řádek by stačilo? Ale v jednom kilobitu RAM by VideoRAM o fixní délce 768 bajtů nebyla dost praktická. Proto je přidělována dynamicky.

Pokud prodloužíte program v Basicu tak, že by se už neměl vejít, začne se z VideoRAM ukrajovat a paměť se přidělí pro Basic. Na obrazovce se pak prostě zobrazuje méně řádek! Navíc, VideoRAM je nejenom dynamická, ale je uložena komprimovaně! Pokud je na řádku jeden znak, nenásleduje za ním jednatřicet mezer, ale rovnou znak konce řádku. VRAM tak má průměrnou délku kolem 200 bajtů. Uživatelé to určitě ocení i v případě, že mají připojeno 16 nebo 64 kilo paměti.

Začátek VRAM je uložen v systémové proměnné (DEFILE), celá VRAM musí ležet v dolních 32 kilo paměťové oblasti.

Snímkovou frekvenci v režimu SLOW řídí nemaskovatelné přerušení, v režimu FAST při čekání na klávesu je snímková frekvence určena softwarově zpoždovací smyčkou. Softwarem je řízené i vysílání synchronizačního pulzu.

Během zpracování signálu přerušení se objevuje na horním bajtu adresové sběrnice obsah registru I, do kterého software uložil básovou adresu fontu v ROM. Odtud si ho přečte SCL. SCL obsahuje čítač, ukazující, kolikátý mikrořádek znakových předloh se vykresluje. Software počká, až se vykreslí levá část

borderu, a provede JP (HL), kde HL obsahuje adresu prvního znaku aktuálního řádku ve VRAM (DEFILE+délky předchozích řádků), ovšem zvětšenou o #8000 (nastaveno A15=1).

Pokud SCL na začátku M1 (při čtení kódu instrukce) zjistí nastavené A15, připne dolních 32 kilo RAM místo adresovaných horních 32 kilo. Na datovou sběrnici se tak načte kód znaku pro zobrazení, který SCL načte. Od toho okamžiku deaktivuje paměť a datovou sběrnici uzemní. Procesor tak při čtení kódu instrukce načte #00, a provádí NOP. Během NOPu SCL z kódu znaku a stavu čítače mikrořádků sestaví adresu, na které se



nachází grafická předloha příslušného mikrořádku daného znaku. K tomu použije i básovou adresu, kterou získala během začátku přerušení. Načte obsah získané adresy a odvysílá ho v rychlém sledu (podle krystalu 6.5 MHz) jako osm pixelů. Během těchto osmi impulzů 6.5 MHz dostal procesor čtyři impulzy 3.25 MHz, provedl tedy čtyři takty, což je přesně délka instrukce NOP, kterou mu SCL podstrčila.

U Sinclaire totiž hraje všechno se vším a vše je navzájem magicky provázáno.

PC procesoru se po NOPu zvětší

o 1, ukazuje tedy na další znak ve VRAM.

Znak konce řádku (#76) má nastavený šestý bit. SCL přestane vyčítat znakové předlohy, nepodstrkuje už NOP a nechá procesor kód provést. Protože kód odpovídá instrukci HALT, procesor se zastaví až do konce řádku.

Jak se ale z HALTu na začátku nového řádku probere, když NMI přijde až na začátku nového snímku?

Jako jinde, i tady je to zajištěno zčásti softwarově. Softwarově se na začátku každého řádku do registru R ukládá konstanta, která se, pochopitelně, během každého NOPu při zobrazování zvětší o 1. Protože HALT odpovídá cyklickému provádění NOPů, provádí se odpočítávání i po skončení zobrazování. Až se hodnota v registru R zvětší o 34, bit 6 registru R spadne do nuly. Přerušení je pochopitelně povolené, a protože šestý bit adresové sběrnice, kde se během refresh cyklu obsah registru R objevuje, je spojen se vstupem INT, vyvolá se maskovatelné přerušení.

Rutina pro obsluhu přerušení pak procesor nasměruje zase tak, aby byl počítač připraven zobrazovat další řádek, a nebo ho, po zobrazení celého obrazu, nechá chvíli pokračovat v provádění původního programu. Z toho ho vyvede zase přerušení nemaskovatelné.

Tak a tím jste se dozvěděli všechno podstatné o tom, jak je v ZX 81 řešeno generování obrazu. Pokud se vám to nezdá geniálně promyšlené, nebo dokonce přímo dábelké, pokud se vám nechce teď hned jít a zvolat "Sinclair je bůh a ZX Spectrum je jeho prorok", tak něco prostě není v pořádku.

Didaktik M a RGB výstup

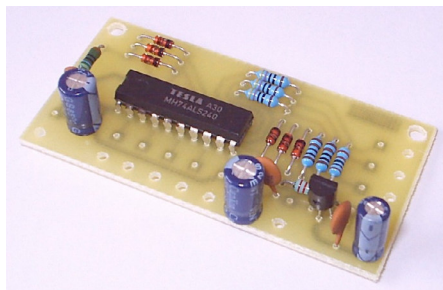
CSS

Úvodní teoretické tlachy:

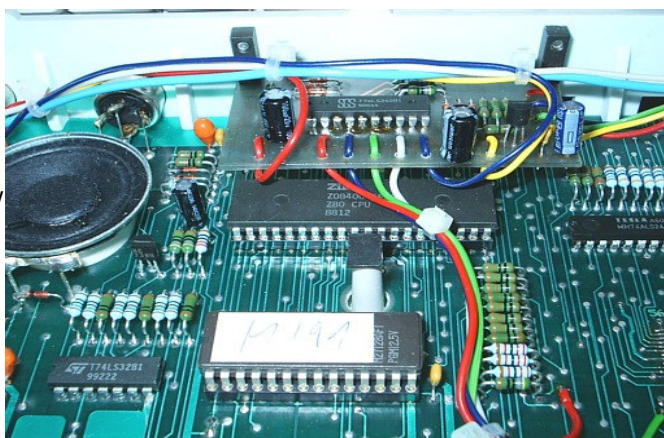
Tento příspěvek volně navazuje na článek ZX Spectrum 128 a RGB přes SCART z minulého čísla ZXM. Popíšeme si zde jednoduchý modul, s jehož pomocí můžeme rozšířit možnosti počítače Didaktik M o připojení televizoru přes signály RGB obsažené v konektoru SCART. Je jen škoda, že výrobce nevybavil tímto výstupem počítač ihned. Prakticky nic to nestojí, vždyť použitý obvod ULA1 z Ruska, jímž je Didaktik M osazen, generuje narozdíl od originálního Ferranti ULA přímo obrazové signály RGB.

O výhodnosti připojení TV přes RGB snad již nemusím nikoho přesvědčovat. Tímto způsobem lze dosáhnout maximální možné kvality zobrazení, kterou je možné takřka přirovnat ke kvalitě obrazu PC na monitoru. Obrazové signály nejsou deformovány několikerým kódováním a dekódováním jako v případě kompozitního videa nebo dokonce další modulací v případě tzv. anténního výstupu. Zjednodušeně by se dalo říci, že čisté obrazové signály vedou z počítače přes vstupní zesilovače přímo do obrazovky televizoru (omlouvám se TV expertům za neobornou terminologii, ale nejsem specialista přes stavbu televizorů, zpracování obrazu apod.). Výsledek je takový, že obraz je dokonale ostrý, čistý, bez jakéhokoliv rušení, nechvěje se. Pokud si například vyplníte obrazovku „bitovou šachovnicí“, nebudou vám přes ní jezdit barevné pruhy v barvách duhy, ale uvidíte krásně čistou „šedou“ plochu. Kdo má dobré oči, možná rozezná i střídající se černé a bílé body.

Popis vlastní konstrukce:



Při tvorbě zapojení byl hlavním požadavek ten, aby veškeré signály odebírané z obvodu ULA byly nejprve odděleny budiči. Tím je zaručeno, že výstupy Uly nebudou dalším obvodem přetíženy, zároveň je ULA chráněna před zničením např. v případě zkratu během našich pokusů, v propojovacím kabelu s televizorem atd. Vlastní sestavený modul je vidět na obrázku 1, umístění v počítači pak na obrázku 2. Modul je horizontálně přišroubován k zadní stěně počítače v místě nad mikroprocesorem. Uchycení je provedeno pomocí dvou dílů ve tvaru „L“



– například kousek plechového pásku ohnutého do tvaru L s vyvrtanými otvory.

Nyní se podívejte na schéma zapojení – obrázek 3. Signály RGB a jasový signál nejprve procházejí 1. polovinou budiče 74LS240, jež zajišťuje oddělení Uly od dalšího obvodu. Tento budič je trvale v aktivním stavu (vstup /EN trvale uzemněn). Po průchodu 1.

polovinou budiče jdou signály jak na 2. polovinu, tak na obvod složený ze třech diod a jednoho rezistoru vedoucího na +5V který zajistí aktivaci 2. poloviny signálem /EN jen v případě, že není zobrazována černá barva. Efekt je ten, že černá je vždy černá bez ohledu na nastavený jas (tak to přece má být). Výstupy 2. poloviny budiče 74LS240 jdou již přes rezistory na vstupy RGB konektoru SCART. Zároveň je úroveň těchto 3 signálů mírně zdvihána signálem jasu (pokud je aktivní) přes 3 rezistory zapojené v sérii s 3 diodami. Ty zajišťují, že se obrazové signály nebudou navzájem ovlivňovat.

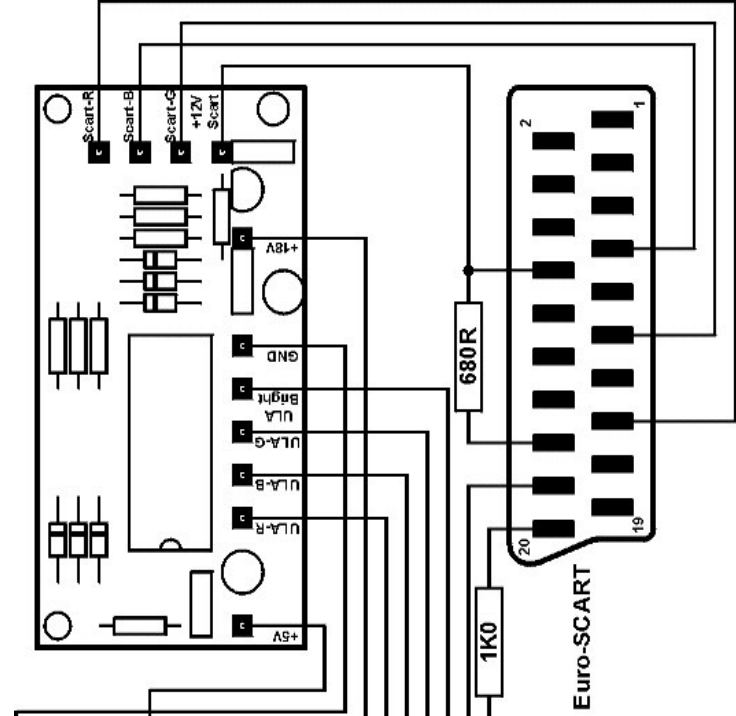
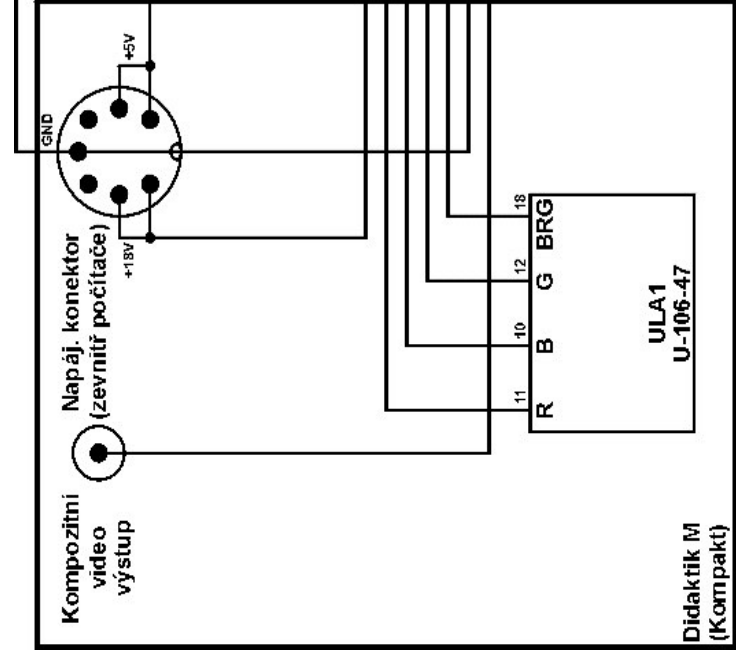
Modul dále obsahuje zdroj konstantního napětí +12V tvořený miniaturním stabilizátorem 78L12. Toto napětí se přivádí na pin 8 konektoru SCART. To umožňuje aktivovat synchronizaci televize signálem video, který se za tím účelem odebírá z počítače přímo z výstupního konektoru VIDEO, a

přivádí se na pin 20 konektoru SCART. Přímou uvnitř kabelové koncovky SCART je ještě zapojen rezistor 680R mezi piny 8 a 16. Protože pin 16 má definovanou vstupní impedanci 75R, vznikne na něm napětí asi 1,2V, které slouží k přepnutí TV do režimu zobrazení přes RGB. Celkové zapojení modulu RGB uvnitř počítače a propojení s TV přes SCART je vidět na obrázku 4.

Příjemné pokoukání.

Dokumentaci k této konstrukci si můžete též stáhnout z mých webových stránek

<http://electronics.mysteria.cz/muzeum.htm>.



Porty a programování

DivIDE

nairam

29.12.2004
02:13:30

Zilogator

ATA	IN	OUT
PORT 163 (#A3)	DATA REGISTER	
PORT 167 (#A7)	ERROR REGISTER	FEATURES REGISTER
PORT 171 (#AB)	SECTOR COUNT REGISTER	
PORT 191 (#BF)	STATUS REGISTER	COMMAND REGISTER

STATUS REGISTER IN 191 #BF	
0	previous command ended in an error
1	index: set to 1 each disk revolution
2	disk data read corrected
3	sector buffer requires servicing
4	seek complete
5	write fault
6	drive is ready
7	controller is executing

ERROR REGISTER IN 167 #A7		
BIT	VALUE	
0	0	DAM found
	1	DAM not found
1	0	Track 000 found
	1	Track 000 not found
2	0	Command completed
	1	Command aborted
3	0	Reserved
4	0	ID found
	1	ID not found
5	0	Reserved
6	0	No error
	1	Uncorrectable ECC error
7	0	Block OK
	1	Bad Block detected

COMMAND FUNCTION		OUT 191(#BF),a
98	E5	check power mode (IDE)
99		execute drive diagnostics
50		format track
77		format unit
EC		identify drive IDE (HDD)
A1		identify drive ATAPI (CDROM,ZIP)
97	E3	idle (IDE)
95	E1	idle immediate (IDE)
91		initialise drive parameters
1x		recalibrate
E4		read buffer (IDE)
08		read DMA with retry (IDE)
09		read DMA without retry (IDE)
04		read multiples (IDE)
20		read sectors with retry
01		read sectors without retry
02		read long with retry
03		read long without retry
40		read verify sectors with retry
41		read verify sectors without retry
7x		seek
EF		set features (IDE)
06		set multiple mode (IDE)
99	E6	set sleep mode (IDE)
06	EE2	standby (IDE)
94	E0	standby immediate (IDE)
03		write buffer (IDE)
0A		write DMA with retry (IDE)
0B		write DMA without retry (IDE)
05		write multiple (IDE)
99		write same (IDE)
00		write sectors with retry
31		write sectors without retry
02		write long with retry
03		write long without retry
0C		write verify (IDE)
07		flush cache
00		nop

Diagram illustrating the LBA (Logical Block Address) structure, showing the mapping of bits to physical components:

- DRIVE 0/1:** Bits 23-16 (8 bits).
- HEAD SELECT:** Bits 15-8 (8 bits).
- LBA:** Bits 7-0 (8 bits).
- DRIVE / HEAD:** Bits 23-16 (8 bits).
- CYLINDER REGISTER:** Bits 15-8 (8 bits).
- SECTOR NUMBER:** Bits 7-0 (8 bits).

The address is shown in hexadecimal format: 0187 (#BB), 0183 (#B7), 0179 (#B3), and 0175 (#AF).

```

ENTRY-POINTS                                JUMPER
AUTOMAPER / MAPRAM                            JP2 = CLOSE

#0000      0          START
#0008      0          ERROR1
#0038      56         INT IM1
#0066      100        NMI
#04C6      10000       SAVE
#0562      1378       LOAD
#3D00-FF   15616-15871 TRDOS

Disconnect automaper
#1FF8-FF   8184-8191  return
              (ZxROM #1FF8 RET)
              (JP #1FF9)

```

CONMEM PORT 227 (#E3)		7.BIT + (1-0.BIT)	
		0	1FFF 2000
OUT	227, 128	8K EEPROM	8K BANK 0
OUT	227, 129	8K EEPROM	8K BANK 1
OUT	227, 130	8K EEPROM	8K BANK 2
OUT	227, 131	8K EEPROM	8K BANK 3
OUT	227, 0	16K ZX ROM	

CONTROL REGISTER PORT 227 (#E3)							
7	6	5	4	3	2	1	0
CONMEM	MAPRAM					BANK	BANK

AUTOMAPER / MAPRAM		PORT 227(#E3)	1-0.BIT	
		0	1FFF 2000	3FFF
OUT 227,0	8K EEPROM / BANK 3 READ ONLY	8K BANK 0		
OUT 227,1	8K EEPROM / BANK 3 READ ONLY	8K BANK 1		
OUT 227,2	8K EEPROM / BANK 3 READ ONLY	8K BANK 2		
OUT 227,3	8K EEPROM / BANK 3 READ ONLY	8K BANK 3 / BANK 3 READ ONLY		
OUT 227,128-131	—————→		CONMEM	

AKTIVACIA **MAPRAM** 6.BIT

OUT 227,64	BANK 3 READ ONLY	8K BANK 0-3
------------	------------------	-------------

Emulátor Apple I na ZX Spectru

dex

Ahoj Spectristi,

tak už opadla první vlna úžasu a překvapení po Foreveru (on nad tím vlastně nikdo stejně nežasl) a pomalu se blíží doba, kdy se ten pocit stane běžnou realitou. Pro ZX Spectrum vznikl další program.

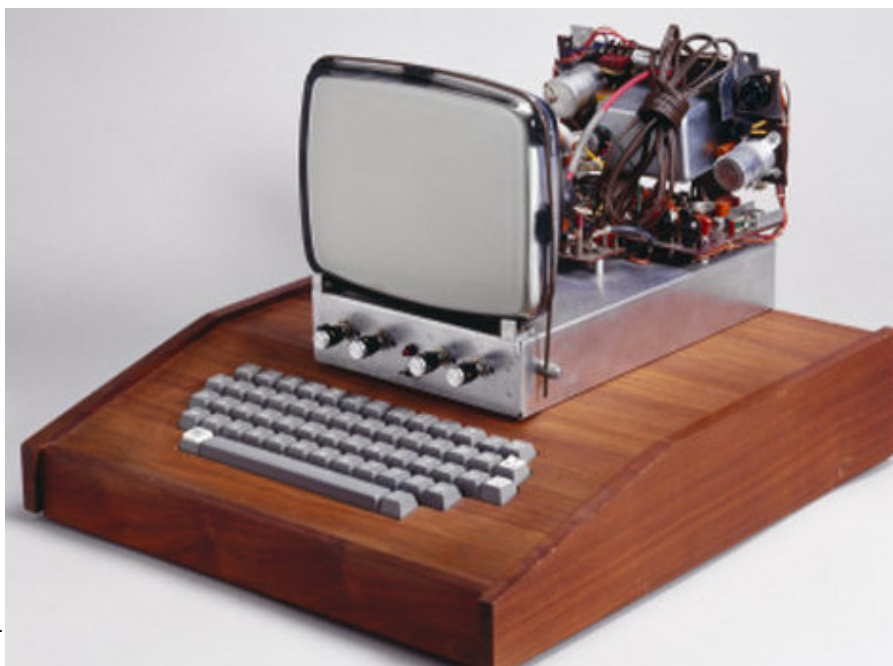
Je jím emulátor počítače Apple 1.

Není to sice první emulátor pro ZX Spectrum (pokud čtete ZX Magazin, tak už znáte další dva), ale drží tři primáty:

- Je to (bude to) nejbližší a nejúplnější emulátor Apple 1 vůbec (programy pro PC a Macintosh, které lze najít na internetu, se v mnoha podstatných věcech jako Apple 1 vůbec nechovají a nerespektují jeho zvláštnosti).
- Je to první emulátor pro Spectrum, emulující počítač, jehož procesor má zcela jinou instrukční sadu než Z80 (předchozí emulátory využívaly kompatibilitu s 8080 a nativní běh Z80).
- Emulátor zahrnuje i relativně pohodlný ladící systém, assembler a disassembler.
- S omezeními ho lze spustit i na Spectru 48.

Jaký je současný stav vývoje?

Emulátor je v hrubých rysech hotov, demonstrační betaverze neobsahuje žádnou podporu kazetových ani



instrukce nevyužívá, ve své době byly jejich instrukční kódy rezervovány pro možné budoucí rozšíření.

Zadruhé, reálný stroj je po zapnutí "zaseknutý" a je nutné ho po zapnutí resetovat. Emulátor naběhne rovnou.

Dále je použitý font o jeden bod nižší než originální. Šlo by využít fullscreen mód, záměrem ale bylo vyšetřit místo na obrazovce pro

jiných I/O operací. Vývojová unstable verze má už hotové kazetové operace na straně Apple 1, chybí dopsat fyzické ukládání a čtení souboru na Spectru, a to pro tři nejrozšířenější systémy (*tape compatible* - MB-02 a emulátory, snad včetně *DivIDE*, *D80*, *BETASHIT*). Součástí balíčku hotového emulátoru budou i programy pro Apple 1, hlavně ve strojovém kódu (*Šachy*, *přistání na Měsíci*, *StarTrek* a další klasika, *Forth*, rozšířený *Basic* s plovoucí desetinnou čárkou, *assembler*, ...).

Co emulátor neumí?

Je toho hned několik.

Především emuluje jenom klasické dokumentované instrukce procesoru 6502. Ty ilegální interpretuje jako NOP, takže se může stát, že parametr takto použité instrukce vyhodnotí jako jinou platnou instrukci. Nutno dodat, že dostupný software pro Apple 1 tyto

zobrazení údajů monitoru. Z důvodu nižšího rozměru je pak poněkud odlišná i grafická podoba fontu. Nejedná se ale o odchylku příliš závažnou.

Za další, originální stroj umožňuje kromě úpravy na krokování jednotlivých instrukcí, která v emulaci je zahrnuta, i úpravu na krokování jednotlivých taktů, která v emulátoru chybí.

Možnost zobrazení stavu datové a adresové sběrnice pomocí LED diod byla nahrazena monitorem.

Emulátor pochopitelně neumožňuje rozšiřování systému připojováním dalšího hardware (uživatelské sloty A a B) ani jiné úpravy. Neemuluje tiskárnu, která se k originálu dala připojit, ani RS 232 interface od Vince Briela.

Slot C je poněkud neohospodárně obsazen kazetovým interface (v betaverzi

chybí) a slot D je stejně nevhodně využít pro připojení monitoru a klávesnice.

Důležitá změna, která se týká i hardwarové obsluhy, se týká připravenosti terminálu, především monitoru.

Firma Apple přesně určila, jak mají vypadat vstupní/výstupní rutiny, a jejich součástí je test připravenosti terminálu, jelikož není možné sypat na originálním počítači data na monitor neomezenou rychlostí. Verze pro Spectrum je naopak ready neustále. Důrazně proto doporučuji nespolehat se na to, že je emulátor neustále připraven na výstup znaku, a používat korektní rutiny. Jinak nebude program na originálním stroji správně fungovat.

Ačkoliv emulátor správně emuluje všechna mně známá zrcadlení a přeslechy z adresové sběrnice, adresy pro vstup a výstup na terminál používejte zásadně pouze podle dokumentace. Správný tvar je tento:

```
readkey    lda D011
           bpl readkey
           lda D010
displayout bit D012
           bpl displayout
           sta D012
```

Co emulátor vůbec neemuluje, je programování obvodu 6820, z tohoto důvodu odmaskovává znak #7F pro výstup. Předpokládá se využití obvodu pouze ve stavu, do něž je uveden rutinou na začátku ROM Apple 1, součástí inicializační sekvence je i znak #7F.

Výrazným prvkem je zařazení řídicích kódů #2 (#82) a #3 (#83) kompatibilních s počítači Replica 1 revision C a vyšší od Vince Briela. Význam kódů je CLEAR SCREEN a CURSOR HOME.

Stejně jako u Replicy je Basic uložen v ROM a dostupný hned po zapnutí (na Apple se nahrával do RAM, předadresované do slotu E hardwarovou úpravou).

Jaká je rychlost emulace?

Instrukce nejsou časované na takty, v závislosti na I/O operacích pak běží emulace rychleji než originál (asi

tříkrát) až pomaleji (zhruba desítkrát). To si zaslouží podrobnější rozbor, který, pokud vás aktuálně nezajímá, můžete přeskočit.

Jádro emulace procesoru, 6502 Virtual Machine (*protože je to dlouhé, dále zkracuji jen jako VM*), běží podstatně pomaleji než originál procesoru. 6502 v Applu běží zhruba na jednom MHz (refresh ho zpomaluje na 0.960). Taktová náročnost instrukcí je zhruba stejná jako u Z80 (zhruba 8 taktů na instrukci). Z80 má tedy sice náskok triapiúnásobku rychlosti, pro full speed práci by ale bylo nutno emulovat jednu instrukci 6502 pomocí tří až čtyř instrukcí Z80.

Občas se to sice podaří dodržet, běžná instrukce je ale emulována kódem o pár instrukcí delším, navíc je potřeba přičíst vlastní režii emulátoru. Tím se rozumí vyčítání instrukcí z emulované paměti (případně stránkování), zobrazení výpisu monitoru (jeho zapnutí zpomalí běh), čtení řídicích kláves emulátoru, které nejsou součástí emulované klávesnice Apple, a emulační cyklus.

Z toho vyplývá celkový výkon cca desítkrát nižší než má původní 6502.

Reálný stroj ovšem brzdí procesor výrazně i při I/O operacích, kde stroj emulovaný koná vše v rámci zpracování jedné emulované instrukce pomocí relativně optimalizovaných rutin.

Z toho vyplývá jednoduchý závěr:

Čím více výstupů a méně počítání, tím je emulace rychlejší.

Čím více počítání a méně výstupů, tím je emulace reálně pomalejší.

Čím více vstupů (a výstupů, t.j. počítač čeká na člověka), tím je emulace subjektivně rychlejší.

Čím méně vstupů (a méně výstupů, t.j. člověk čeká na počítač), tím je emulace subjektivně pomalejší.

Dobře proto funguje práce v monitoru Apple 1, reálnou rychlostí běží i hry psané ve strojovém kódu. Naopak pomaleji běží programy psané ve vyšších programovacích jazycích

(Basic).

Pro budoucí využití VM proto platí: VM je vhodná pro časově méně náročné kódy (snad SID hudby z C64). VM je vhodná pro emulaci počítačů přímo využívajících strojový kód jako programovací jazyk nebo jeho velký podíl.

Z vyšších programovacích jazyků je vhodný Forth, přijatelné C, ještě použitelný pak celočíselný Basic.

VM je vhodná pro emulaci počítačů s jednoduchým hardware nebo hardware totožným se ZX Spectrem (obvodová řešení s AY-3-891x apod.). VM je vhodná pro procesy hojně komunikující s uživatelem.

Dobře těmto podmínkám vyhovují amatérské počítače konce 70. let s hexadecimální klávesnicí a sedmsegmentovkovým displejem, případně stroje pracující v textovém módu (KIM-1, Commodore PET). Emulace grafické paměti s přímým přístupem, pokud není organizací totožná s VRAM ZX Spectra, bude zákonitě na ZX Spectru pomalejší.

Vyloučena je emulace hardwaru nenapodobitelného na ZX Spectru (byť je možná, například vysoká barevnost emulovaná odstíny šedi, emulace vysokého rozlišení za cenu ztráty čitelnosti nebo HW spritů za cenu neúnosného zpomalení).

Emulace alien procesoru vypadá jako složitá věc, jak si ji vlastně naprogramovat?

Řešení je jednoduché. Dvě části emulátoru budou dostupné jako open source.

První je assembler/disassembler. Je 6502 specifický, naprosto neoptimalizovaný, ale funkční. I s jakýmsi popisem v češtině byl uveřejněn v seriálu článků na SpeccyWebu MB Maniax. Adresa je www.mb-maniax.net.

Nepředpokládám jeho širší využití ani že ho bude někdo optimalizovat a vyvíjet, i když, kdo ví.

Druhá je VM samotná.

Je decentně optimalizovaná na rychlost (ale nijak dokonalá ani ideální). VM v současné podobě emuluje 6502 a nic jiného (*může ale sloužit jako předloha k tvorbě VM pro jiný procesor. Do toho se ale sotvakdo pustí, je to fakt dřina*).

Ke svému běhu potřebuje VM jen dvě subrutiny, nazvané LOADA a STOREA.

LOADA má jako vstupní parametr v HL adresu, jejíž obsah se má načíst do A.

STOREA má jako vstupní parametr v HL adresu, do které se má uložit obsah A které je taky vstupním parametrem.

Protože I/O prostor pro paměť a porty je na 6502 společný, obě subrutiny samy rozhodují podle adresy, do a z kterého prostoru bude operace směřovat. Samy si tak zajišťují prostor emulované RAM, emulované ROM, čtení klávesnice, zobrazování a další I/O.

Pak už zbývá jen doplnit emulační cyklus o režii (řídící kávesy) a odchytávací body (pro emulaci jinak nerealizovatelných operací, např. přesměrování tape operací na disk). A tím je nový emulátor počítače založeného na 6502 v kostce hotov.

Ovládání Apple 1 vypadá jako hromsky složitě. Co má člověk dělat, když se na obrazovce objeví lomítko a kurzor?

Řešení je opět jednoduché. Stisknout šipku nahoru nebo dolů, najet na volbu HELP a stisknout Enter. Protože nápoředný text má délku asi 14 kB, není dostupný na ZX 48. Navíc je psaný anglicky, takže asi ne všem pomůže, takoví nešťastníci se musí spolehnout na informace obsažené v tomto článku. Kopie helptextu by měla být i v ZIP archivu s betaverzí emulátoru jako .TXT. Pokud tam není, hodte ho na hlavu tomu debilovi, co vám to tak blbě zkopíroval.

Jak že se přesně liší verze pro ZX 48 a ZX 128?

Programově se neliší nijak, protože neexistují.

Existuje jen jedna verze programu,

která se rozhodí po dostupné paměti.

Na ZX 48 chybí helptext, je emulováno jen 16 kB RAM místo 32 kB, a ve fiální verzi zřejmě přibude ještě nějaké omezení ohledně vstupu a výstupu na kazetu.

Příklad ovládání disassembleru:

Vstupte do dissassembleru. Stiskněte M, zadejte 0000. Stiskem EXT MODE přejděte do režimu assembleru. Po stisku Enter vložte instrukci, ukončete Enterem, šipkou dolů se posuňte za ni. Vložte tak tento program:

```
lda #03
bit D012
bpl 0002
sta D012
clc
adc #01
cmp #7F
beq 0000
jmp 0002
```

Potom buď stiskem C nastavte PC na nulu a vraťte se do emulátoru či program odkrokuje stisky šipky doleva, nebo se vraťte do emulátoru a spusťte pomocí OR. I zde můžete krokovat stiskem šipky vlevo.

Tak ve zkratce úvod k helptextu:

Apple 1 je první osobní počítač světa vyvinutý Steve Wozniakem a Steve Jobsem. Obsahoval procesor MOS 6502 na cca 1 MHz, 4-32 kB RAM (bez dalších rozšíření), televizní terminál a ASCII klávesnici s dalšími dvěma tlačítky navíc (reset a clearscreen). K dispozici byl i kazetový interface, celočíselný Apple Basic se nahrával z kazety do RAM.

Emulátor zahrnuje kompletní sestavu Apple 1, klávesnici a terminál, dokumentovaný instrukční soubor 6502 a 32 (na ZX 128) nebo 16 (na ZX 48) kB RAM. Apple Basic je uložen v ROM jako u počítačů Briel Computers. Zahrnuta je i emulace obvodu pro krokování jednotlivých instrukcí, zabudovaný monitor, assembler, disassembler a jednoduché trasování.

Ovládání emulátoru:

CS + 1 (EDIT) - teplý RESET (hardwarové tlačítko).

CS + 2 (CAPS LOCK) - CAPS LOCK, při startu zapnut.

(podle doporučení firmy Apple; v příkazech monitoru i Basicu je nutno používat velká písmena).

CS + 3 (TRUE VIDEO) - vypnout monitor (rychlejší).

CS + 4 (INV. VIDEO) - zapnout monitor (pomalejší).

CS + 5 (vlevo) - provedení jedné instrukce.

CS + 6 (dolů) - vyvolá MENU, kurzor nahoře.

CS + 7 (nahoru) - vyvolá MENU, kurzor dole.

CS + 8 (doprava) - plynulý běh.

CS + 9 (GRAPH) - CLEAR SCREEN (hardwarové tlačítko).

CS + 0 (DELETE) - ESCAPE. Zruší vstup řádku.

Pro zrušení jednoho znaku (rubout) se používá SS + 0 (podtržítka, “_”).

SS + Q - CTRL+D, ukončuje povel AUTO.

SS + W - kód “clear screen”, použitelný v programech v Basicu.

SS + E - kód “cursor home”, použitelný v programech v basicu.

Některé znaky jsou dostupné v kombinaci se symbolshiftem.

Ovládací klávesy pro disassembler (dostupný z MENU):

SPACE nebo EDIT - ukončit disassembler.

M - nastavení adresy.

ENTER - změna obsahu adresy.

CS + SS (EXT MODE) - přepíná mezi assemblerem a hexa editorem.

A, X, Y, S, C, P - nastaví registr A, X, Y, S, PC nebo P.

CS + 5 (vlevo) - provedení jedné instrukce.

CS + 6 (dolů) - posun na následující instrukci.

CS + 7 (nahoru) - posun na předchozí byte.

CS + 8 (doprava) - posun na následující byte.

Příkazy systémového monitoru:

Prompt systémového monitoru je “\”.

X v následujícím textu nahrazuje hexa číslo (0-9,A-F).

XXXX představuje adresu (X, XX, XXX nebo XXXX).

XX představuje byte (X nebo XX).

XXXX - nastaví adresu, zobrazí její obsah.

XXXX XXXX - zobrazí obsahy daných adres.

XXXX.XXXX - zobrazí obsah od adresy “.” do adresy.

.XXXX - výpis od poslední použité počáteční adresy do XXXX.

XXXX:XX XX - ukládá data od zadané adresy.

:XX XX - ukládá data od poslední zadané počáteční adresy.

XXXXR - spustí kód od zadané adresy.

R - spustí kód od poslední zadané počáteční adresy.

“Online” opravy - zadání XXXX:yyyyXX interpretuje jako XXXX:XX.

Příkazy Apple Basicu

Prompt Basicu je “>”. Návrat do monitoru zajistí RESET.

Studený start (maže program) je od E000 nebo E2B0.

Teplý start (ponechá program) je od E2B3.

expr = aritmetický výraz

var = proměnná (číslo, pole, řetězec)

val = číslo (-32767 až 32767 včetně)

Výrazy:

-expr záporná hodnota expr.

NOT expr 0 pokud expr nenulové, 1 pokud expr je nula.

expr * expr násobení.

expr / expr zaokrouhlené dělení.

expr + expr sčítání.

expr - expr odčítání.

expr = expr 1 pokud rovno (pravda), 0 pokud nerovno (nepravda).

expr > expr 1 pokud první je větší než druhý.

expr < expr 1 pokud první je menší než druhý.

expr >= expr 1 pokud první je větší či roven druhému.

expr >= expr 1 pokud první je menší či roven druhému.

expr # expr 1 pokud nerovno, to samé jako expr <> expr.

expr AND expr 1 pokud žádný není 0.

expr OR expr 1 pokud oba nejsou nulové.

expr MOD expr zbytek po dělení.

var\$ = var\$ porovnání řetězců, 1 pokud stejné.

var\$ # var\$ porovnání řetězců, 1 pokud různé.

Funkce:

ABS (expr) absolutní (kladná) hodnota expr.

SGN (expr) 0 pokud expr 0, 1 když kladná, -1 když záporná.

PEEK (expr) čte obsah paměťové adresy (decimálně).

RND (expr) vrací náhodné číslo mezi 0 a expr.

LEN (var\$) vrací délku řetězce.

Povely:

HIMEM = (expr) nastaví horní hranici paměti pro program (decimálně).

Počáteční hodnota je 4096. Maže program.

LOMEM = (expr) nastaví dolní hranici paměti pro program (decimálně).

Počáteční hodnota je 2048. Maže program.

AUTO val1,val2 doplňuje čísla řádků od val1 s krokem val2 (nebo 10). Pro ukončení stiskněte CTRL+D (SS+Q na ZX).

CLR maže proměnné, pole, prováděné cykly FOR a GOSUB.

DEL val1,val2 maže řádky od val1 do val2.

LIST val1,val2 zobrazí programové řádky od val1 do val2, řádek val1, nebo celý program.

RUN val1 provede CLR, pak spustí program od řádku val1 nebo od začátku.

SCR smaže celý program (scratch).

Příkazy:

Příkazy označené “C” lze použít i jako povely (t.j. v přímém prováděcím módu).

C LET var=expr vyhodnotí expr a přiřadí hodnotu var. LET lze vynechat (var=expr).

INPUT item zobrazí hlášení a “?”, očekává od uživatele vstup hodnoty pro proměnnou. Vstup více proměnných musí být oddělený čárkou.

INPUT A : INPUT A,B,C\$,D(4)

INPUT “ENTER A,A\$,B(3)”,A,A\$,B(3)

C PRINT item zobrazí obsah

proměnné, výsledek výrazu nebo

zprávu, oddělené “,” (zobrazení

do dalšího sloupce) nebo “;,” (bez odělování).

PRINT “A\$=”;a\$,2*48

C TAB (expr) zobrazí zadaný počet mezer.

FOR var=expr1 TO expr2 STEP expr3

NEXT var opakuje příkazy mezi FOR a

NEXT, hodnota var začíná od expr1,

s krokem expr3 dosahuje hodnoty expr2, pak ukončuje cyklus.

C IF (expr) THEN příkaz

IF (expr) THEN číslo řádku

pokud expr je nepravda, provede

další příkaz za IF, pokud je expr prav-

da, pak se provede příkaz nebo skok na řádek uvedený za THEN.

C GOTO expr skok na řádek s daným číslem.

GOSUB expr

RETURN GOSUB skočí na řádek s

daným číslem, GOSUB skočí zpět na

řádek následující za posledním provedeným GOSUB. Může být vnořeno.

C DIM var1 (expr), var2 (expr)

nastaví rozměry pole nebo řetězce.

C REM text Uvozuje poznámku,

neovlivňuje běh programu.

END ukončuje běh programu.

C POKE expr1,expr2

uloží hodnotu expr2 na adresu paměti expr1 (decimálně).

C CALL expr skočí na zadanou adresu

paměti, propojuje Basic s rutinami

ve strojovém kódu. Návrat do Basicu

zajišťuje instrukce RTS (return from subroutine).

Hlášení o chybách:

*** SYNTAX ERR syntaktická chyba nebo překlep.
 *** >32767 ERR vložená nebo spočtená hodnota je menší než -32767 či větší než 32767.
 *** >255 ERR hodnota omezená na rozsah 0 až 255 je mimo rozsah.
 *** BAD BRANCH ERR pokus o skok na neexistující řádek.
 *** BAD RETURN ERR pokus o provedení více RETURN než bylo provedeno GOSUB.
 *** BAD NEXT ERR pokus o provedení více NEXT než bylo provedeno odpovídajících FOR.
 *** >8 GOSUBS ERR více než 8 vnořených GOSUB.
 *** >8 FORS ERR více než 8 vnořených smyček FOR.
 *** END ERR poslední provedený příkaz nebyl END.
 *** MEM FULL ERR potřeba paměti je větší než dostupná paměť.
 *** TOO LONG ERR příliš mnoho vnořených závorek.
 *** DIM ERR řetězec už byl DIMenzován.
 *** RANGE ERR rozsah podřetězce byl větší než DIM nebo menší než 1.
 *** STR OVFL ERR počet přiřazených znaků překročil DIMenzovanou hodnotu.
 *** STRING ERR provedena nepovolená operace s řetězcem.
 RETYPE LINE do INPUT byla vložena nepovolená data. Očekává se nový vstup.
 \ objeví se po zadání více než 128 znaků, očekává nový vstup.

Proměnná:	Jméno:	Příklad:
-----	-----	-----
číslo	písmeno	A N
	nebo písmeno a číslo	A1 N6
pole	písmeno	A() N()
	nebo písmeno a číslo	A1() N6()
řetězec	písmeno	A\$ T\$
	* První prvek pole A(1) je identický s A.	
	* A\$(a,b) nebo A\$(a) jsou podřetězce (od, do [konce]).	

Mapa paměti:

```

0000 zero page 0024-002B systémové proměnné monitoru
                004A-00FF systémové proměnné Basicu
0100 1st page 0100-01FF zásobník
0200          0200-027F buffer pro vstup řádku
                0280-07FF volná RAM
0800          0800-0FFF defaultně vyhrazeno pro program v Basicu
1000          1000-3FFF volná RAM
4000          4000-7FFF volná RAM (ZX 128)
                zrcadlo RAM 0000-3FFF (ZX 48)
-----[ konec RAM ]-----
8000          8000-9FFF nepoužito
A000 expansion A000-AFFF rozšiřující slot A, nepoužito
B000 expansion B000-BFFF rozšiřující slot B, nepoužito
C000 expansion C100-C1FF (kazetový interface + tape ROM) nyní nepoužito
D000 hardware D010 klávesnicový registr PIA
        controls D011 klávesnicová data PIA
                D012 terminálový registr PIA
                D013 terminálová data PIA
-----[ začátek ROM ]-----
E000          E000-EFFF Basic ROM
F000          F000-FEFF nepoužito
                FF00-FFFF Monitor ROM

```

ZX Spectrum

zachraňuje

data z poškozeného HDD!

8bc

Náhlé úmrtí v rodině není žádná sláva. Stejně tak jsme na tom, když náhle odejde do věčných lovišť pevný disk se všemi důležitými daty, zvláště pak, když na zálohování zase tak nedbáme. Máme pro vás návod, jak zachránit data, která jsou s PC normálním způsobem neopravitelná. A věřte nebo ne, pomůže vám staré dobré ZX spectrum!

Rádi bychom se s vámi podělili o zážitek a zkušenost, kterou nejspíš každý nezažije (a nikdo o ni určitě nestojí). Bylo parné letní odpoledne, když nás navždy opustil harddisk Samsung SP1614N (160 GB). Nikdo to nečekal a tak to bylo něco jako rána z čistého nebe. Do té doby disk fungoval zcela bez problémů. Ačkoliv naše dosavadní zkušenost s disky této značky byla velmi dobrá, nic netrvá věčně a u harddisků platí dvojnásob, že to "jednou přijde". Smutně jsme si vzpomněli na několik sloganů o zálohování dat a začali přemýšlet nad způsoby, jak zachránit data z "mrtvého" disku. Ačkoliv se HDD v BIOSu identifikoval, číst z něj bylo zřehla nemožné

- zuřivě klepal a skřípal. Zkoušeli jsme několik utilit na obnovu dat z poškozených disků (např. Easy Recovery od Ontracku, R-Studio apod.), nicméně jakmile

software narazil na problémovou oblast, celý systém vytuhl a bylo po záchraně dat. Trochu zoufalým pokusem byla i výměna elektroniky ze stejného zdravého disku na nemocný. Bezúspěšně.

Už jsme to viděli velmi špatně



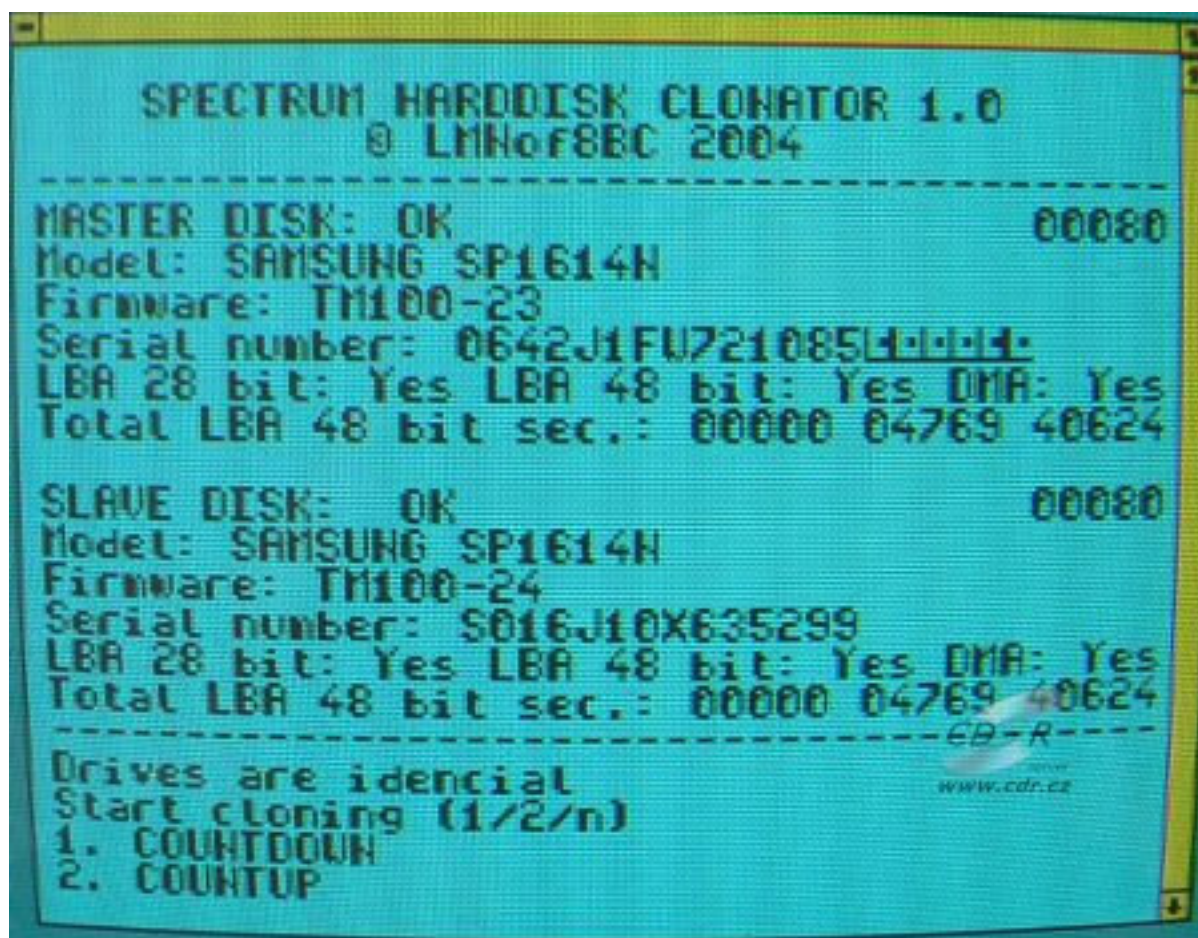
- začali jsme se pít po firmách specializujících se na obnovu dat z poškozených disků. V tom přišel další nápad - pokusit se vadný disk "naklonovat"

na zdravý a recovery souborů provádět až na kopii na zdravém disku. Zkusili jsme DriveCopy od Powerquestu, DOSovský klonovač harddisků HDClone... všechno bylo marné - obě utility ztroskotaly při pokusu číst z disku. Všechno bylo marné.

Námi testované utility měly snahu data číst, ale úspěšnost naklonování byla mizivá - podařilo se nakopírovat pouze zlomek sektorů od začátku disku. Teprve teď jsme si začali uvědomovat, co se asi s naším diskem stalo - dle výsledků to vypadalo na poškozenou hlavu nebo poškozené plotny.

Už jsme byli rozhodnutí zaslat disk firmě specializující se na obnovu dat z poškozených harddisků...

a v ten okamžik opět vypluly vzpomínky na 80. a 90. léta, kdy jsme veškerý volný čas trávili u legendárního (a dnes raritního) počítače ZX Spectrum. V



té době jsme (kromě zuřivého hraní her samozřejmě) vyráběli a prodávali diskový řadič MB-02+ (dodnes patrně nejlepší zařízení svého druhu). MB-02+ kromě jiného disponuje DMA obvodem a dvěma IDE porty (ty, které známe ze základních desek pro připojování harddisků a optických mechanik). Což takhle zkusit naklonovat vadný harddisk na zdravý na té nejnižší úrovni - na úrovni ATA příkazů. Za pokus to stálo... i na ZX Spectru.

Netrvalo dlouho a po 3 dnech nočního kódování v assembleru Z80 byla na světě utilita ZXCLONATOR, která nejprve nadetekuje připojená IDE zařízení a poté umožňuje kopírování harddisku, sektor za sektorem, z MASTER na SLAVE zařízení (v našem případě z nemocného disku na zdravý). Lze zadat LBA pozici (pomocí tří 16bitových čísel), od které

chceme kopírování provádět a jakým směrem chceme postupovat (vzestupně / sestupně). Narazili-li ZXCLONATOR na vadnou oblast, lze ji rychle přeskočit o určitý počet sektorů. Ačkoliv to zpočátku vypadalo nepravděpodobně, ZX Spectrum za 8 dní překopíroval nemocný 160 GB harddisk na zdravý kousek. Otázkou už jen bylo, zda-li zachráněná data byla čitelná. Nebyla.

Disk s překopírovanými daty se tvářil, jakoby na něm žádný file systém nikdy nebyl (původně NTFS). Teprve tehdy přišel ke slovu Ontrack s jeho Easy Recovery. Zkusili jsme pustit RAW recovery (obnova dat bez existujícího file systému). A za půl dne - světe div se - naprostá většina navždy ztracených souborů byla opět na světě. Bez ZX Spectra by se nám to nikdy nepodařilo.

Co z toho plyne? Možná existují specializované softwary na obnovu dat z poškozených harddisků, které nejsou přístupné běžným uživatelům. Možná existuje i jiné než naše retro-řešení (třeba zkusit napsat podobnou nízkoúrovňovou utilitu na PC - kolegovy i mé programátorské znalosti však končí u ZX Spectra a assembleru Zilog Z80). Možná by bylo snazší poslat disk firmě, pro niž je obnova dat z mrtvých harddisků chlebem. Bylo však vzrušující vidět v roce 2004 ZX Spectrum v akci - při činnosti, která na PC s dostupnými nástroji prostě možná nebyla!

Jako důkaz o tom, že se nejedná o aprílový žertík v srpnu, přikládáme několik fotografií ze záchranné akce.

A kdy jste si naposled dělali zálohu dat vy?

K - mouse

Kempston mouse turbo interface

velesoft

Tento interface je kompatibilní s původním Kempston mouse interface, které podporuje pouze dvě tlačítka a pracuje jen s originál amiga myši. K-MOUSE TURBO interface má několik dalších vylepšení, ale je zpětně plně kompatibilní. Rusové ve svém zapojení Kempston mouse, které okopírovali z originálu, podporují již třítlačítkovou amiga myš. Náš „turbo“ interface umožňuje připojit PS/2 nebo USB myš, případně i originál myš amigáckou. Podporuje až 4 tlačítka myši a dokonce i kolečko.

Nyní si povíme něco o tom, jak vlastně Kempston mouse interface funguje a na jakých portech je čitelné. O komunikaci s myši se stará samotné interface, uživatel jen čte z patřičných portů aktuální souřadnice pro osu X a Y, nebo stav tlačítek. Každá osa má rozsah od 0 do 255 (1 bajt). Po prvním zapnutí interface jsou souřadnice v obou osách 0. Při pohybu myši vpravo je hodnota osy X postupně inkrementována, při pohybu vlevo se hodnota naopak snižuje. To samé platí i pro osu Y, která je inkrementována při pohybu dolů. Teď se podíváme například na hodnoty osy X při delším pohybu myši vpravo. Postupně se zvyšuje souřadnice z 0, 1, 2, 3 ... až do 255, což je vlastně maximální hodnota jednoho bajtu. Interface ale neošetřuje mezní hodnoty, takže se při pokračování pohybu vpravo souřadnice (bajt) dál zvyšuje a po 255 „přeteče“ znovu do 0 a pokračuje dál (1, 2, 3 ... 255). To platí i při pohybu vlevo, kdy se hodnota snižuje až do 0 a pak „přeteče“ od 255 a dál snižuje

(254, 253, 252 ... 0). Z portu 64223 zjistíme aktuální stav všech tlačítek (stisknuté/povolené), ale to už nám přiblíží následující přehled.

OSA X je čitelná na portu 64479(#FBDF)

OSA Y je čitelná na portu 65503(#FFDF)

TLAČÍTKA jsou čitelná na portu 64223(#FADF)

Čtená data z portu:

D0 = pravé tlačítko

D1 = levé tlačítko

D2 = prostřední tlačítko (používá jen RUSKÁ a TURBO K-MOUSE)

D3 = 4. tlačítko (používá jen TURBO K-MOUSE)

D4 až D7 = 4-bitové binární čítač polohy kolečka (podle ruského zapojení)

Jednotlivé bity představují jednotlivá tlačítka. Přitom bit v log.0 značí stisknuté a log.1 nestisknuté. Jak už jsme si řekli, původní Kempston mouse interface používá jen dvě tlačítka (D0 a D1). Ruská Kempston mouse používá tlačítka tři (D0, D1 a D2). K-MOUSE TURBO používá celkem 4 tlačítka (D0, D1, D2, D3). Na D4 – D7 je čitelná 4-bitová hodnota polohy kolečka. Tento čítač pracuje stejně jako osy X a Y. Při pohybu kolečka směrem dolů se čítač inkrementuje (0, 1, 2, ... 14, 15, 0, 1). Tak to by bylo k hlavním portům myši.

AMIGA MOUSE a

KEMPSTON JOYSTICK na portu 31

Interface obsahuje jeden přepínač, dva switche a konektor CANON 9. Switch vpravo slouží k povolení portu 31, na kterém je buď čitelná hodnota 0 (pokud je levý switch odstraněn). Nulování portu 31 zabrání některém hráči v detekci fire joysticku. Při nasunutém levém switchi bude na portu 31 (#1F) čitelná buď AMIGA MOUSE nebo KEMPSTON JOYSTICK. Defaultně je po zapnutí nastavená na portu 31 amiga mouse. Mezi amiga mouse a kempston joystickem se přepíná jednoduše stiskem tlačítka myši (bude aktivní amiga mouse) nebo fire na joysticku (bude aktivní joystick) zapojeném do konektoru CANON 9. Při emulaci amiga mouse mají data portu 31 následující uspořádání:

D0 = XB

D1 = YB

D2 = XA

D3 = YA

D4 = levé tlačítko

D5 = pravé tlačítko

D6 = prostřední tlačítko

D7 = čtvrté tlačítko (podporuje jen TURBO K-MOUSE)

Na D0 a D2 jsou přímo stavy clonek myši pro osu X, na D1 a D3 pro osu Y. Na původně nevyužitém bitu D7 je nově přidané čtvrté tlačítko myši. Při stisknutém tlačítku je bit v log. 1.

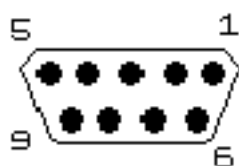
Stiskem fire joysticku se odpojí amiga mouse a bude čitelné pouze kempston joystick. Pak platí klasické rozložení:

D0 = joy VPRAVO
 D1 = joy VLEVO
 D2 = joy DOLU
 D3 = joy NAHORU
 D4 = joy FIRE 1
 D5 = joy FIRE 2
 D6 = joy FIRE 3
 D7 = vždy log.0

Při aktivním směru nebo fire je odpovídající bit nastaven na log. 1. (po aktivaci joysticku přestává pracovat i K-MOUSE a na jejích portech načteme chybné hodnoty)
 Jednoduše řečeno: pravý switch zapne čitelný port 31, a levý switch pak (pokud je port zapnutý) určí, jestli bude na portu AMIGA MOUSE/JOYSTICK nebo bude nulován (kvůli hrám).

Možnost vypnout port 31 je důležitá, pokud k počítači hodláte k TURBO K-MOUSE interface připojit i jiné zařízení používající tento port (např. UR4, KEMPSTON interface, MB02+, D40, M/P). Pokud port zůstane i přesto povolený, může dojít k poškození některé z periférií. Port 31 musíte zablokovat i při provozu na počítačích s již integrovaným obvodem 8255 (Didaktik GAMA nebo Didaktik Kompakt). To platí i pro počítače TIMEX, které mají interní kempston joystick interface. Při provozování s betadiskem port 31 bude přístupný, ale musíte K-MOUSE interface připojit až za něj. V opačném případě, při připojení betadisku až za K-MOUSE, musí být port 31 blokován!

Na následujícím obrázku je popis zapojení konektoru CANON 9.



1 UP
 2 DOWN
 3 LEFT
 4 RIGHT
 5 FIRE_2
 6 FIRE_0
 7 +5V
 8 GND
 9 FIRE_1

Extra mód

Dvoupolohové přepínač je určen k povolení nebo blokování extra módu. Při povolení je možné kdykoli za provozu současným stiskem prvního+druhého+třetího tlačítka vyvolat extra mód, umožňující měnit nastavení interface jako třeba citlivost myši nebo prohození funkce levého a pravého tlačítka. Vyvolání extra módu: stiskněte a uvolněte současně tři tlačítka. Tak se aktivuje extra mód, kde následným stiskem jednoho z tlačítek měníte vlastnosti interface. V extra módu mají tlačítka tento význam: Levé tlačítko = tvrdě nuluje port 31 (pokud je povolený) a nebude čitelná A-MOUSE ani JOYSTICK.

Pravé tlačítko = prohodí funkci levého a pravého tlačítka (projeví se jen u K-MOUSE)

Prostřední tlačítko = sníží/zvýší 4x citlivost (rychlost) pohybu myši
 Po uvolnění stisknutého tlačítka v extra módu se automaticky extra mód opustí.

V době, kdy je extra mód aktivní, nejsou na portu K-MOUSE (64223) čitelná tlačítka.

Ovladače pro K-mouse

V této části se podíváme na několik verzí ovladače pro Kempston mouse. V podstatě nejrozšířenější verze ovladače:

```
;XECUTE MOUSE
;TAKE COORDINATES FROM
CURPOS==COORD
MOUSE LD HL,(COORD)
      LD BC,#FBDF
      LD DE,(OLDCO)
      IN A,(C)
      LD (OLDCO),A
      SUB E
      JR Z,NM_X
      JP P,MX_PL
      ADD A,L
      JR C,ZER_X
      XOR A
```

```
ZER_X LD L,A
      JR NM_X
MX_PL ADD A,L
      JR C,BEX_Z
      CP #FD ;MAXIMUM X
      JR C,BEX_B
BEX_Z LD A,#FD ;MAXIMUM X
BEX_B LD L,A
NM_X LD B,#FF
      IN A,(C)
      LD (OLDCO+1),A
      SUB D
      JR Z,NM_Y
      NEG
      JP P,MY_PL
      ADD A,H
      JR C,ZER_Y
      XOR A
ZER_Y LD H,A
      JR NM_Y
MY_PL ADD A,H
      JR C,BEY_Z
      CP #BE ;MAXIMUM Y
      JR C,BEY_B
BEY_Z LD A,#BE ;MAXIMUM Y
BEY_B LD H,A
NM_Y LD A,H
      CP #FF
      JR C,BIGY
      LD H,#FF
BIGY CP #02 ;MINIMUM Y
      JR NC,SMALY
      LD H,#02 ;MINIMUM Y
SMALY LD A,L
      CP #FF
      JR C,DIRY
      LD L,#FF
DIRY CP #02 ;MINIMUM X
      JR NC,DIMENS
      LD L,#02 ;MINIMUM X
DIMENS LD (COORD),HL
      LD BC,#FADF
      LD HL,CONTRB
      IN A,(C)
      CPL
      AND 7
      RLCA
      RLCA
      RLCA
      RLCA
      OR (HL)
      LD (HL),A
      RET
```

```
COORD DEFB 0,0;sem se uklada-
ji souradnice pro osu X a Y
OLDKO DEFB 0,0;WORKING
```

Jiná verze:

```
;MOUSE DRIVER WITH FIRE BUTTON
AUTOCONFIG
```

```
;(_) Andrey Rachkin'95
```

```
JR MDRV
DIRECTZ DEFB 0 ;FIRE
DEFB 0 ;UP
DEFB 0 ;DOWN
DEFB 0 ;RIGHT
DEFB 0 ;LEFT
DEFB 0 ;CANCEL
```

```
MCOORD DEFW 0 ;LAST CURSOR
```

```
COORDS ;IN PIXELZ
```

```
MPORTS DEFW 0 ;LAST READED
```

```
MAUSY COORDS
```

```
NONDEF AND 3 ;HERE COMEZ
```

```
BUTTONZ CONTROL IF FIRE BUTON
NOT DEFINED
```

```
JR Z,MDRV4
```

```
;IF NONE BUTTON
```

```
;PUSHED
```

```
CP 1
JR Z,NONDEF_
XOR A
LD (MDRV3+2),A
LD A,5
LD (MDRV2+2),A
NONDEF_ LD HL,0
LD (MDRV1),HL
POP IX
```

```
; ***** MAIN PROC OF MOUSE-
DRIVER *****
```

```
MDRV PUSH IX
LD HL,DIRECTZ
PUSH HL
POP IX
XOR A
LD (HL),A ;CLEARING
INC HL ;OF
LD (HL),A ;DIRECTZ
INC HL ;BUFER
LD (HL),A
INC HL
LD (HL),A
INC HL
LD (HL),A
INC HL
```

```
LD (HL),A
INC HL
LD BC,#FADF ;BUT-
TONZ CONTROL
IN A,(C) ;READ-
ING FROM PORT OF BUTTONS
CPL
MDRV1 JR NONDEF ;JR UNTIL
FIRE ;BUT-
TON NOT DEFINED
RRA
MDRV2 RL (IX+0) ;FIRE
RRA
MDRV3 RL (IX+5) ;CANCEL
;***** COORDS CONTROL *****
MDRV4 LD HL,(MCOORD)
;FORM LAST CURSOR COORDS
LD DE,(MPORTS)
;FROM LAST READED MOUSE COORDS
LD BC,#FBDF
IN A,(C) ;READING
FROM PORT X-COORD (0-#FF)
LD (MPORTS),A
SUB E
JR Z,MDRV9
JP P,MDRV6
LD (IX+4),1 ;MOVE
LEFT
ADD A,L
JR C,MDRV5
XOR A ;MIN
X-COORD
MDRV5 LD L,A
JR MDRV9
MDRV6 ADD A,L
LD (IX+3),1 ;MOVE
RIGHT
JR C,MDRV7
CP #FE ;MAX
X-COORD
JR C,MDRV8
MDRV7 LD A,#FE ;MAX
X-COORD
MDRV8 LD L,A
MDRV9 LD B,#FF
IN A,(C) ;READING
FROM PORT Y-COORD (0-#FF)
LD (MPORTS+1),A
SUB D
JR Z,MDRV14
NEG
JP P,MDRV11
LD (IX+1),1 ;MOVE
```

```
UP
ADD A,H
JR C,MDRV10
XOR A ;MIN
Y-COORD
MDRV10 LD H,A
JR MDRV14
MDRV11 ADD A,H
LD (IX+2),1 ;MOVE
DOWN
JR C,MDRV12
CP #BF ;MAX
Y-COORD
JR C,MDRV13
MDRV12 LD A,#BF ;MAX
Y-COORD
MDRV13 LD H,A
MDRV14 LD (MCOORD),HL ;NEW
CURSOR POSITION IN PIXELZ
POP IX
RET
; ***** DRIVER END *****
```

A další varianta s akcelerací rychlosti pohybu:

```
;*** Kempston-mouse driver
OLDKO DEFW 0
COORD DEFW 0
CONTRB DEFB 0
MOUSE ;(c) 1994-98 Max Ro-
manoff, Slava Mednonogov ;*
*****
LD BC,#FBDF
LD DE,(OLDKO)
LD HL,(COORD)
IN A,(C)
LD (OLDKO),A
SUB E
JR Z,MSEx0
JP P,MSEx1
NEG
CALL MSElog
NEG
ADD A,L
JR C,MSExe
XOR A
JR MSExe
MSEx1 CALL MSElog
ADD A,L
JR C,MSEx11
CP 248 ;max X
JR C,MSExe
```

```

MSEx11 LD A,248 ;max X      ;(COORD+0)= souradnice X sipky      ;o v atribut.casti videoram.
MSExe LD L,A                ;(COORD+1)= souradnice Y sipky      ;vychozi souradnice sipky jsou
MSEx0 LD B,#FF              ;(OLDCO+0)= stara osa X mysi      ;pri prvnim spustenim ovladace
IN A,(C)                    ;(OLDCO+1)= stara osa Y mysi      ;ulozene na adresach CO-
LD (OLDCO+1),A              ;Pred prvnim spustenim ovla-      ORD+0,+1
SUB D                        dace                               ;=====
JR Z,MSEy0                  ;je nutne spustit rutinku od      ==
JP M,MSEy1                  ;adresy KOREKCE, ktera ovla-      XMAX equ 255
CALL MSElog                  daci                               XMIN equ 0
NEG                          ;podstrci aktualni souradnice      YMAX equ 191
ADD A,H                      ;mysi jako predchozi hodnotu.      YMIN equ 0
JR C,MSEye                  ;Tak zajisti, ze sipka bude po      ;start ovladace K-MOUSE
XOR A                        ;spusteni presne na vychozich      org 49152
JR MSEye                    ;souradnicich.                      START
MSEy1 NEG                    ;ovladac lze volat dvojim        ;- - - - -
CALL MSElog                  zpuso                               ;tato rutina se vola pred
ADD A,H                      ;bem. Pri volani od adresy        prvni
JR C,MSEy11                  ;KMNORMAL se testuje pouze osa    ;m spustenim ovladace. Slouzi
CP 190 ;max Y                ;X,Y a 3tlacitka mysi stejne      ;pouze k vyrovnani odchylek
JR C,MSEye                  ;jako u original K-MOUSE.          ;mysi s vychozi polohou sipky.
MSEy11 LD A,190 ;max Y      ;Dalsi zpusob volani je od      KOREKCE
MSEye LD H,A                 ;adr.                               ;ulozi aktualni osu X
MSEy0 LD (COORD),HL          ;KM-TURBO, kdy ovladac testuje      ld a,%11111011
LD B,#FA                     ;osy X,Y a 4 tlacitka              in a,(223)
IN A,(C)                      ;mysi.                            ld (OLDCO),a
LD (CONTRB),A                ;Ovladac vzdy uklada stav        ;ulozi aktualni osu Y
RET                           tlaci                             ld a,%11111111
;                             ;tek negovany, takze log.1 =      in a,(223)
MSElog CP 4                   ;stisknute tlacitko              ld (OLDCO+1),a
RET C                         ;(BUTTONS+0) =                    ret
ADD A,A                       ;D0=prave tlacitko              ;- - - - -
JR C,MSElgC                   ;D1=leve tlacitko                ;start ovladace pro puvodni
CP 8*2                         ;D2=prostredni tlacitko          ;K-MOUSE(osy X,Y a 3 tlacitka)
RET C                         ;D4=4.tlacitko(jen u KM-TURBO)    KMNORMAL ld a,7
ADD A,8                       ;v praxi staci jen spoustet      ld (BUTMASK+1),a
JR C,MSElgC                   pod                               jr ORIGKM
CP 120                         ;prerusenim tento ovladac a      ;start ovladace pro novou K-
RET C                          pak                               MOU
MSElgC LD A,120               ;z adresy COORD+0aCOORD+1 vy-      ;SE-TURBO(osy X,Y,4 tlacitka
RET                             cit                             KMTURBO ld a,15
                                ld (BUTMASK+1),a
                                ORIGKM
                                ;osa X
                                MAXIMX ld a,XMAX
                                ld (SEM1+1),a
                                ld (BE_Z+1),a
                                xor a
                                ld (MODIFY),a
                                ld a,(COORD)
                                ld l,a
                                ld a,(OLDCO)

```

Následuje částečně okomentovaný zdrojový kód ovladače myši pro osu X,Y a tlačítka. Tento ovladač jsem použil do všech mnou upravených her.

;OVLADAC K-MOUSE/K-MOUSE TURBO
;(MAXIMX+1) = maximum v ose X
;(MINIMX+1) = minimum v ose X
;(MINIMX2+1)= -II-
;(MAXIMY+1) = maximum v ose Y
;(MINIMY+1) = minimum v ose Y
;(MINIMY2+1)= -II-

;at nove souradnice pro sipku.
;Ovladac jeste uklada primou
;adresu bajtu videoram, kam
uka
;zuje souradnice sipky.
;Na adrese (KMVRAM)lezi
2B.adr.
;videoram pro vykresleni sip-
ky.
;A na adrese (KMATR)je 2B.adr.
;ukazujuci na odpovidajici
mist

```

ld e,a
ld a,%11111011
in a,(223)
ld (OLDCO),a
call RUTINA
MINIMX cp XMIN
jr nc,SMALX
MINIMX2 ld l,XMIN
SMALX ld a,l
ld (COORD),a
ld a,h
ld (ODCH),a

;osa Y
MAXIMY ld a,YMAX
ld (SEM1+1),a
ld (BE_Z+1),a
ld a,237
ld (MODIFY),a
ld a,(COORD+1)
ld l,a
ld a,(OLDCO+1)
ld e,a
ld a,%11111111
in a,(223)
ld (OLDCO+1),a
call RUTINA
MINIMY cp YMIN
jr nc,SMALY
MINIMY2 ld l,YMIN
SMALY ld a,l
ld (COORD+1),a
ld a,h
ld (ODCH+1),a

;tlacitka
ld a,%11111010
in a,(223)
cpl
;podle rezimu prace maskuje
jen
;3(KMNORMAL)nebo
4(KMTURBO)tlac
;itka mysi
BUTMASK and 15
ld (BUTTONS),a

;nakonec se spocita adr.vram
;podle souradnic mysi
GO
ld e,a
ld a,(COORD+0)
ld l,a
and 7
;ulozi posun v pixelech
ld (KMPIX),a

ld a,l
srl a
srl a
srl a
ld l,a
ld a,(COORD+1)
rla
rla
and %11100000
or 1
;ulozi nizsi bajt adr.vram
ld (KMVRAM),a
;a jeste jednou. protoze nizsi
;bajt je shodny i pro adr.
atrib
;utove casti vram
ld (KMATR),a

ld a,(COORD+1)
and 7
ld l,a
ld a,(COORD+1)
and %11000000
rrca
rrca
rrca
or 64
or 1
;ulozi vyssi bajt adr.vram
ld (KMVRAM+1),a

ld (KMATR+1),a

ret
;-----
--
RUTINA
sub e
;odchylku zalohuj do H
ld h,a
;pri nulove odchylce preskoc
jr z,NM
;tady se meni NEG/NOP podle
X/Y
MODIFY neg
jp p,M_PL
add a,l
jr c,ZER
xor a
;posun souradnice sipky
ZER ld l,a
jr NM
M_PL add a,l
jr c,BE_Z
;MAX
;modifikuje 0 podle X,Y,K
SEM1 cp 0
jr c,BE_B
;modifikuje 0 podle X,Y,K
BE_Z ld a,0
BE_B ld l,a
NM ld a,l
cp 255
jr c,BIG
ld l,255
BIG ret
;-----
--
;predchozi souradnice X,Y
OLDCO defb 0,0

;pozice sipky
COORD defb 128,96

;odchylka v ose X,Y
ODCH defb 0,0

;tlacitka
BUTTONS defb 0

;adr.bajtu vram podle sourad-
```

```

nic
KMVRAM defw 0

;adr.atrib.casti vram
KMATR defw 0

;posun sipky v pixelech vpravo
;muze byt v rozmezi 0-7
KMPIX defb 0

```

```
LEN equ $-START
```

Všechny tyto ovladače mají ošetřené maximum a minimum pro osy. Někdy ale tyto testy okrajů nejsou třeba, nebo jsou přímo nežádoucí. Pak si vystačíme s velice jednoduchým ovladačem. Dejme tomu máme šipku, kterou chceme pohybovat po celé ploše obrazovky, to je plocha o rozměru 256 (osa X) x 192 (osa Y). Při pohybu šipka může přelézat okraje obrazovky. Problém je jen v tom, že osa Y má u K-MOUSE stejný rozsah (0-255) jako osa X. Nejjednodušší řešení bude upravit rozsah osy Y z 0-255 na 0-191. Nějak takhle bych to řešil já:

```

KMDRV LD A,255
IN A,(223) ;Načte do A osu Y (0-255)
SRL A ;a vydělí ji dvěma
RL L ;Bit 0 zálohuje do reg.L
SRL A ;Ještě jednou dělí Y dvěma (teď je A = původní Y:4)
RL L ;Zálohuje bit 1 původní osy Y
LD H,A ;H = původní osa Y:4
RLCA ;Y:4 zase vynásobí dvěma (A = původní Y:2)
ADD A,H ;K Y:2 přičte Y:4, a tak A=3x (Y:4)
LD H,A ;a výsledek (beze zbytku) vloží do H
RR L ;Načte z L původní bit 1 osy Y
JR NC,NEXT1 ;a pokud byl v

```

```

log.1, tak
INC H ;inkrementuje novou hodnotu osy Y
NEXT1 RR L ;Ještě načte i bit 0 osy Y
JR NC,OSA_Y ;a pokud byl v log.1, tak
INC H ;inkrementuje novou hodnotu osy Y
;Nyní už je spočítaná nová osa Y v registru H
OSA_Y LD A,251
IN A,(223) ;Pokračuje ve čtení osy X, kterou beze změny
LD L,A ;uloží do reg.L
BUTON LD A,250
IN A,(223) ;Nakonec se čte port tlačítek
CPL ;který negujeme, kvůli jednoduššímu testování stisku
AND 7 ;a omezíme test jen na tři tlačítka

```

;Na výstupu je v reg.H nová osa Y v rozsahu 0-191 a v reg.L osa X(0-255)
;V registru A je na bitech D0-D2 stav tlačítek(bit v log.1=stisknuté) a
;současně je nastaven flag NZ (non zero),pokud je stisknuté jakékoli ze tří tlačítek. Jediná nevýhoda ovladače je v tom, že se změnou rozsahu osy Y se
;změní ve stejném poměru i rychlost pohybu kurzoru. Místo 4:4 bude poměr
;4:3, ale věřte mi, že běžné uživateli nic nepozná.

Ovladač může být ještě kratší, ale s omezením na plochu 128x128. Jednoduše toho docílíme vydělením každé osy dvěma:

```

LD A,251
IN A,(223)
SRL A
LD L,A

LD A,255
IN A,(223)

```

```

SRL A
LD H,A

LD A,250
IN A,(223)
CPL
AND 7

```

Rychlost pohybu šipky se tím současně sníží v obou osách na polovinu.

Obsluha kolečka myši

Stav (poloha) kolečka je čitelný na bitech 4-7 portu tlačítek. Následující ovladač obsluhuje jen samotné kolečko a zdroják byl přepsán z diskmagazínu DON NEWS 19. Na adrese (W_COORD) je po návratu nová osa kolečka v rozsahu 0-255 bez ošetření maxima a minima.

```

WHEEL LD H,0
CALL IN_WHEEL
SUB H
JR Z,S_DRIV+3
JP M,MS
CP 15-6
JP NC,GET_UP1
JP GET_DN

MS NEG
CP 15-6
JP NC,GET_DN1
JP GET_UP

GET_DN1 NEG
AND #0F

GET_DN CALL W_SPEED
LD H,A
LD A,(W_COORD)
SUB H
JR S_DRIV

GET_UP1 NEG
AND #0F

GET_UP CALL W_SPEED
LD H,A
LD A,(W_COORD)
ADD A,H
S_DRIV LD (W_COORD),A
RET

W_SPEED CP 1
RET Z

```

```
LD B,A
LD E,A
ADD A,E
DJNZ $-1
RET
```

```
IN_WHEEL LD BC,#FADF
          IN A,(C)
          AND #F0
          RRCA
          RRCA
          RRCA
          RRCA
          LD (WHEEL+1),A
          RET
```

```
W_COORD DEF B 0
```

Před prvním spuštěním doporučuji provést CALL IN_WHEEL. Tak se vyrovná odchylka polohy kolečka.

Úprava her a programů

Úprava programů pro K-MOUSE není nijak složitá. Dáme si za úkol například upravit nějakou 2D střilečku z bočního podledu (CHRONOS nebo R-TYPE). V první řadě si zjistíme, kam si hra ukládá souřadnice (pozici) rakety a v jakém rozsahu se pohybují (maximum a minimum ve vertikálním a horizontálním směru). Najdeme si v paměti volné místo pro ovladač. Pak zjistíme adresu rutiny pro čtení klávesnice nebo joysticku a zmapujeme si její funkci. Tyto rutiny většinou vrací stav o směru pohybu a fire jako jednobajtovou hodnotu v nějakém registru. Pro čtení kempston joysticku najdeme často něco podobného:

```
#C000 LD BC,#1F
#C003 IN A,(C)
#C005 LD (#5B00),A
atd....
```

Tady je ideální místo čtení joysticku volat ovladač myši s testem tlačítek. Ten po návratu vrátí v registru A hodnotu 16 při stisknutí tlačítka, nebo 0 při nestisknutí. Od adr.#C000 přepíšeme kód na:

```
#C000 CALL MFIRE
#C003 NOP
```

```
#C004 NOP
#C005 LD (#5B00),A
Tak zajistíme přesměrování na test
tlačítka, jehož kód můžeme napsat
nějak takto:
MFIRE LD A,#FA
          IN A,(#DF) ;do reg.
A načte stav tlačítek z portu
#FADF
          BIT 1,A
;testujeme jen levé tlačítko
          LD A,#00
          JR NZ,END ;pokud
není stisknuté, vrátíme v reg.
A hodnotu 0
FIRE LD A,#10 ;při
stisku tlačítka emulujeme stav
fire joysticku
END RET
```

Ještě si vyhledáme rutinu, která mění souřadnice podle původně načtené hodnoty z portu joysticku. To bude vypadat asi jakto:

```
#8200 LD A,(#5B00) ;vzvedne
stav joysticku
#8203 LD B,A
#8204 RR B
#8205 CALL NZ,#8255 ;odskok
při směru vpravo
#8208 RR B
#8209 CALL NZ,#8289 ;odskok
při směru vlevo
atd....
```

Na adrese #8255 musí být něco jako:

```
LD A,(#6253) ;protože
se tato rutina volá pro směr
vpravo, je jasné, že je na
této adrese osa X
CP #CF ;test
maxima v ose X
RET Z ;návrat,
pokud je pozice na maximu
INC A
LD (#6253),A ;posune
souřadnici a uloží zpět
RET
```

V předchozím případě, při přesměrování na test fire, nám rutina vrací kromě bitu 4 (fire) všechny bity v log.0. Takže program nikdy nebude

detekovat směry joysticku. Toho hned využijeme. Na adrese #8200 přepíšeme první instrukci na CALL MOUSEDRV (volání ovladače myši pro osu X a Y). Adresa MOUSEDRV ukazuje na volné místo v ram, kam umístíme ovladač. Na začátku ovladače si zazálohujeme všechny registry, které budeme používat, pak vložíme vlastní ovladač, na jehož konci těsně před návratem opět zazálohované registry obnovíme.

Ovladač myši:

```
MOUSEDRV PUSH AF
          PUSH HL
          PUSH DE
```

Následuje kód ovladače

```
RETMDRV POP DE
          POP HL
          POP AF
```

```
LD A,(#6253) ;
Před návratem vložíme do reg-
istru A opět stav joysticku
(vlastně jen fire)
RET
```

V ovladači nastavíme správně rozsahy pro osy X,Y a adresy, kam si hra ukládá souřadnice. Teď už by nám mělo fungovat ovládání souřadnic a fire pomocí myši.

U některých her stačí jen místo instrukce pro čtení portu joysticku vložit jediné CALL našeho ovladače. Ten rovnou změní souřadnice, a v reg. A vrátí stav fire. Ne každá hra se s tím ale vyrovná, tak je lepší volat ovladač pro osy až přímo na začátku rutiny, která původně souřadnice měnila. V úvodním menu si zjistíme, kam si hra ukládá zvolený typ ovládání a na začátek ovladače přidáme ještě test, jestli je nastavený kempston. Pokud bude ovládání jiné, ovladač se rovnou vrátí, aniž by měnil souřadnice (před návratem musíme provést ještě instrukci LD A,(#5B00), kterou jsme přepsali).

Nakonec ještě přepíšeme v menu název „KEMPSTON“ na „K-MOUSE“.

Adresace portů

KEMPSTON MOUSE TURBO interface používá všechny své porty úplně adresované.

Funkce port
adresace portu binárně (od A15 do A0)

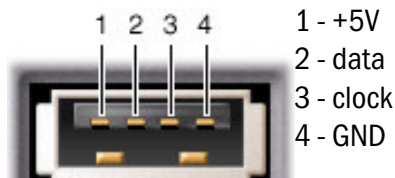
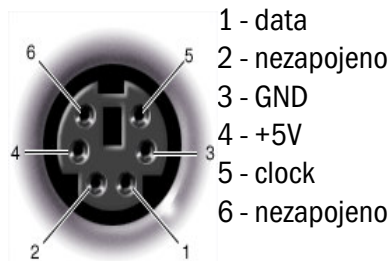
tlačítka 6 4223(#FADF)
xxxx0x0 11011111
osa X 644 79(#FBDF)
xxxx0x1 11011111
osa Y 65503(#FFDF)
xxxx1x1 11011111

a-mouse/joystick 31(#1F)
xxxxxxx 00011111

Porty K-MOUSE nekolidují s téměř žádnou periferií. Problém může nastat jen v případě, kdy k ZX připojíte K-MOUSE a IDEFACE pro MB02 s porty podle Drona. Tam by docházelo ke kolizi na datové sběrnici. Pouze přidání port 31 může kolidovat se spoustou zařízení, proto je vypínatelný.

PS/2 nebo USB?

Na desku interface můžete osadit buď PS/2, nebo USB konektor. Po osazení je nutné na spodní straně plošného spoje přerušit některé vodiče vedoucí na piny připájeného konektoru. V každém případě se tento zákrok musí provést ještě před prvním zapnutím interface, jinak dojde ke zkratu na pinech obvodu PIC! Pro úplnost uvádím opět pinout obou konektorů.



Provoz s připojenou A-mouse

Interface může pracovat s připojenou myší PS/2 nebo USB, ale vzhledem k tomu, že máme k dispozici i konektor canon 9, určený pro připojení kempston joysticku, můžeme do něj zapojit i originál amiga myš. Po stisku tlačítka takto připojené myši přestane pracovat původní připojená PC myš a funkčně ji plně nahradí amiga myš. A k čemu nám to vlastně bude? Kdo má doma amiga myš a nechce zbytečně kupovat jinou ps/2, stačí, když ji připojí místo joysticku. Navíc tím ušetří dalších 90 Kč, protože obvod PIC (je v patici) ani PS/2 konektor nebude třeba osazovat. Konektor canon je plně zapojený, a tedy fungují všechna 3 tlačítka. Fyzicky připojená amiga myš bude vždy pracovat maximálně jako třítláčková.

Externí a interní verze

Externí verze má už přímo na desce napájené všechny konektory. Hlavní obvody XILINX XC9572XL a PIC jsou osazeny v patičkách. Při poruše stačí obvod jednoduše vyměnit za nový. Není nutný zásah do počítače. Může se kdykoli odpojit a použít u jiného počítače. U počítačů Didaktik M, Didaktik Kompakt a ZX128+2A je vystouplý profil nad sběrnici, a bude třeba mezi interface a konektor zapájet krátký mezikus, který sběrnici prodlouží. Pokud z interface odstraníme plastový kryt, může se použít i u těchto typů počítače bez nutnosti prodlužovat sběrnici. Interní verze se zatím připravuje. Je určena pro zapojení přímo na desku počítače. Musíme si nejprve najít vhodné místo pro umístění tak, aby šel zavřít horní kryt. Tato verze bude kompletně v SMD provedení na malé

desce. Nebude obsahovat žádný konektor ani přepínač. Interface se připojí pomocí plochého vécežilového kabelu k signálům procesoru, ostatní přepínače a konektory se propojí rovněž plochým kabelem k desce interface. Plochý kabel k procesoru musí být pokud možno co nejkratší! Interní verze určitě půjde připojit do Didaktiků M/Kompakt, ZX Spectrum+ a ZX128+2, ve kterých je dostatek místa. Na druhou stranu například do klasického ZX Spectra (gumák) se už asi nic interního nevejde. Nevýhodou je nutnost pájet v počítači hromadu drátů a nemožnost interface kdykoli odpojit. SMD součástky budou připájené přímo na desce a budou se obtížně měnit (při případné poruše).

Přeprogramování interface

Deska obsahuje konektor (6 pinů) pro připojení k programátoru obvodů XILINX CPLD. Konektor je zapojený podle kabelu vývojového kitu KRAKEN, který si můžete objednat na www.hw.cz

Osazení plošného spoje

Na desku osadíme patici DIL 18, určenou pro obvod PIC. Ten může být typu PIC 16F84A nebo PIC16F627/PIC16F628, a to v pouzdru PDIP18 ve verzi 4MHz. Při použití PIC16F627/PIC16F628 neosazujeme rezistor R4 a kondenzátor C4, nebo jen jednoduše u PICu nožku č.16 nezasuneme do patice. Konektor pro sběrnici ZX Spectra použijeme SLOT 62, zkrácené o 4 krajní piny.

Popis funkce

O komunikaci s PS/2 myší se stará programovatelný obvod PIC. PIC pracuje jako konvertor z PS/2 na amiga mouse. Bližší informace o tomto konvertoru můžete najít na <http://rdc.euro.ru/ps2m> nebo stáhnout přímo z Aminetu: <ftp://ftp.wustl.edu/pub/aminet/>

hard/hack/ps2m.lha .

K-MOUSE interface obsahuje ještě druhý důležitý obvod, CPLD (XILINX XC9572XL), jehož autorem jsem já. Moje CPLD se stará o konverzi signálů AMIGA myši na KEMPSTON MOUSE. Tento obvod umí pracovat i jako transparentní kvůli možnosti číst na portu 31 stavu clonek AMIGA myši. Obvod PIC není nutné osazovat, interface může bez problémů pracovat i bez něj, ovšem za předpokladu, že použijeme originál AMIGA myš zapojenou do konektoru canon. Při osazení PICu můžeme používat běžnou PS/2 myš a u K-MOUSE tak podporovat i čtvrté tlačítko + kolečko myši.

Při návrhu obsahu CPLD jsem vycházel z původního zapojení ruské KEMPSTON MOUSE, které si můžete prohlédnout v jednom z předchozích ZX Magazinů.

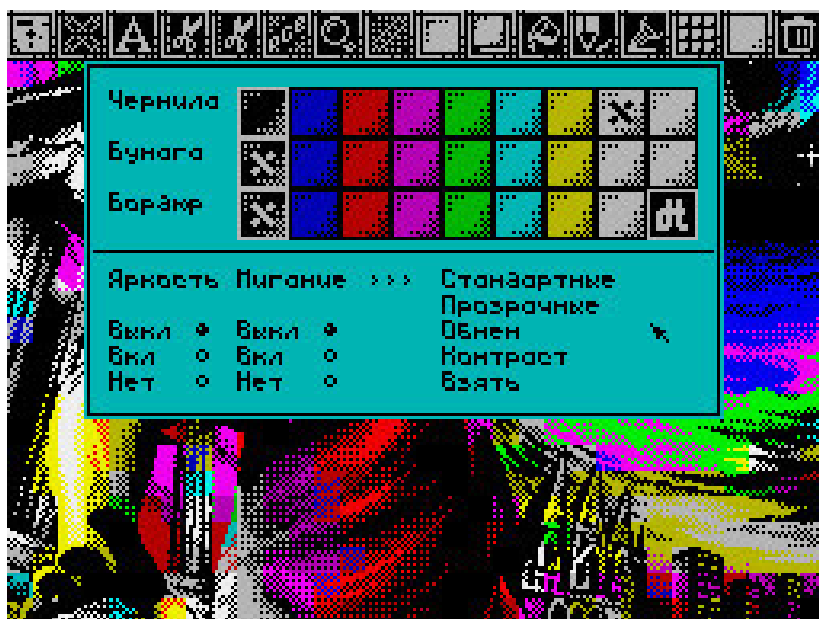
Takové zapojení je sice jednoduché, levné a funkční, ale souřadnice pro Kempston mouse mění 4x pomaleji než by mohlo. Zapojení jsem vylepšil a myš pracuje 4x rychleji (stejně jako u většiny emulátorů ZX). Některé programy jsou ovšem dost náchylné na rychlejší pohyby myši. To je přesně příklad ARTISTu 2. Ovladač pro K-MOUSE tam zpracovává jen malé odchylky při pohybu, a při větší rychlosti začne šipka dost nesmyslně poskakovat. Proto jsem do K-MOUSE TURBO interface přidal možnost přepnutí citlivosti myši podle původního zapojení.

Software pro K-mouse

Existuje několik originál her a programů, které podporovaly ovládání myši. Mezi ně patří například CARRIER COMMAND nebo BATTLE COMMAND. U poslední hry je zřejmě špatně

napsaný ovladač a tak myš nefunguje. Také kreslicí program ARTIST 2 umí pracovat s myši.

Pak je tu ještě spousta her a programů z Ruska, například UFO, HERO QUEST 1 a 2, SPACE CRUSADE, ARCANOID, GUNMAN, LAST COURIER 2, WOLF 2004, LEMMINGS 2, HEADBALL, MOORHUHN, SPARROWS, TECH-NODROM, WALKER, FAST BREED, KAZMUS, SHASKI, KING, FARSPACE, a z programů třeba BURRIAL GRAPHIC EDITOR, různé TRACKERY, COMMAND-ERY a diskové magazíny.



Nedávno jsem našel ještě ICON GRAPHIX 128. Nakonec jsou tu ještě moje předělávky her pro K-MOUSE. U některých her podporuji i 4. tlačítko myši, které umí obsloužit pouze K-MOUSE turbo interface. Všechny úpravy jsem řešil tak, aby bez jakéhokoli omezení mohli hru používat i majitelé třítlačítkové ruské Kempston myši, případně ji spouštět v emulátorech. Zatím jsem upravil jen několik her: R-TYPE, HEROQUEST 1 a 2, LEMMINGS 1, CHRONOS, SILKWORM a SIMCITY. Většinou ve verzích 128 REMIX. U R-TYPE funguje levé tlačítko jako FIRE1, pravé jako FIRE2 a prostřední zapne/vypne autofire. Ve hře LEMMINGS 1 (128remix) levé tlačítko je FIRE, pravým tlačítkem aktivujete sledování pozice lemminga, a prostřední tlačítko zruší funkci sledování. V

SIMCITY má levé tlačítko funkci fire, prostřední = pohyb v menu nahoru, pravé = pohyb v menu dolů. Pokud máte vícetlačítkovou myš, stiskem 4. tlačítka vyvoláte mapu ostrova. Upravil jsem jak původní, tak českou (přeložil UNIVERSUM) verzi SIMCITY.

Emulátory a K-mouse

Dnes už každý pořádný emulátor ZX Spectra emuluje i Kempston mouse. Emulátor REALSPECTRUM je jeden z nejlepších, co se týče věrnosti

emulace ZX, ale podporuje jen dvě tlačítka myši. Nejlepší zkušenosti mám s X128 a UNREAL-SPECTRUM, kde fungují již tři tlačítka. V UNREALSPECTRU bude nutné zapnout i funkci SWAP BUTTONS, která prohodí funkci levého a pravého tlačítka. Pak je tu emulátor ZX-EMUL, které podporuje i kolečko a je 100% kompatibilní s K-MOUSE TURBO.

Tento bezva emulátor najdete na <http://lion17home.narod.ru/> a na http://trd.speccy.cz/press/press_an.htm si stáhněte diskmagazíny DON NEWS 19 a 20 podporující kolečko. V DON NEWS 19 je schema zapojení kolečkové myši, zdrojáky ovladačů a tester.

Zdrojový kód

Následující zdrojový kód je psaný v jazyce ABEL

```
MODULE KMLAST9572
// 3.3.2005 VELESOFT
```

```
//plna verze pro externi pruchozí interface s
CPLD - XILINX XC9572XL PC44
//u vseh portu je pouzita uplna adresace !!!
```

```

//D4-D7 lezi 4 bitovy citac polohy kolecka.
Zapojeni bylo prevzato z
//DON-NEWS19(donnws19)

//pin c.34 vypina 4.bitovy citac na portu
tlacitek.

//po zapnuti je port 31 defaultne
nulovan(pokud je tedy povolen switchem)

//Adresace portu K-MOUSE a A-MOUSE:
//tlacitka = 64223 xxxxx0x0 11011111(na D4-D7
je 4-bitovy citac pro kolecko)
//osa X   = 64479 xxxxx0x1 11011111
//osa Y   = 65503 xxxxx1x1 11011111
//A-MOUSE / JOYSTICK = 31 00011111

//K-MOUSE: podpora ctyr tlacitek + kolecka
mysi

//port tlacitek c. 64223:
// D0-prave tlacitko
// D1-leve tlacitko
// D2-prostredni tlacitko
// D3-4.tlacitko
// D4-bit 0 citace kolecka
// D5-bit 1 citace kolecka
// D6-bit 2 citace kolecka
// D7-bit 3 citace kolecka
//((stisknute tlacitko = log.0, nestisknute =
log.1)

//A-MOUSE: port 31
//D0-XB / joy RIGHT
//D1-YB / joy LEFT
//D2-XA / joy DOWN
//D3-YA / joy UP
//D4-leve tlacitko / joy FIRE 1
//D5-prave tlacitko / joy FIRE 2
//D6-prostredni tlacitko / joy FIRE 3
//D7-ctrvrte tlacitko

////////////////////
////////////////////
//VSTUPNI PINY

DISWHEEL PIN 34;//vypne citac kolecka na
portu tlacitek(64223)

DISEXTRA PIN 19;//BLOKUJE EXTRA MOD(prepinac)
ENABLE31 PIN 20;//pravy switch zpřístupni
port 31(v tomto režimu je možné stiskem FIRE
//na joysticku odpojit emulaci

A-MOUSE a na portu 31 bude citelný KEMPSTON
JOY

//po stisku tlačítka myši se opět odpojí
JOYSTICK a bude citelná A-MOUSE
AMOUSE PIN 18;//levý switch
zpřístupní A-MOUSE na portu 31(pokud je port
povoleno).

//pokud není A-MOUSE přístupná, bude na portu
31 vždy hodnota 0.
XENABLE31 NODE ISTYPE 'REG';//trojstisk + LEFT
BUTTON - vyp/zap portu 31(nuluje port/emu-
lace A-MOUSE)

STISK NODE istype 'reg'; //reg.
detekce trojstisku - zapne extra mod a
tlacitka nejsou citelné
POVOL NODE istype 'reg'; //reg.
detekce uvolnění trojstisku
CLEAR NODE istype 'reg'; //reg.
nulující STISK a POVOL při opuštění extra modu
EXBUT NODE istype 'reg';

//registry s předchozím stavem clonek myši v
ose X
BXA NODE istype 'reg';
BXB NODE istype 'reg';
BX = [BXA,BXB];

//registry s předchozím stavem clonek myši v
ose Y
BYA NODE istype 'reg';
BYB NODE istype 'reg';
BY = [BYA,BYB];

//signály směru při pohybu myši
//((v další verzi použít k emulaci joysticku)
LEFT NODE istype 'COM';
RIGHT NODE istype 'COM';
UP NODE istype 'COM';
DOWN NODE istype 'COM';

//adresy z CPU
A0 PIN 8;
A1 PIN 6;
A2 PIN 4;
A3 PIN 2;
A4 PIN 39;
A5 PIN 40;
A6 PIN 42;
A7 PIN 43;
A8 PIN 38;

A10 PIN 37;

//ridici signály z CPU
IORQ PIN 1;
RD PIN 44;

//signály z A-MOUSE(PICu)
XA PIN 26;//joy
DOWN
XB PIN 24;//joy
RIGHT
YA PIN 22;//joy UP
YB PIN 25;//joy
LEFT
RIGHTB PIN 35;//joy FIRE 2
LEFTB PIN 33;//joy FIRE 1
MIDDLEB PIN 36;

FIRE4 PIN 27;//4.tlačítko myši
//FIRE5 PIN 34;//5.tlačítko myši

//nové vstupy pro clonky kolecka
KA PIN 29;
KB PIN 28;

KUP NODE istype 'COM';
KDOWN NODE istype 'COM';
KYA NODE istype 'REG';
KYB NODE istype 'REG';
KY = [KYA,KYB];

RKUP NODE ISTYPE 'REG';
RKDOWN NODE ISTYPE 'REG';
RK = [RKUP,RKDOWN];
ARKUP NODE ISTYPE 'REG';
ARKDOWN NODE ISTYPE 'REG';
ARK = [ARKUP,ARKDOWN];

K7..K0 NODE ISTYPE 'REG';//citac je ted jen
4-bitovy
KCOUNTER = [K7..K0];

//VYSTUPNI PINY
D7 PIN 14 istype
'com';
D6 PIN 9 istype
'com';
D5 PIN 7 istype
'com';
D4 PIN 3 istype
'com';
D3 PIN 5 istype
'com';

```

```

D2          PIN 11 istype      BX.clk = LEFT & !RIGHT # RIGHT & !LEFT;      & !RIGHTB;
'com';

D1          PIN 12 istype      BYA := YA;                                //clock pro citace os X a Y
'com';      BYB := YB;                                XCOUNTER.clk = LEFT & !RIGHT

D0          PIN 13 istype      BY.clk = UP & !DOWN # DOWN & !UP;                                #
'com';      RIGHT & !LEFT;
DATA      = [D7..D0];          //soucasny stisk trech tlacitek mysy aktivuje
                                extra mod
                                STISK.clk = !LEFTB & !MIDDLEB & !RIGHTB &
                                !STISK & !DISEXTRA;                                DOWN & !UP;
                                #
//INTERNI VYSTUPY
XQ9..XQ0 NODE istype 'reg';      STISK := 1;                                //citac pro osu X
XCOUNTER = [XQ9..XQ0];          //citac pro osu X      STISK.ar = CLEAR; //nuluje pri
X      opusteni extra modu
YQ9..YQ0 NODE istype 'reg';      //po uvolneni trojtisku se ceká na dalsi
YCOUNTER = [YQ9..YQ0];          //citac pro osu Y      stisk tlacitka
Y      POVOL.clk = LEFTB & MIDDLEB & RIGHTB &
OUTDATA NODE istype 'com';      STISK;
MSLOW      NODE ISTYPE 'REG';      POVOL := 1;
                                POVOL.ar = CLEAR; //nuluje pri
                                opusteni extra modu
                                #
//////////////////////////////////////                                //output enable pro datove signaly
                                //jsou pristupne pouze porty K-MOUSE a A-
EQUATIONS      //opusteni extra modu stiskem LB,MB nebo RB      MOUSE(31)
                                #
                                OUTDATA = !A10 & !A8 & A7 & A6 & !A5
//smery pohybu mysy      STISK & POVOL & !RIGHTB                                & A4 & A3 & A2 & A1 & A0 & !RD & !IORQ//port
LEFT = !BXA & !BXB & XA & !XB      #                                64223 - tlacitka(+vypinatelne kolecko)
                                # !A10 &
                                # BXA & !BXB & XA & XB      STISK & POVOL & !MIDDLEB;                                A8 & A7 & A6 & !A5 & A4 & A3 & A2 & A1 & A0 &
                                # BXA & BXB & !XA & XB      CLEAR := 1;                                !RD & !IORQ//port 64479 - osa X
                                # !BXA & BXB & !XA &      CLEAR.ar = !STISK & !POVOL;                                # A10 &
!XB;                                // trojtisk(extra mod) + prave tlacitko =
                                funkce prohozeni leveho a praveho tlacitka
RIGHT = !BXA & !BXB & !XA & XB      EXBUT.clk = STISK & POVOL & LEFTB & MIDDLEB                                //RD & !IORQ//port 65503 - osa Y
                                # !BXA & BXB & XA & XB                                // A10
                                # BXA & BXB & XA & !XB                                & !A8 & A7 & A6 & !A5 & A4 & A3 & A2 & A1 &
                                # BXA & !BXB & !XA &      EXBUT := !EXBUT;                                A0 & !RD & !IORQ//port 65247 - port odchylky
!XB;                                EXBUT.ar = STISK & POVOL & !LEFTB                                kolecka
                                & !MIDDLEB & !RIGHTB;                                # !ENABLE31 &
DOWN = !BYA & !BYB & YA & !YB      // trojtisk + leve tlacitko = funkce
                                ENABLE31(zap/vyp nulovani portu 31)                                #
                                # BYA & !BYB & YA & YB      XENABLE31.clk = STISK & POVOL & !LEFTB                                OUTDATA & !IORQ//eliminace zakmitu
                                # BYA & BYB & !YA & YB                                & MIDDLEB & RIGHTB;                                #
                                # !BYA & BYB & !YA &      XENABLE31 := !XENABLE31;                                OUTDATA & !RD//eliminace zakmitu
!YB;                                XENABLE31.ar = STISK & POVOL & !LEFTB                                DATA.OE = OUTDATA;
UP = !BYA & !BYB & !YA & YB      & !MIDDLEB & !RIGHTB;
                                # BYA & BYB & YA & YB
                                # BYA & BYB & YA & !YB
                                # BYA & !BYB & !YA &      // trojtisk + prostredni tlacitko = rychlost
!YB;      pohybu SLOW / TURBO 4x
                                MSLOW.CLK = STISK & POVOL & LEFTB & !MIDDLEB
                                & RIGHTB;
                                MSLOW := !MSLOW;
                                MSLOW.ar = STISK & POVOL & !LEFTB & !MIDDLEB
                                # !A10 & !A8 & A7 & RIGHTB
//uchova aktualni stav clonek mysy
BXA := XA;
BXB := XB;

```

```

& !EXBUT //defaultne je na D0 portu 64223
citelne prave tlacitko
# !A10 & !A8 & A7 & STISK//
blokuje cteni tlacitek v EXTRA MODU
# !A10 & A8 & A7 & XQ0 &
!MSLOW //data osy X(turbo tychlost 4x)
# A10 & A8 & A7 & YQ0 &
!MSLOW //data osy Y(turbo tychlost 4x)
# !A10 & A8 & A7 & XQ2 &
MSLOW //data osy X(mala citlivost pohybu)
# A10 & A8 & A7 & YQ2 &
MSLOW //data osy Y(mala citlivost pohybu)
# A10 & !A8 & A7 & K0//
odchylna osy kolecka
# !A7 & !XB & !AMOUSE & !XENABLE31
& !STISK; //AMIGA-MOUSE

D1 = !A10 & !A8 & A7 & RIGHTB &
EXBUT
# !A10 & !A8 & A7 & LEFTB &
!EXBUT
# !A10 & !A8 & A7 & STISK
# !A10 & A8 & A7 & XQ1 &
!MSLOW
# A10 & A8 & A7 & YQ1 &
!MSLOW
# !A10 & A8 & A7 & XQ3 &
MSLOW
# A10 & A8 & A7 & YQ3 &
MSLOW
# A10 & !A8 & A7 & K1
# !A7 & !YB & !AMOUSE & !XENABLE31
& !STISK;

D2 = !A10 & !A8 & A7 & MIDDLEB
# !A10 & !A8 & A7 & STISK
# !A10 & A8 & A7 & XQ2 &
!MSLOW
# A10 & A8 & A7 & YQ2 &
!MSLOW
# !A10 & A8 & A7 & XQ4 &
MSLOW
# A10 & A8 & A7 & YQ4 &
MSLOW
# A10 & !A8 & A7 & K2
# !A7 & !XA & !AMOUSE & !XENABLE31
& !STISK;

D3 = !A10 & !A8 & A7 & FIRE4
# !A10 & !A8 & A7 & STISK
# !A10 & A8 & A7 & XQ3 &
!MSLOW
# A10 & A8 & A7 & YQ3 &
!MSLOW
# !A10 & A8 & A7 & XQ5 &
MSLOW
# A10 & A8 & A7 & YQ5 &
!MSLOW
# !A7 & !YA & !AMOUSE & !XENABLE31
& !STISK;

D4 = !A10 & !A8 & A7 & (K0 # !DIS-
WHEEL)//na D4-D7 je 4-bitovy citac polohy
kolecka(zapojeni podle DON-NEWS19)
// # !A10 & !A8 & A7 & STISK
# !A10 & A8 & A7 & XQ4 &
!MSLOW
# A10 & A8 & A7 & YQ4 &
!MSLOW
# !A10 & A8 & A7 & XQ6 &
MSLOW
# A10 & A8 & A7 & YQ6 &
MSLOW
# A10 & !A8 & A7 & K4
# !A7 & !LEFTB & !AMOUSE & !XEN-
ABLE31 & !STISK;

D5 = !A10 & !A8 & A7 & (K1 # !DIS-
WHEEL)
# !A10 & !A8 & A7 & STISK
# !A10 & A8 & A7 & XQ5 &
!MSLOW
# A10 & A8 & A7 & YQ5 &
!MSLOW
# !A10 & A8 & A7 & XQ7 &
MSLOW
# A10 & A8 & A7 & YQ7 &
MSLOW
# A10 & !A8 & A7 & K5
# !A7 & !RIGHTB & !AMOUSE & !XEN-
ABLE31 & !STISK;

D6 = !A10 & !A8 & A7 & (K2 #
!DISWHEEL)
# !A10 & A8 & A7 & XQ6 &
!MSLOW
# A10 & A8 & A7 & YQ6 &
!MSLOW
# !A10 & A8 & A7 & XQ8 &
MSLOW
# A10 & A8 & A7 & YQ8 &
MSLOW
# A10 & !A8 & A7 & K6
# !A7 & !MIDDLEB & !AMOUSE &
!XENABLE31 & !STISK;

D7 = !A10 & !A8 & A7 & (K3 #
!DISWHEEL)
# !A10 & A8 & A7 & XQ7 &
!MSLOW
# A10 & A8 & A7 & YQ7 &
!MSLOW
# !A10 & A8 & A7 & XQ9 &
MSLOW
# A10 & A8 & A7 & YQ9 &
MSLOW
# A10 & !A8 & A7 & K7
# !A7 & !FIRE4 & !AMOUSE &
!XENABLE31 & !STISK; //citelne ctvrt tlacitko
pro A-MOUSE

//OBSLUHA KOLECKA

//KYA a KYB je pro uchovani predchoziho stavu
clonek kolecka
KYA := KA;
KYB := KB;
//a uklada se pri kazdem pohybu kolecka
KY.clk = KUP & !KDOWN
# KDOWN & !KUP;
//generovani smeru pohybu kolecka mysi
KUP = (!KYA & !KYB & KA & !KB
# KYA & !KYB & KA & KB
# KYA & KYB & !KA & KB
# !KYA & KYB & !KA &
!KB);
KDOWN = (!KYA & !KYB & !KA & KB
# !KYA & KYB & KA & KB
# KYA & KYB & KA & !KB
# KYA & !KYB & !KA &
!KB);
//hodiny citace pri pohybu kolecka
KCOUNT.CLK = KUP & !KDOWN
#
KDOWN & !KUP;
WHEN (KUP & !KDOWN)
THEN KCOUNTER := KCOUNTER-1
ELSE
WHEN (KDOWN & !KUP)
THEN KCOUNTER := KCOUNTER+1
END

```

Více o paralelním rozhraní i8255

Sweet Factory

O paralelním rozhraní 8255 se už v ZXM psalo, ale tentokrát to vezmeme trochu podrobněji z hlediska programátora.

Co má společného Didaktik Gama, D40/D80, Didaktik Kompakt, UR-4 a M/P? Nevíte? Přece obvod pro paralelní rozhraní 8255. Je to moc zajímavá věc (ještě zajímavější je obvod Z80 PIO) a proto si o tomto obvodu něco řekneme. Jak se nastavuje, jak se vysílá, jak se přijímá a taky nějaký ukázkový program

pro tisk na tiskárně.

8255 má tři osmibitové brány označeny A,B,C. Obvod je ke spektru je připojen na 4 porty: brána A na port 31, brána B na port 63, brána B na port 95 a řídicí povely na port 127. Obvod může pracovat ve třech režimech:

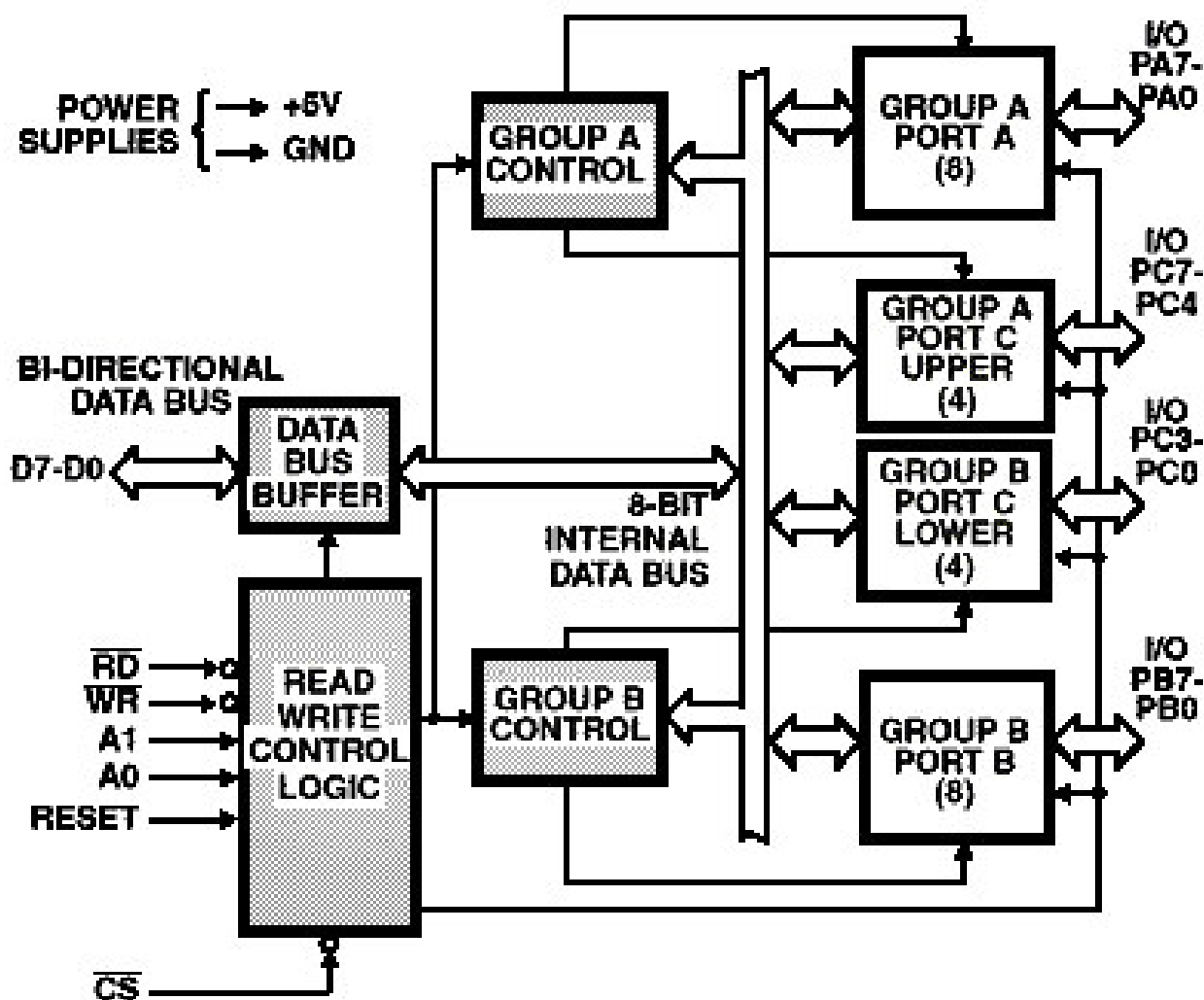
mód 0:

brána A může být celá vstupní nebo výstupní, brána B taky a brána C může být takto rozdělena na 2 poloviny (bity 0-

3,4-7). Pokud je brána výstupní, je zapsaná hodnota na ní stále čitelná až do nastavení hodnoty jiné. Tzn. když OUTneme nějakou hodnotu, zařízení si ji může přečíst i později. Naopak to ale neplatí. Pokud zařízení nastaví na bráně nějakou hodnotu (třeba Kempston joystick), hodnotu lze číst právě v ten moment, když jej zařízení nastaví (a drží).

mód 1:

V tomto módu pracují brány A nebo B (podle nastavení) specifickým způsobem. Brána



C má pak některé bity využity na řídicí signály. Ostatní bity je možno využít standardním způsobem jako v módu 0. Tento mód se taky nazývá "handshake". Nejprve je nutno zkontrolovat, zda je zařízení schopno přijímat data (signál INTR je nastaven na 0 - na bráně C), pak je možno poslat hodnotu na zvolenou bránu a obvod sám vygeneruje signál STROBE, který sdělí zařízení, že si může hodnotu převzít. Jakmile se stane i toto, zařízení potvrdí příjem posláním impulsu do signálu ACK (acknowledge - poděkování) a obvod po tomto signálu nastaví stavový bit INTR na 1 (může se využít pro generování přerušování - na ZX nepoužito). Pak je možno vysílat další hodnoty. Příjem dat z nějakého zařízení, je v tomto módu stejný. Zařízení si zkontroluje, zda je 8255 připravena, pak vyšle hodnotu na bránu a taky impuls STROBE. Data se zapíše do paměti v 8255 a sám obvod vygeneruje signál ACK. Aby bylo možno přijímat další data, je nutné přijatá data přečíst (in a, (port)).

mód 2:

Ten sám vyzkoušený nemám, takže jen stručně. V tomto módu pracuje jen brána A, ostatní musí pracovat v módu 0, event. v módu 1 (pouze B). Tento mód by se dal nazvat obousměrným přenosem v módu 1. Při zapísání dat na bránu A se nejdříve přijme byte z periférie a poté jí pošle hodnotu zapsanou na bránu A. Signály se aktivují ve stejném sledu jako v módu 1. Signály STROBE a ACK jsou negovány.

Nyní víme, jak můžeme 8255 využít, takže si vysvětlíme, jak se dá takový mód nastavit a jak se využívá právě řídicí port 127. Řídicí port můžeme rozdělit na

dva nastavovací povely: nastavení jednotlivých bitů brány C a nastavení celého obvodu 8255. Při nastavování jednotlivých bitů C (jsou tam většinou řídicí signály periférie) se držíme následující zásady:

bit 7 = 0

bit 1-3 = číslo nastavovaného bitu

bit 0 = hodnota bitu

Například chceme nastavit bit 4 brány C do 1: bit 0 bude tedy 1 a na bitech 1-3 bude %100 (4). Celé slovo bude takto: %00001001 (9). Pak tedy pošleme tento příkaz: OUT 127,9.

Nastavit celý obvod 8255 se dá s nastaveným 7. bitem a to podle následující tabulky:

bit 7: musí být 1

bit 6,5: mód pro bránu A: 00 = mód 0

01 = mód 1

10 = mód 2

bit 4: směr pro bránu A: 1 = vstup, 0 = výstup

bit 3: směr pro horní polovinu brány C: 1 = vstup, 0 = výstup

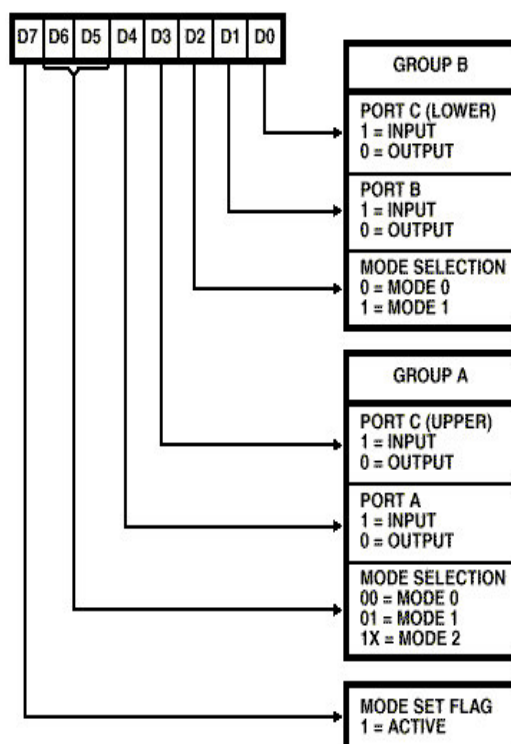
bit 2: mód pro bránu B: 0 = mód 0, 1 = mód 1.

bit 1: směr pro bránu B: 1 = vstup, 0 = výstup

bit 0: směr pro dolní polovinu portu C: 1 = vstup, 0 = výstup

Příklad: máme zapojený

Kempston joystick na bráně A a



tiskárnu EPSON LX400 na bráně B. na PC7 je zapojen signál BUSY a na PC3 je zapojen signál STROBE. Takže brána A musí být vstupní, B výstupní, horní polovina C vstupní a dolní výstupní. Bit 7 bude vždy 1, bity 5,6 budou určovat mód 0 (tedy 00), brána A je vstup, tedy 4. bit bude 1, horní polovina C vstupní (bit 3:1), mód brány B bude 0, takže bit 2=0, směr pak výstupní (lezou po ní data ven) = bit 1:0 a konečně dolní polovina výstupní, tedy bit = 0. Dohromady: %10011000 = 152. Takže nastavení, aby fungoval joystick i tiskárna najednou je příkaz: OUT 127,152.

Pokud je brána A nebo B nastavena do režimu 1, bity brány C mají specifický význam:

bit	pro výstup	pro vstup
0	INTR-B	INTR-B
1	STROBE-B	ACK-B
2	ACK-B	STROBE-B
3	INTR-A	INTR-A
4	nepoužit	STROBE-A
5	nepoužit	ACK-A
6	ACK-A	nepoužit
7	STROBE-A	nepoužit

Jestli obvod jede v módu 2, tak význam jednotlivých bitů brány C je následující:

bit	význam
0-2	nevyužity
3	INTR
4	STROBE-IN
5	ACK-OUT
6	ACK-IN
7	STROBE-OUT

Periférie: Nejznámější periférie je Kempston Joystick, občas se taky připojují tiskárny a zapisovače. Někdo s 8255 ovládá modelovou železnici, někdo má připojenou jen Amiga myš. Zkusíme se podívat na to, jak se dá, ze strojového kódu, ovládat taková periférie.

Začneme joystickem. Kempston joystick je většinou připojen na bránu A a spíná log.1 na bity 0-5 v tomto pořadí: doleva, doprava, dolů, nahoru a střelba. Zkuste třeba následující krátký program v Basicu:

```
10 PRINT AT 10,10;IN 31;"  ":
GOTO 10
```

Pak zkuste verglovat joystickem a uvidíte čísla, která když si pak převedete do dvojkové soustavy odpovídají bitům, která jsem napsal. Třeba když budete držet FIRE, na obrazovce uvidíte 16, ve dvojkové soustavě je to: %10000 a to je právě bit 5. Rozepsáno do strojáku, rozeskok v nějakém programu by mohl vypadat asi takto:

```
port      in  a,(31)      ;načti
           and  %11111     ;ponech
           rra              ;rotuj
doprava, nižší bajty vstupují
do CARRY
           jr   c,DOLEVA   ;první
na řadě je vlevo
           rra              ;znovu
rotujeme, tedy do carry vs-
toupí bit 1.
           jr   c,DOPRAVA ;a to je
doprava
           rra              ;opět
           jr   c,DOLU
           rra
           jr   c,NAHORU
           rra
           jr   c,STRELBA
```

Ted' je na řadě tiskárna.

Necháme polemik, zda je lepší to, či ono zapojení, ale zkusíme si dvě zapojení naprogramovat. Jedno se jmenuje "Special Didaktik" navržené kdesi ve Skalici a druhé bude Desktopem označované "Strobed port B". Nejprve zapojení prvního: data: brána A, STROBE: 3.bit brány C, BUSY: 7.bit brány C. Ted' už

víme, jak spočítat inicializaci obvodu, tedy dojdeme k číslu %10001000 (136). Ještě než začneme tisknout, je třeba obvod inicializovat:

```
ld  a,136
out (127),a
```

Special Didaktik pracuje tedy v módu 0, takže budeme muset sami generovat signál STROBE. Napíši zde krátký assembler, jak by mohl vypadat prográmeček na posílání jednoho znaku do tiskárny:

```
A_OUT
;vstupní znak je v registru a
           push af
;ulož znak
WAIT      in  a,(95)
;přečti bránu C
           bit  7,a
;testuj BUSY
           jr   nz,WAIT
;tiskárna je busy, opakuj
           pop  af
;obnov znak
           out  (31),a
;pošli jej do tiskárny

           ;nyní pošleme impuls STROBE
           ld  a,%00000111
;nastav jedničku do bitu 3
           out  (127),a
;na bráně C
           nop
;chvilku
           nop
;počkej
           ld  a,%00000110
;nastav nulu do bitu 3
           out  (127),a
;pošli do brány C
           ret
vrať se
```

Druhé zapojení o něco jednodušší, protože o generování signálu STROBE se stará sama 8255ka, ale hrozí to, že signál může být jiný (krátký/dlouhý) než

jaký jej tiskárna požaduje. Zapojení drátků je následující: data: brána B, STROBE: 1.bit brány C, ACK: 2.bit brány C. Inicializační slovo je tedy: %10010100 (148). Vzhledem k tomu, že signál STROBE nemusíme generovat, můžeme tiskárnu ovládat jednoduše i z Basicu. Třeba odeslání znaku "A" stačí toto: OUT 63,CODE "A". Nediňte se ale, že tiskárna jej rovnou nevytiskne. Vytiskne až celý řádek, a ten ukončíme: OUT 63,13. Možná bude třeba posunout papír v tiskárně na další řádek: OUT 63,10. Vidíte, jednoduché. Zkusíme assembler:

```
A_OUT
;vstupní znak bude v reg. A
           push af
;ulož znak
WAIT      in  a,(95)
;přečti bránu C
           and  %1
;přečti stav bitu 0.
           jr   z,WAIT
;pokud není nahozen, čekej
           pop  af
;obnov znak
           out  (63),a
;pošli znak a o další se ne-
starej
           ret
;vrať se
```

Ještě jedna důležitá věc.

Pokud je brána nastavena jako výstupní, je možné do ní zapísané hodnoty zpětně přečíst. Tedy když dám OUT (31),255, tak pro PRINT IN 31 bych měl dostat hodnotu 255. To je docela vhodné k tomu, pokud potřebujeme poslat na určitý bit brány nějakou hodnotu a nechceme s ostatními hýbat, uděláme jednoduše IN port, AND ponechane_bity, OR nova_hodnota, OUT port.

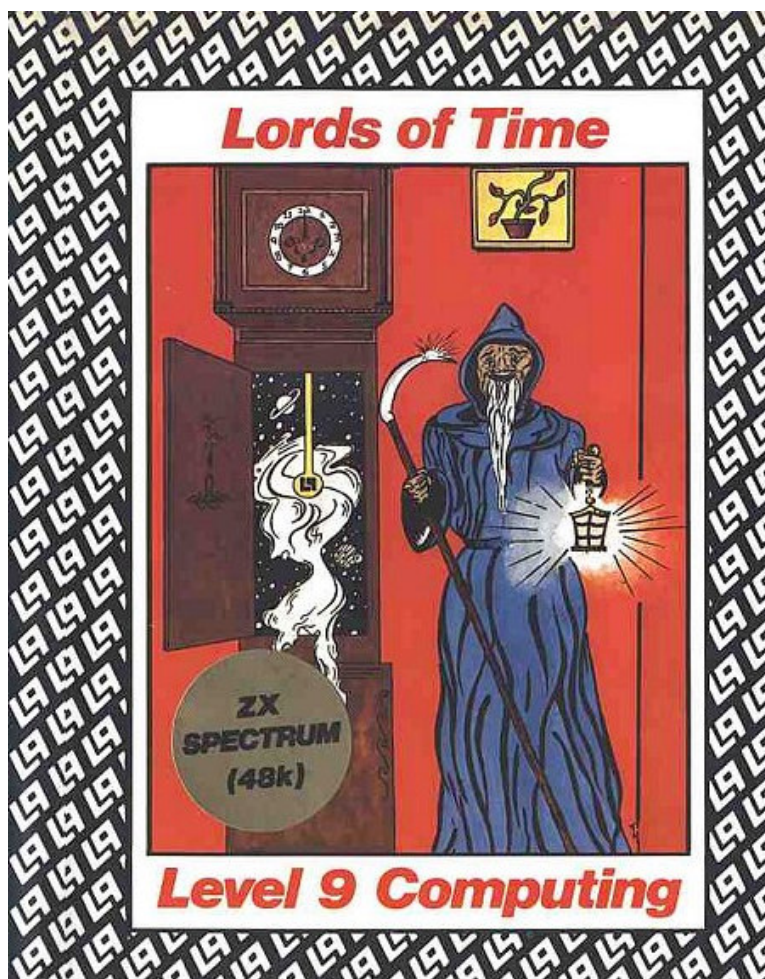
Lords of Time

(Level 9)

Ivan Preclík

Jak jsem minule slíbil, uvádím zde svůj návod na textovou hru od legendárního tvůrčího týmu Level 9. Tentokrát jsem si vyzkoušel a zahrál pro mne docela neznámou hru Páni času. Jak název hry napovídá, stáváte se cestovatelem v čase. Hra vás provede snad všemi časovými úseky od nejvzdálenější minulosti přes současnost až po budoucnost. Vzhůru tedy za dobrodružstvím cestovatele, který se musí vypořádat s mnoha útrapami a nástrahami na své nekonečné cestě. V průběhu hraní celé hry vás doprovází vámi dosažené skóre. Hru nedohrajete dokud nedosáhnete 1000 bodů z 1000 možných.

Ve hře nepůjde o nic menšího, než zachránit svět. Příběh se má zhruba takto. Sám velký Otec čas, jenž stvořil náš svět a snad i celý vesmír, tě žádá o tvou pomoc. Jeho devět služebníků pánů času se mu vymklo kontrole a začínají si dělat, co se jim zachce. Každý z pánů času měl odděleně dohlížet nad svým časovým obdobím. Páni času se však spojili, narušili strukturu času a začali mísit minulost s budoucností a přítomností. Na Zemi pak zavládne absolutní chaos vedoucí k definitivnímu zániku našeho světa. Musíš projít všemi devíti časovými obdobími a každého z pánů času porazit. Nakonec budeš muset vyléčit



narušenou časovou strukturu léčivým lektvarem, k němuž budeš potřebovat přísady nasbírané ve všech časových obdobích. Vzhůru do boje! Ó, proboha, spaste své duše! Nukleární mesiáš přichází!

Na začátku hry se nacházíš ve svém obývacím pokoji. Rozhlédni se (LOOK), prohlédni si obrázek (EXAMINE PICTURE). V tomto okamžiku obraz náhle obživne a zjeví se ti Otec čas, který ti vysvětluje tvé poslání. Vezmi přesýpací hodiny (TAKE HOURGLASS), jdi na sever (N), vezmi svíčku (TAKE CANDLE), vezmi zápalky (TAKE MATCHES), zapal svíčku (LIGHT CANDLE), prozkoumej hodiny (EXAMINE CLOCK),

natáhni hodiny (WIND CLOCK). Nyní se otevřou dveře. Vstup dovnitř (IN). Teď se nacházíš ve stroji času, tj. ve střední části dobrodružství. Otoč zubem na pozici 1 (TURN COG 1). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 1. Rozhoupej kyvadlo (SWING PENDULUM), jdi na sever (N). Nyní se ocitáš na štěrkové příjezdové cestě.

Časová zóna 1:
Současnost

Jdi na západ (W), jdi na západ (W), otevři dveře (OPEN DOOR), otevři dveře (OPEN DOOR), jdi na západ (W), jdi nahoru (UP), jdi na sever (N), vezmi batoh (TAKE RUCKSACK), nos batoh (WEAR RUCKSACK). Od této chvíle uneseš více předmětů, zvětšila se totiž kapacita tvého úložného prostoru. Vezmi lano (TAKE ROPE), prozkoumej smetl (EXAMINE RUBBISH), vezmi otvírák na konzervy (TAKE TIN OPENER), jdi na jih (S), jdi na západ (W), vezmi šperkovnici (TAKE JEWELLERY CASE), jdi na východ (E), jdi dolů (DOWN), vezmi kopí (TAKE SPEAR), otevři dveře (OPEN DOOR), jdi na východ (E), jdi na východ (E), jdi na sever (N), jdi na sever (N), vyber si (TAKE PICK), vezmi prkna (TAKE PLANKS), jdi na jih (S), jdi na jih (S), jdi na západ (W), jdi na západ (W), otevři dveře (OPEN DOOR), jdi na západ (W), jdi na jih (S), otevři kredenc

(OPEN CUPBOARD), vezmi kočičí žrádlo (TAKE CATFOOD), jdi na západ (W), vezmi zrcadlo (TAKE LOOKING GLASS), vezmi metronom (TAKE METRONOME), otevři dveře (OPEN DOOR), jdi na západ (W), jdi na severozápad (NW), vezmi Kozlíka Lékařského (TAKE VALERIAN), jdi na jihovýchod (SE), jdi na západ (W), jdi nahoru (UP), jdi na jih (S), máchni Kozlíkem Lékařským (WAVE VALERIAN). V tomto okamžiku se ocitáš u stromu vrby, který se s tebou dá do řeči a vysvětluje ti, proč chce zemřít. Ve skutečnosti to však pro tebe znamená, že vrba musí být pokácena, abys mohl získat ukrytý poklad. Jdi na sever (N), jdi na sever (N), svaž prkna dohromady k sobě (TIE PLANKS TOGETHER), polož prkna (DROP PLANKS). Tímto se vytvoří přes potok most. Jdi na západ (W), prozkoumej Narcis (EXAMINE NARCISSUS). Jak jste si všimli, tak rostliny jsou v této hře živé bytosti, s kterými můžete rozmlouvat. Dej rostlině Narcis zrcadlo (GIVE LOOKING GLASS). A tímto jsi získal magnetický kámen čili magnetovec. Jdi přes (ACROSS), jdi na jih (S), jdi nahoru (UP), jdi na východ (E), jdi na jihozápad (SW), máchni magnetovcem (WAVE LODESTONE), odemkni dveře (UNLOCK DOOR), vstup dovnitř (IN), vezmi sekeru (TAKE AXE), vezmi lopatu (TAKE SHOVEL), otevři dveře (OPEN DOOR), vyjdi ven (OUT), jdi na severovýchod (NE), jdi na západ (W), jdi nahoru (UP), jdi na jih (S), posekej vrbu (CHOP WILLOW), vezmi slzu (TAKE TEARDROP), polož sekeru (DROP AXE), jdi na sever (N), jdi nahoru (UP), jdi na východ (E), jdi na jihozápad (SW), jdi na východ (E), jdi na východ (E), prozkoumej kompost (EXAMINE COMPOST). Odhalí se ti stříbrná mince, kterou ale ještě nemůžeš vzít. Prozkoumej kompost (EXAMINE COMPOST). Teď se ti odhalí houbový prstenec, který se později použije ke svolání tzv. zubní víly, která v noci vymě uje dětem vytržený zub za peníze. Jdi na východ (E), prozkoumej auto Porsche (EXAMINE PORSCHE), vezmi mu benzín (TAKE PETROL), jdi

dolů (DOWN). Nyní jsi zpátky uvnitř stroje času. Polož magnetovec (DROP LODESTONE), polož šperkovnici (DROP JEWELLERY CASE), polož slzu (DROP TEARDROP), polož přesýpací hodiny (DROP HOURGLASS), polož metronom (DROP METRONOME), polož Kozlíka Lékařského (DROP VALERIAN). Otoč zubem na pozici 2 (TURN COG 2). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 2. Rozhoupej kyvadlo (SWING PENDULUM), jdi na sever (N). Nyní stojíš na roztržitém ledu v hlubokém údolí.

Časová zóna 2: Doba ledová

Jdi na sever (N), jdi na sever (N), jdi na sever (N), vypusť benzín (POUR PETROL), zapal zápalku (LIGHT MATCH), zapal benzín (LIGHT PETROL). Tímto les začne prudce hořet a Mamut strachy vyskočí ze své kůže. Vezmi kabát (TAKE COAT), obleč si kabát (WEAR COAT), vezmi kel (TAKE TUSK), jdi na východ (E), jdi nahoru (UP), jdi nahoru (UP), vezmi kost (TAKE BONE), jdi nahoru (UP). Nyní vidíš hladového tygra, kterého bolí zuby. Otevři konzervu s masem (OPEN TIN), dej mu masovou konzervu (GIVE TIN), vytáhni mu zub (TAKE TOOTH). Tygr šťastně odcválá pryč. Vstup dovnitř (IN), jdi na jih (S), zakřič (SHOUT), vezmi rampouch (TAKE ICICLE), jdi na sever (N), jdi na východ (E), jdi na východ (E). Tady potkáš stříbrnou lišku. Dej ji kost (GIVE BONE), jdi na východ (E), rozbij zeď (BREAK WALL), jdi na jih (S), vezmi mnohoramenný svícen (TAKE CANDELABRA), jdi na sever (N), jdi na západ (W), jdi na jih (S), polož rampouch (DROP ICICLE). Jezero, které je před tebou náhle zamrzne a to ti umožní ho přejít. Jdi na severozápad (NW), jdi na severozápad (NW), rozbij ledovou krychli (BREAK CUBE). Tímto je sněhová královna vysvobozená a dostáváš od ní meč. Jdi na jih (S), vstup dovnitř (IN), jdi na západ (W), jdi dolů (DOWN), jdi dolů (DOWN). Nyní jsi zpátky uvnitř stroje času. Polož kel (DROP TUSK), polož mnohoramenný

svícen (DROP CANDELABRA), polož meč (DROP SWORD). Otoč zubem na pozici 1 (TURN COG 1). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 1. Rozhoupej kyvadlo (SWING PENDULUM), jdi na sever (N). Nyní se ocitáš na šterkové příjezdové cestě.

Časová zóna 1: Současnost – Druhá návštěva

Jdi na západ (W), jdi na západ (W), otevři dveře (OPEN DOOR), jdi na západ (W), jdi na západ (W), otevři dveře (OPEN DOOR), jdi na západ (W), jdi na jihozápad (SW), jdi na východ (E), jdi na východ (E), sněž houbu (EAT MUSHROOM), rozhlédni se (LOOK). Objeví se zubní víla. Dej ji zub (GIVE TOOTH). Zubní víla ti za něj na oplátku nabízí stříbrnou minci. Odmítni to (NO). Zubní víla zvyšuje svou nabídku a přidá ti ještě světlušku. Přijmi (YES), jdi na východ (E), jdi dolů (DOWN). Nyní jsi zpátky uvnitř stroje času. Polož kopí (DROP SPEAR), polož stříbrnou minci (DROP SILVER COIN). Toto byl jediný úsek hry, kdy jste se museli vracet v čase. Další časové zóny hry již budou následovat v chronologickém pořadí. Otoč zubem na pozici 3 (TURN COG 3). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 3. Rozhoupej kyvadlo (SWING PENDULUM), jdi na sever (N). Ocitáš se pod malým kopcem.

Časová zóna 3: Doba kamenná

Jdi na západ (W), jdi na západ (W). Tady potkáš nebezpečného Allosaura. Jdi na východ (E). Allosaurus tě pronásleduje. Jdi na východ (E), jdi na východ (E), jdi na východ (E). Tady Allosaurus začne bojovat s Tyranosaurusem Rexem a ty ses ho tak zbavil, takže pokračuješ dál sám. Jdi na západ (W), jdi na západ (W), jdi na západ (W), jdi na západ (W), jdi na západ (W), vezmi listí (TAKE LEAVES), vezmi zrcadlo (TAKE MIRROR), máchni listím (WAVE LEAVES). Toho si všimne

Brontosaurus a ch apne po listech a tím tě zvedne nahoru ven. Jdi na východ (E). Brontosaurus ti teď blokuje cestu a ty nemůžeš jít dál. Máchni zrcadlem (WAVE MIRROR). Toto je ovšem kouzelné zrcadlo, které Brontosaura vcucne do sebe. Jdi na východ (E), jdi nahoru (UP), vstup dovnitř (IN), polož zrcadlo (DROP MIRROR). Zrcadlo se tímto roztříští a z kousků skla se znova vynoří Brontosaurus, který začne honit jeskynní lidi. Jdi na sever (N), vezmi vejce (TAKE EGG), jdi na sever (N), vezmi kamenný hrnec (TAKE STONE POT), jdi na sever (N), vezmi oblázek (TAKE PEBBLE), jdi na východ (E), jdi na jihovýchod (SE), vezmi kamenný kyj (TAKE STONE CLUB), jdi na východ (E), hod' oblázkem (THROW PEBBLE), jdi na východ (E), vezmi figurínu (TAKE FIGURINE), jdi na západ (W), jdi na jihovýchod (SE). Teď vidíš jeskynního muže, jak táhne jeskynní ženu za její vlasy. Hod' kyjem (THROW CLUB). Kyj trefí jeskynního muže a srazí ho z mostu. Zachráněná žena ti pak dá hroudu zlata. Jdi na jih (S), řekni heuréka (SAY EUREKA), vstup dovnitř (IN). Jsi v jeskyni, kde vidíš něčí podpis – JWAAMTETS. Je jasné, že je to zašifrované a správným přehozením písmen ti vyjde toto – JAMES WATT. Řekni JAMES WATT (SAY JAMES WATT), vstup dovnitř (IN), vezmi kolo (TAKE WHEEL), jdi dolů (DOWN). Nyní jsou opět zpět uvnitř stroje času. Polož figurínu (DROP FIGURINE), polož hroudu zlata (DROP NUGGET), polož vejce (DROP EGG), polož kamenný hrnec (DROP STONE POT), polož otvírák na konzervy (DROP TIN OPENER). Otoč zubem na pozici 4 (TURN COG 4). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 4. Rozhoupej kyvadlo (SWING PENDULUM), jdi na sever (N). Ocítáš se na pláži.

Časové pásmo 4: Vikingské období

Jdi na sever (N). Vidíš vikingskou stráž, která se třese zimou. Dej jim kožich (GIVE FUR COAT). Stráž ti za to dává

dlouhý roh na svolávání dobytka. Jdi na jih (S), jdi na jihovýchod (SE), jdi na východ (E), jdi na východ (E), kopej (DIG). Teď ses ocitl v díře. Znovu kopej (DIG). Odkryl jsi kus pergamentu. Vezmi pergamen (TAKE PARCHMENT), jdi na západ (W), přečti si pergamen (READ PARCHMENT). Jsou na něm veršované rady o tvých dalších krocích. Jdi na jihozápad (SW), odstrč si kámen (PUSH STONE), jdi na západ (W), jdi na západ (W), jdi na západ (W), jdi na západ (W), jdi nahoru (UP), jdi nahoru (UP), polož dlouhý roh na svolávání dobytka (DROP LUR). Totiž někde blízko v okolí se potlouká pirát Pete a tímto zabráníte, aby tento dlouhý roh neukradl. Jdi na východ (E), otevři bednu (OPEN CHEST). Pirát Pete k vám náhle přiskočí a ukradne jeden z vašich majetků. Jdi na západ (W), vezmi dlouhý roh na svolávání dobytka (TAKE LUR). Pirát Pete opět k vám přiskočí a ukradne něco dalšího, ale jen zřídká by vám vzal dlouhý roh. Zatrub na dlouhý roh (PLAY LUR). Tu náhle přijedou Vikingové a odvedou si piráta Peta. Tobě pak dají za odměnu olivovou větévku. Vlez do bedny (IN CHEST), vezmi nějaký předmět (TAKE). Tímto předmětem je myšleno cokoli, co vám bylo ukradeno. Tady si za příkaz vzít dosadíte sami, co je třeba. Zatáhni za kliku (PULL HANDLE). Podlaha se pod tebou odsune pryč a ty jsi zpátky uvnitř stroje času. Polož dlouhý roh na svolávání dobytka (DROP LUR), polož olivovou větévku (DROP OLIVE BRANCH), polož pergamen (DROP PARCHMENT). Vezmi valoun (TAKE NUGGET), vezmi meč (TAKE SWORD), otoč zubem na pozici 5 (TURN COG 5). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 5. Rozhoupej kyvadlo (SWING PENDULUM), jdi na sever (N). Ocítáš se na konci nějaké cesty.

Časové pásmo 5: Starověk

Jdi na sever (N), prozkoumej milník (EXAMINE MILESTONE). Je na něm napsáno – Vesnice naděje. Tento

milník budeš potřebovat, až budeš v časové zóně 9! Vezmi milník (TAKE MILESTONE), jdi na východ (E), otevři dveře (OPEN DOOR), vstup dovnitř (IN), vezmi světlé anglické pivo (TAKE ALE). Barman ti však řekne, že za něho musíš zaplatit. Dej mu valoun (GIVE NUGGET), vezmi světlé anglické pivo (TAKE ALE), otevři dveře (OPEN DOOR), vyjdi ven (OUT), jdi na východ (E), jdi na východ (E), jdi na východ (E). Tady se setkáš s královským poslem, který je velmi žíznivý. Dej mu světlé anglické pivo (GIVE ALE). On ti na oplátku dává nějaké jídlo. Prohlédni si jídlo (EXAMINE FOOD). Je to něco ostrého. Jdi na západ (W), jdi na západ (W), jdi na sever (N), jdi na východ (E), jdi na východ (E). Tady dojdeš k drakovi. Dej mu jídlo (GIVE FOOD). Drak ho sní a zeptá se tě, jestli nechceš na něm odletět pryč. Přijmi to (YES). Vy a drak přeletíte přes příkop a přistanete na jižním svahu. Vezmi křídlo (TAKE WING), vezmi žábu (TAKE FROG), polib žábu (KISS FROG). Žába se promění v šikovného prince, kterého budeš od této chvíle poslouchat. Jdi na jih (S), vstup dovnitř (IN), vezmi železnou rukavici (TAKE GAUNTLET), jdi na jihovýchod (SE), jdi na západ (W), vezmi brnění (TAKE ARMOUR), obleč si brnění (WEAR ARMOUR), jdi na jih (S). Tady vidíš černého rytíře. Jdi na západ (W). Černý rytíř tasí svůj meč. Jdi na západ (W). Černý rytíř na tebe útočí, ale princ ti přispěchá na pomoc a zabije ho, potom odstraní jeho tělo. Jdi na západ (W), jdi dolů (DOWN), jdi dolů (DOWN), kopej (DIG). Dostáváš se tak do jámy. Kopej (DIG). Tady odkrýváš ze země klenotnickou truhlici. Vezmi truhlici (TAKE COFFER), kopej (DIG). Prokopál ses zpátky do stroje času. Polož truhlici (DROP COFFER), polož brnění (DROP ARMOUR), polož železnou rukavici (DROP GAUNTLET), polož křídlo (DROP WING), polož milník (DROP MILESTONE). Otoč zubem na pozici 6 (TURN COG 6). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 6. Rozhoupej kyvadlo

(SWING PENDULUM), jdi na sever (N). Nyní se nacházíš ve vstupní hale.

Časové pásmo 6: Středověk

Jdi na sever (N), jdi na západ (W), jdi na jih (S), vezmi roh na pití (TAKE DRINKING HORN), vezmi bonbóny (TAKE SWEETMEATS), jdi na sever (N), jdi na východ (E), jdi na východ (E), jdi na jih (S), vezmi hrací karty (TAKE PLAYING CARDS), prober karty (SHUFFLE CARDS). Mezi kartami najdeš žolíka. Vezmi žolíka (TAKE JOKER), vezmi zvoneček (TAKE BELL), zazvo zvonečkem (RING BELL). Tu se náhle objeví šašek. Dej mu žolíka (GIVE JOKER). Šašek ti za to nasadí na tvoji hlavu jeho šaškovskou čepici. Jdi na sever (N), jdi na západ (W), jdi nahoru (UP), jdi nahoru (UP). Tady sis všiml, že obkladové desky jsou duté, ale prozatím s tím nemůžeš nic dělat. Jdi na sever (N), jdi na západ (W), jdi na jih (S), vezmi kobereček (TAKE RUG), jdi na jih (S). Tady tě uvidí hlídací psi, kteří začnou štěkat a pravděpodobně vzbudí svého majitele. Dej jim bonbóny (GIVE SWEETMEATS). Psi je sežerou a usnou po nich. Otevři pohovku (OPEN OTTOMAN), prozkoumej pohovku (EXAMINE OTTOMAN), vezmi klenotní korunu (TAKE JEWELLED CROWN), jdi na sever (N), jdi na sever (N), jdi na východ (E), jdi na východ (E), jdi na jih (S), jdi na jih (S), vezmi loutnu (TAKE LUTE), hraj na loutnu (PLAY LUTE). Teď slyšíš strašidelné kroky, jak se plahočí po krátkých schodech a pak pozvolna mizí. Jdi na sever (N), jdi na sever (N), jdi na západ (W), jdi na jih (S), hraj na loutnu (PLAY LUTE). Obkladové desky se dají do pohybu a odkryjí tajemné dveře. Jdi na jih (E). Právě jsi vstoupil do bludiště z živých plotů. Jdi na západ (W), jdi na západ (W), jdi na jihozápad (SW), jdi na východ (E), jdi na sever (N), jdi na jihovýchod (SE), vstup dovnitř (IN), napij se vody (DRINK WATER). Tato voda tě udělala natolik silným, abys mohl později odstranit mřížku v římských lázních. Jdi na sever

(N). Opět jsi zase zpět uvnitř stroje času. Polož roh na pití (DROP DRINKING HORN), polož zvoneček (DROP BELL), polož karty (DROP CARDS), polož čepici (DROP CAP), polož kobereček (DROP RUG), polož klenotní korunu (DROP JEWELLED CROWN), polož loutnu (DROP LUTE). Vezmi železnou rukavici (TAKE GAUNTLET), vezmi stříbrnou minci (TAKE SILVER COIN), Otoč zubem na pozici 7 (TURN COG 7). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 7. Rozhoupej kyvadlo (SWING PENDULUM), jdi na sever (N). Nyní jsi na místě zvaném Rocky Plain.

Časové pásmo 7: Budoucnost

Jdi na západ (W), jdi na západ (W). Tady potkávaš androida. Dej mu stříbrnou minci (GIVE SILVER COIN). Android ti za to dává galaktický peníz, který tě oprav uje strávit den ve sportovní aréně. Jdi na východ (E), jdi na sever (N), jdi nahoru (UP), jdi nahoru (UP). Zde musíš vymyslet, jak se brat kosmickou hvězdou. Nasaď si železnou rukavici (WEAR GAUNTLET), protože hvězda je rozžhavená doběla. Vezmi hvězdu (TAKE STAR), jdi dolů (DOWN), jdi dolů (DOWN), jdi na jih (S), jdi na východ (E), jdi na východ (E), jdi na sever (N), jdi na sever (N), otevři dveře (OPEN DOOR), jdi na sever (N), jdi na západ (W). V tomto okamžiku vstupte do plaveckého bazénu a ihned si zchladte vaši hvězdu dřív, než propálí vaši železnou rukavici. Po této ochlazovací proceduře z hvězdy zůstane smaragd. Vezmi smaragd (TAKE EMERALD), jdi na východ (E), jdi na sever (N), vezmi matraci (TAKE MATTRESS), vezmi raketu (TAKE ROCKET), jdi na jih (S), otevři dveře (OPEN DOOR), jdi na jih (S), jdi na jih (S), jdi na jih (S), jdi na východ (E), skoč dolů (DOWN). Matrace ztlumí tvůj pád. Jdi na jih (S), jdi na východ (E), světelnou šavli (TAKE LIGHTSABRE), jdi na západ (W), jdi na jih (S), otevři dveře (OPEN DOOR), jdi na západ (W), vezmi lahvičku (TAKE PHIAL). Jen malou poznámku. Tuto

lahvičku nelze kdekoliv odložit, protože se může snadno roztříštit. Zapal raketu (FIRE ROCKET). Raketa začne stoupat a vy šplháte po laně, které je k raketě připevněné. Jdi na sever (N). Najednou jsi zastaven stráží. Dej ji galaktický peníz (GIVE GROAT). Nyní máš přístup na jeden den do sportovní arény. Jdi na sever (N), boj se cybermanem (FIGHT CYBERMAN), co se ti připleťl do cesty. Jdi na východ (E), prozkoumej pracovní stůl (EXAMINE WORKBENCH), vezmi šroubovák (TAKE SCREWDRIVER), jdi na západ (W). Zde vidíš nějaké rozbité roboty, ale ještě se jich v žádném případě nedotýkej. Jdi na jih (S), vstup dovnitř (IN), skoč dolů (DOWN). Matrace ztlumí tvůj pád. Vezmi rubín (TAKE RUBY), jdi nahoru (UP), vylez ven (OUT), jdi na sever (N), otevři robota (OPEN ROBOT), prozkoumej robota (EXAMINE ROBOT), vezmi křemíkový čip (TAKE SILICON CHIP), jdi na jih (S), vstup dovnitř (IN), skoč dolů (DOWN). Matrace opět ztlumí tvůj pád. Jdi dolů (DOWN). Jsi zase zpět uvnitř stroje času. Polož rubín (DROP RUBY), polož šroubovák (DROP SCREWDRIVER), polož čip (DROP CHIP), polož raketu (DROP ROCKET), polož světelnou šavli (DROP LIGHTSABRE), polož matraci (DROP MATTRESS), polož hvězdu (DROP STAR), polož smaragd (DROP EMERALD), polož železnou rukavici (DROP GAUNTLET). Vezmi roh na pití (TAKE DRINKING HORN). Otoč zubem na pozici 8 (TURN COG 8). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 8. Rozhoupej kyvadlo (SWING PENDULUM), jdi na sever (N). Ocitáš se na cestě vedoucí někam na sever.

Časové pásmo 8: Římské období

Jdi na sever (N), jdi na západ (W), vstup dovnitř (IN), poklekni (KNEEL), modli se (PRAY). Tu se zjeví bůh Merkur a dává ti pár okřídlených sandálů. Vyjdi ven (OUT), jdi na východ (E), jdi na sever (N), jdi na sever (N), jdi na sever (N), vezmi sponu

(TAKE BUCKLE). Gladiátor ti však vzápětí sponu ukradne a utíká na jih k aréně. Jdi na jih (S), jdi na jih (S), jdi na jih (S), jdi na severovýchod (NE), jdi na východ (E), vstup dovnitř (IN), vezmi síť (TAKE NET), vezmi trojzubec (TAKE TRIDENT), vyjdi ven (OUT), jdi na západ (W), jdi na jihozápad (SW), jdi na jih (S), jdi na jih (S), vstup dovnitř (IN), hod' síť (THROW NET). Do sítě se tak zaplete pro vás nebezpečný lev. Hod' trojzubcem (THROW TRIDENT). Trojzubec připíchne síť k zemi a lev je tak pro vás bezpečně uvězněn na jednom místě. Obuj si sandály (WEAR SANDALS), vezmi sponu (TAKE BUCKLE), otevři dveře (OPEN DOOR), vyjdi ven (OUT), jdi na východ (E), jdi na sever (N), jdi na severovýchod (NE), jdi na severovýchod (NE), napl roh na pití (FILL HORN), jdi na jih (S), zatáhni za mřížku (PULL GRATE). Vysadil jsi mřížku bránící ti ke vstupu do římských lázní. Jdi na jih (S), jdi na jih (S). Nyní zjišťuješ, že je ti ukrutné vedro a jsi přehřátý. Napij se vody (DRINK WATER), jdi na západ (W), jdi na západ (W), jdi na jih (S). Jsi zase zpět uvnitř stroje času. Polož sandály (DROP SANDALS), polož roh (DROP HORN), polož sponu (DROP BUCKLE), polož krabičku zápalek (DROP MATCHBOX), polož kolo (DROP WHEEL).

!!NYNÍ BYS MĚL POSBÍRAT OSM PŘÍŠAD, KTERÉ JSI DOPOSUD NASHROMÁŽDIL, A KTERÉ JSOU VELEDŮLEŽITÉ DO POSLEDNÍ ČÁSTI HRY!!

Vezmi slzu (TAKE TEARDROP), vezmi kel (TAKE TUSK), vezmi vejce (TAKE EGG), vezmi olivovou větvičku (TAKE OLIVE BRANCH), vezmi křídlo (TAKE WING), vezmi čepici (TAKE CAP), vezmi čip (TAKE CHIP), vezmi sponu (TAKE BUCKLE), vezmi milník (TAKE MILESTONE). Otoč zubem na pozici 9 (TURN COG 9). Tímto úkonem jsi nastavil stroj času k přechodu do časové zóny 9. Rozhoupej kyvadlo (SWING PENDULUM), jdi na sever (N). Ocítáš

se na poli plném květin, které by mělo být rájem na zemi, ale není tomu tak.

Časové pásmo 9: Daleká budoucnost

Jdi na jih (S). Tu si přečteš oznámení, kde se píše toto: „Ať zanechá všechnu svou naději každý, kdo zde vstoupí.“ Polož milník (DROP MILESTONE). Zjeví se ti Otec čas a říká ti: „Ty se ničeho nemusíš obávat. Nebude žádná budoucnost pro tento svět, jestliže páni času vyhrají. Ještě je stále dost času, aby byli poraženi. Dveře jsou otevřeny.“ Jdi na jih (S), jdi na jih (S), jdi na jih (S), jdi dolů (DOWN), jdi na západ (W), jdi na jih (S), kopej (DIG). Našel jsi ztracenou láhev. Vezmi láhev (TAKE BOTTLE). V láhvi je jed! Jdi na sever (N), jdi na východ (E), jdi na jih (S), odemkni klece (UNLOCK CAGES). Zvířata z klecí uhánějí pryč. Jdi na jih (S), odemkni mříže (UNLOCK BARS). Lidé, co byli uvěznění jsou nyní volní a pospíchají pryč. Za odměnu za své osvobození ti dávají krabici s ďábelským okem, které je konečnou přísadou potřebnou ke kompletnímu vyřešení tvé výpravy. Jdi na východ (E), jdi na východ (E), vstup dovnitř (IN), jdi dolů (DOWN), jdi dolů (DOWN). Tady pozor na nebezpečnou kytku, která tě chce spolknout. Upusť láhev (DROP BOTTLE). Láhev se rozbije a uvolní tak jedovatou kapalinu, která usmrtí onu kytku. Jdi na východ (E), jdi na východ (E), jdi nahoru (UP), jdi nahoru (UP), jdi nahoru (UP), vezmi plášť (TAKE CLOAK), obleč si plášť (WEAR CLOAK). Tento plášť tě učiní neviditelným. Otevři padací dveře (OPEN TRAPDOOR), jdi nahoru (UP). Nyní se nacházíš pod stolem devíti zlých pánů času. Jdi nahoru (UP). Teď jsi již vzadu za pány času, kteří tě ale nemohou vidět.

!!NYNÍ SE BUDEŠ POHYBOVAT V MLHÁCH ČASU. MÁŠ VŽDY MOŽNOST JEN DVOU POHYBŮ V MLHÁCH, ABYS MOHL HODIT ODDĚLENÉ PŘÁVĚ DVĚ PŘÍŠADY DO LITINOVÉHO KOTLE!!

Jdi na východ (E), vhod' do kotle slzu (THROW TEARDROP), vhod' do kotle kel (THROW TUSK), jdi na západ (W), jdi na východ (E), vhod' do kotle vejce (THROW EGG), vhod' do kotle olivovou větvičku (THROW OLIVE BRANCH), jdi na západ (W), jdi na východ (E), vhod' do kotle křídlo (THROW WING), vhod' do kotle čepici (THROW CAP), jdi na západ (W), jdi na východ (E), vhod' do kotle čip (THROW CHIP), vhod' do kotle sponu (THROW BUCKLE), jdi na západ (W), jdi na východ (E), vhod' do kotle krabici (THROW BOX). Hotovo! Nyní je to konečně kompletní a na závěr již zbývá jen finální zpráva: „Recept je kompletní! Vaše okolí pozvolna mizí a struktura času je opravena. Máš vyhráno! Dosáhl jsi skóre 1000 bodů z 1000 možných.

Hru jsem hrál zhruba rok a stejně tak vznikl i tento obsáhlý návod. Hra vyniká svou ukrutnou rozsáhlostí mnoha místností a lokací, přičemž celá hra by jistě vydala na celou knížku. Tento návod není určitě nejkratší, ale není ani nejdelší. Troufám si tvrdit, že hru by šlo dohrát třeba i stokrát jinak. Víím, o čem mluvím, protože když jsem mnohokrát bloudil, musel jsem hledat jinou cestu, a proto je možné, že jsem některé místnosti hry ani nenavštívil. Chtělo by to určitě mapu, ale kdo by ji udělal? Je to skoro nemožné. Problematickou částí hry je samotný stroj času, který neslouží jen k přemísťování v čase, ale jako i skladiště vašich předmětů. Proto sbírejte všechno, co vidíte, neboť předmět, který nemá využití v době svého nálezů, bude určitě klíčově důležitý v době jiné. Proto je velmi důležité si hru vždycky, když jste ve stroji času, uložit. Hra se mi samozřejmě líbila, přičemž zajisté prověřil trpělivost a důvtip každého hráče. No a odradí ty ostatní, které přestane brzy bavit.

Přeji všem příjemnou zábavu. Snad zase někdy příště na stránkách ZXM.

Rally driver

Ivan Preclík

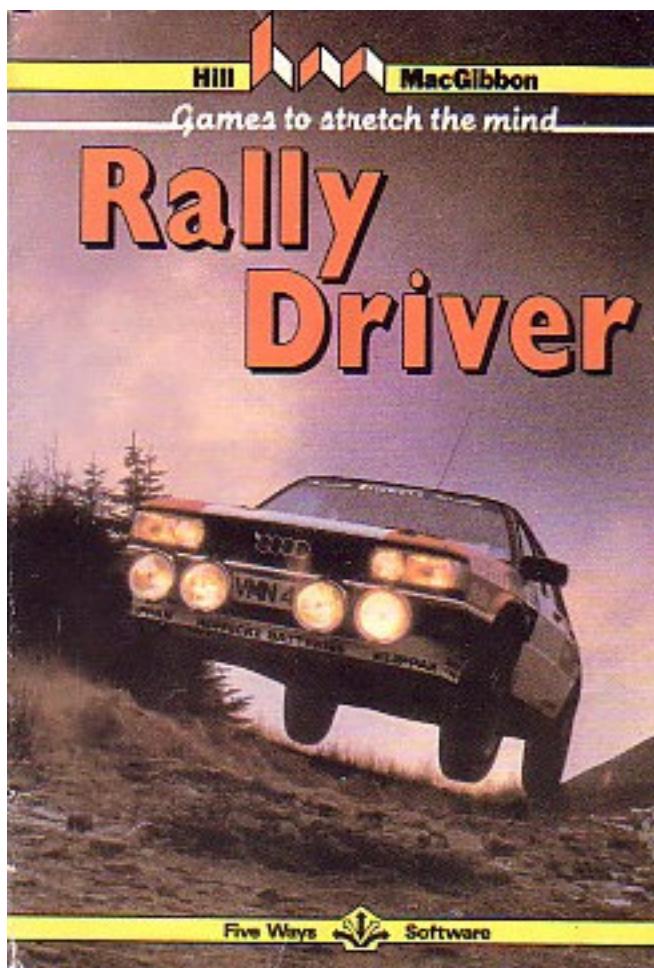
Po několika letech jsem probral své spectrácké kazety a našel jsem náhodou i jednu originálku pod názvem „4 MOST SPORT“. Jsou tedy na ní 4 sportovně laděné hry a na jednu z nich jsem pro vás přichystal recenzi.

Pokud se rozhodnete zahrát si tuhle hru, budete muset soutěžit v dost náročné automobilové rally, která vyžaduje mnohem více než jen rychlé řízení. Hra je takovým křížencem mezi akcí a strategií. Jsou zde zastoupeny obě tyto oblasti. Pro úspěšné zvládnutí jednotlivých tratí si budete muset každou trať pečlivě naplánovat a vzít do úvahy další okolnosti jako počasí, délku trati a také rychlé nebo naopak pomalé úseky trati.

Ve hře samotné platí, abyste jeli rychle a hlavně pozorně, jelikož se občas stane, že cestu vám zkříží nějaké farmářské zvíře a srážka by byla pro vás osudná.

Rally se koná ve třech závodech. Když si hru poprvé zkoušíte, vždycky začínáte v závodě I a musíte se pracovat přes II až do III. Každý závod se ještě skládá z několika dílčích rychlostních zkoušek, které probíhají ve dne i v noci. Abyste vyhráli pohár, musíte se umístit do třetího místa tedy mezi prvními třemi jezdci po posledním a závěrečném závodě.

V závodě samozřejmě nejste sami.



Konkurence je opravdu tuhá. Pouze 7 nejlepších závodníků může postoupit ze závodu I do závodu II a potom jen 5 nejlepších může postoupit ze závodu II do závodu III.

Ke hře samotné přikládám originální ofocenou mapu jednotlivých tratí. Doufám, že se tu někde objeví, protože bez ní se ve hře nikde nedostanete. Každý závod se skládá z určitého množství kontrolních stanovišť, která jsou vyznačena na právě mnou zmíněné mapě. Musíte projet všemi těmito kontrolními stanovišti, abyste dokončili celý závod. Z vlastní zkušenosti mohu říct, že existuje vždycky několik cest, jak se dostat a projet každým kontrolním stanovištěm. Tady vyloženě záleží na vašich navigačních schopnostech

spočívajících v tom, jak jste si vybrali nejlepší kurs vaší jízdy.

Co vám doporučuji je, abyste před startem každého závodu si pečlivě prostudovali mapu a vybrali vaši nejkratší trasu, kterou pojedete k cíli. Není vám však dovoleno jezdit všemi cestami, které jsou zobrazeny na mapě! Vše potřebné je vyznačeno v legendě mapy, která je umístěna dole v rámečku pod mapou. Jak můžete vidět, jsou tu cesty, které jsou vždy otevřené pro rally, pak jsou cesty, které jsou jen občas otevřené a nakonec jsou tu cesty, co jsou naprosto nepřístupné. Občas i podmínky počasí mohou způsobit, že některá z cest bude uzavřena. Například po vydatném dešti mohou být některé cesty zaplaveny. Abyste zjistili, které

z cest jsou zavřené nebo otevřené, musíte zastavit na některém z kontrolních stanovišť, kde vám komisař poskytne potřebné instrukce týkající se změněných podmínek závodu.

A aby to nebylo tak jednoduché, na trati se vyskytují kromě kontrolních stanovišť ještě tzv. průjezdové kontrolní body. Jsou to v podstatě jednoduché informační tabule, které jsou rozesety různě podél tratě. Na každé z těchto tabulí jsou napsány 2 písmena. Nemusíte u průjezdových kontrolních bodů zastavovat, abyste zjistili, co je na nich napsáno. Měli byste však jet v tomto místě maximální rychlostí 45 m/h. Pokud pojedete rychleji, váš palubní navigátor nebude schopen přečíst tato písmena. Měli byste si

zajistě poznamenat tato písmena, abyste se vyhnuli případné časové penalizaci na následujícím kontrolním stanovišti, kde se vás na ně zeptá komisař.

Kontrolní stanoviště na trati jsou místa, kde komisaři kontrolují každého ze soutěžících, aby mohl dál pokračovat v závodě. Abyste mohli zastavit na kontrolním stanovišti, musíte najet svým autem mezi dvě čáry vyznačené na cestě nacházející se u boxu časové kontroly. Pokud zastavíte příliš brzy, musíte dopředu popojet. Pokud naopak špatně zastavíte a přejedete, budete časově penalizováni 50 sekundami. Z kontrolního stanoviště se nemůžete otočit a jet zpátečním směrem, odkud jste přijeli, nýbrž musíte pokračovat rovně dál v závodě. Když zastavíte na kontrolním stanovišti, komisař se vás nejdřív

zeptá na písmena, kterých jste si měli všimnout na některém z průjezdových kontrolních bodů. Naťukajte písmena na klávesnici a stiskněte ENTER.

Pokud zadáte špatné písmena, budete nemilosrdně časově penalizováni 20 sekundami. Pak vám komisař poskytne důležité informace o počasí nebo o jiných změnách podmínek závodu. Jakmile domluví, můžete jet, stiskněte klávesu 9 pro nastartování vašeho vozu a pokračujte dál v závodě.

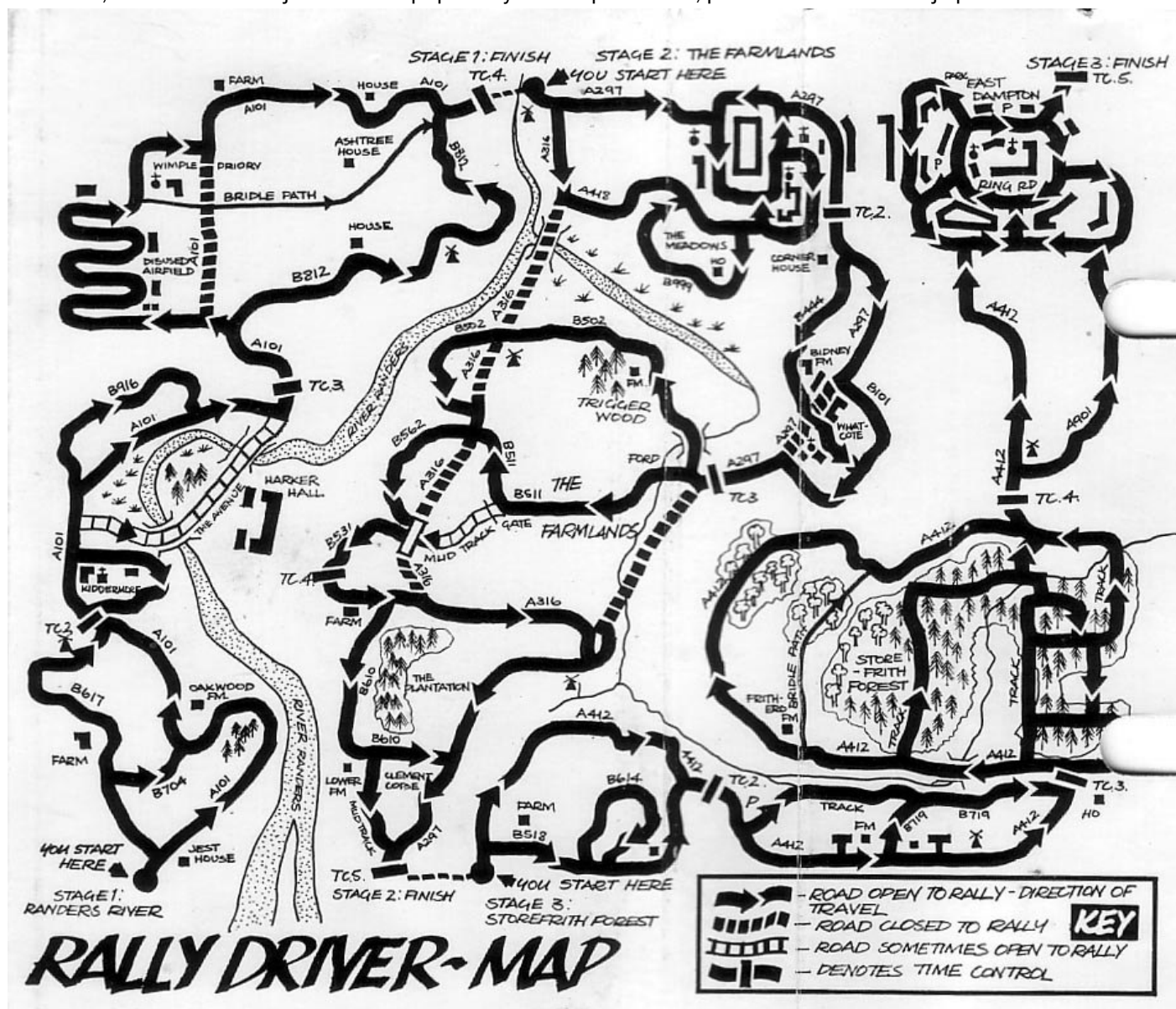
Jakmile dokončíte některý ze závodů, obdržíte vaše dosažené skóre. To se skládá z vašeho času dojetí, který se vám zobrazí v minutách a sekundách, jenž je měřený od vašeho startu po projetí posledním kontrolním stanovištěm a ještě k němu mohou být připočteny časové penalizace, pokud

jste si je během závodu přivodili. Čím nižší je vaše skóre, tím lepší byla vaše jízda!

Časové penalizace můžete obdržet z těchto důvodů:

- Σ Rozbití vašeho vozu – dostanete 30 sekund navíc nebo dokonce až 5 minut za příliš rychlou jízdu.
- Σ Vyjetí mimo trať – dostanete 30 sekund navíc.
- Σ Pokus jet uzavřenou nebo zakázanou cestou – dostanete 30 sekund navíc.
- Σ Srážka se zvířaty nebo vašimi soupeři – dostanete 5 minut navíc.

Hru samotnou hodnotím jako velice zajímavou, rozhodně ze všech 4 her, co byly na kazetě, je Rally driver rozhodně nejlepší. Na 48K hru má



slušnou grafiku, ale má pramálo zvuků a o nějakém hudebním doprovodu nemůže být ani řeč, takže v průběhu hry bude vaše Spektrum skoro němé. Hra je jistě náročná na soustředění a zřejmě její autor nechtěl hráče nikterak rozptylovat. Pohled závodníka na trať, kam jede, je řešený z kabiny vozu. Ve spodní třetině obrazovky je umístěna palubní deska s volantem. Ve zbylých dvou třetinách obrazovky je vykreslována cesta s krajinou, kde jedete. U palubní desky se ještě zastavím. Na ní můžete vidět váš čas, aktuální rychlost, řazení, ale nejdůležitější je uprostřed palubní desky malý display. V něm v průběhu vaší jízdy se občas objevuje text. To je váš tzv. palubní navigátor, který poskytuje důležité informace především o případných časových penalizacích nebo o aktuálním počasí apod.

Hra má dost obtížné ovládání. Řízení je dost citlivé a tak je někdy dost



těžké vůbec udržet vůz na silnici. Velice opatrně se musí najíždět do zatáček, pokud možno snižte patřičně rychlost, jinak vyjedete z trati a máte po nadějích. To samé platí, pokud na cestě objíždíte zvíře nebo nějakou jinou překážku. Při zatáčení nebo vyhýbání se překážky doporučuji jet maximálně rychlostí 25 m/h.

Hru ovládáte těmito klávesami:

SPACE – přeskakování voleb v menu hry.

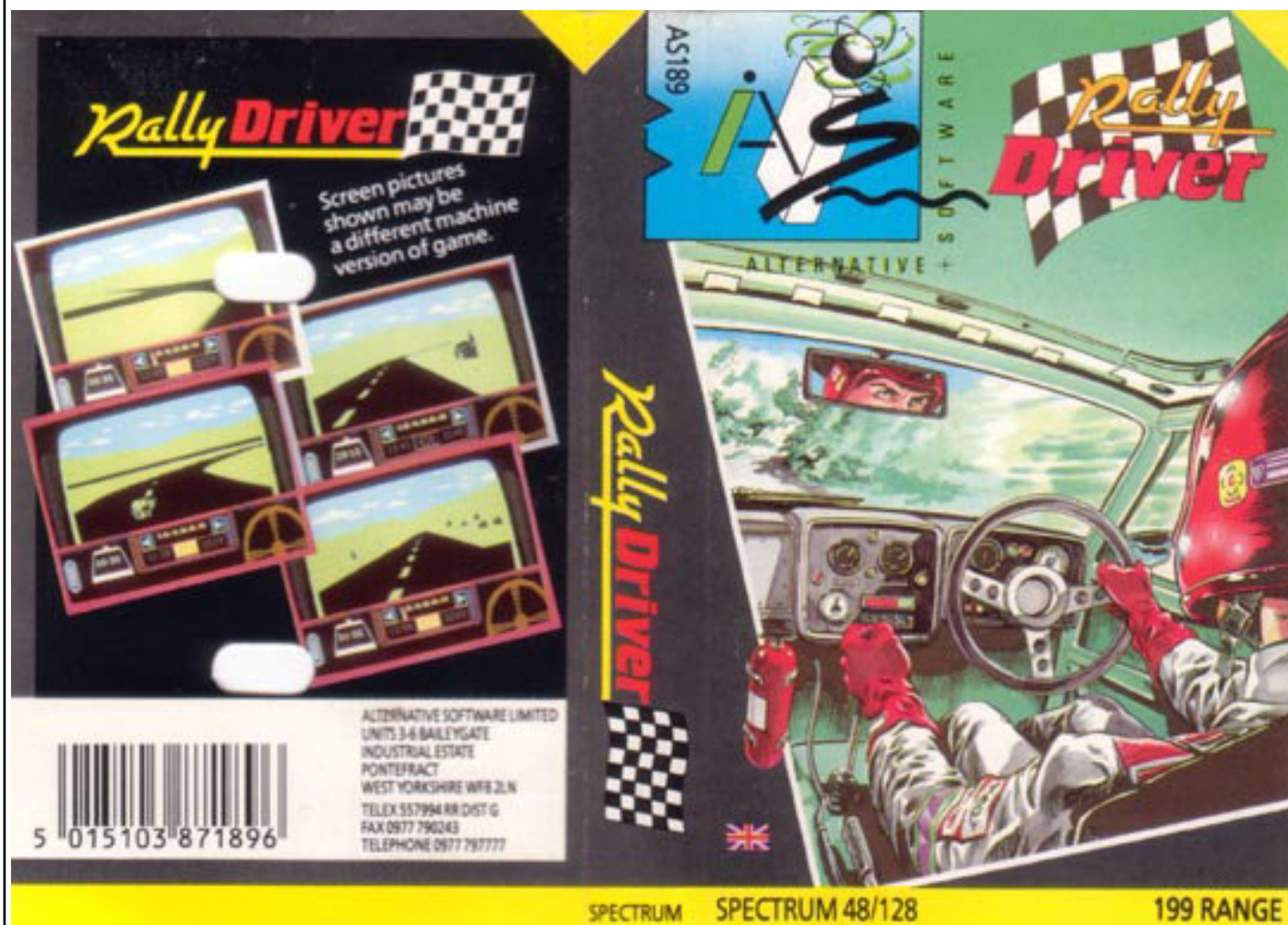
ENTER – potvrzení voleb v menu hry.

- 1 – posun řazení doleva.
- 0 – posun řazení doprava.
- Q – zatáčení doleva.
- O – zatáčení doprava.
- 9 – přidávání plynu.
- 2 – brzdění.
- 3 – ruční brzda on/off.
- S až K – troubení.

Můžete použít i připojený joystick v kombinaci s klávesami.

Nezkoušejte mne nasrat a hru si rozhodně zapařte. Jsem přesvědčen o tom, že se některým z vás bude líbit. Hra prověří mnoho vašich schopností, jako je trpělivost, soustředěnost, rychlé reakce a hlavně dobrý úsudek. Přeji všem mnoho strávených hodin s Rally driver a nenechte se odradit počátečními nezdary.

Zase někdy příště na stránkách ZXM.



Space Raiders

Logout

Hra, kterou jsem od roku 1995 měl jako první na své oblíbené kazetě HRY III. Je to hra od méj oblíbené firmy, distribuovaná samotným tvůrcem ZX Spectra. Jedná se o hru Space Raiders z roku 1982, jejímž autorem není nikdo jiný než světoznámá firma PSION a distributorem nikdo menší, než Sinclair Research.

Pro ty z vás, kteří posledních dvacet tři let trávili zakopáni v bedně o rozměrech 2x2x1 metr, patnáct metrů pod zemí uprostřed pouště na severu Nevady a tudíž snad neví o co v ní jde, následuje stručný popis:

Pomocí pohyblivé gunbase máte likvidovat postupně se přibližující vlny mimozemšťanů a tak ochránit ženy a dívky planety Země před znásilňováním slizkými eMZáky, cenné přírodní zdroje před plněním a mimozemšťany před zpěvem Sámery Issy. Taktika mimozemšťanů je oprav-

du proradná a nejlépe ji lze formulovat jako "Klesnout o řadu a změnit směr !"

V každém kole je proti vám připraveno pět řad útočníků složených z dvou řad červených (po deseti bodech za kus), dvou řad zelených (po dvaceti bodech za jednoho) a jedna řada žlutých (Zabij a dostaneš 30 bodů). Mimozemšťané se pohybují, v pravidelných intervalech přibližují a při tom všem po vás ještě střílí. Pokud se dostanou až k vám, pak je samozřejmě zle a ani Atari 130 XE vám

nepomůže, abyste je vypudili.

Chrabrý ochránce má samozřejmě k dispozici neomezenou municí a navíc ještě tři kryty. Tyto kryty jsou veskrze zajímavá věc. Podléhají střelbě

mimozemšťanů, proto se časem slušně opotřebují. Obnovit je lze tak, že necháte posledního vetřelce sestoupit až na úroveň krytů a tím se staré kryty smažou a po jeho sestřelení dostanete zbrusu nové, ještě nerozbalené. Postupem času určitě přijdete na věci jako prostřilet se skrz a z pohodlí domova vraždit slizouny a další a další.

Hra se ovládá pomocí kláves 'Z' (vlevo), 'X' (vpravo) a 'SPACE' (střelba), napoprvé je docela obtížná i přes občasné bonusy, které přináší nahoře prolétající fialové lodě. Napoprvé jsem se dostal sotva k 1300 bodů za hru, po cca dvaceti minutách asi k pětinasobku tohoto skóre. Mimozemšťani jsou opravdu vytrvalí, postupem hry i rychleji a střelby také neubývá. Hra je barevná, vyluzuje na vnitřním beeperu poměrně zajímavé zvuky a opravdu se u ní dá vydržet hodiny a hodiny. Mohu jen doporučit.

Nějaké ty screenshoty:

Pěkný úvodní obrázek - hnusný eMZák (pod titulkem článku).

Všichni na svá stanoviště, připravit na útok!

Kryty po prvním kontaktu nevypadají moc nově.

Pane prezidente, musím vám oznámit, že obrana Země selhala...



Meteoroids

[Logout](#)

Lidem tvalo mnoho desítek let, než se dostali do vesmíru. Škoda jen, že je plný šutrů a nepřátelských mimozemšťanů



Dnes vám přinášíme další recenzi na další pecku roku 1982. Jedná se o hru Meteoroids od Dana Priestleye, distribuovanou firmou DK'Tronics. O Danu Priestlym bude ještě řeč, hned další recenze budou jeho Tanx. Je to varianta hry Asteroids známé snad ze všech platforem. Pokud by se snad někdo našel, kdo neví o co v Asteroid-ech jde, můžeme mu jednak doporučit jedno skvělé místo v Nevadě a druhak mu to samozřejmě objasníme:

Vesmír není dobré místo k žití. Lodě jsou v něm permanentně ohrožovány meteoroidy, což jsou de facto kusy vesmírných šutrů nejrůznější velikosti, od kamínků po balvany velikosti několikanásobku lodí. Naštěstí naši nejlepší vědci vymysleli zbraň, která nepotřebuje dobíjet a proto je můžeme úspěšně likvidovat. Krom pol-etujících fragmentů skály ještě člověka ve vesmíru ohrožuje Atari, na kterém někdo zapoměl vypnout přehrávání hudby; hlad, když v dosahu není žádné bistro a hlavně mimozemšťani, kteří mají ve zvyku likvidovat vše co se jim dostane do spárů. Ještě že ani oni nejsou imunní vůči pořádné bouchačce.

Po nahrání hry se ocitneme v úsporném, leč přehledném menu, kde máme možnost nastavit parametry hry (klávesa R), či spustit hru pro jednoho (klávesa 1) nebo více (klávesa 2)

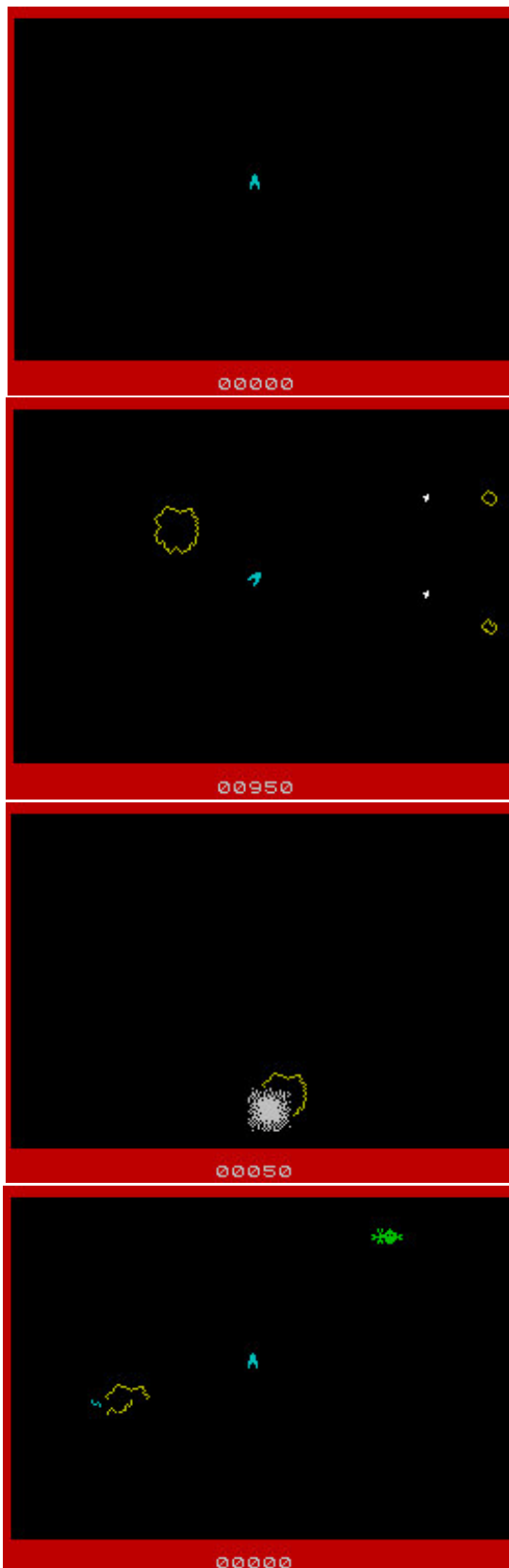
hráčů. Volbu R logicky volíme pokud chceme ve hře něco přenastavit. To 'něco' může být ovládání (aktuální ovládání je zobrazeno v hlavním

menu), rychlost hry, počet životů a další věci, na které každý, tváří v tvář faktu že jsou vypsány na obrazovce, velice snadno přijde. Pokud se rozhodneme vybrat hru více hráčů, musíme zadat jejich počet (maximálně 6). Hráči nehrají najednou, střídají se vždy když jeden přijde o život. Upřímně řečeno je to vzhledem k brajglu co je na obrazovce když je tam pár meteoroidů, mimozemšťan a hráčova raketka jedině plus.

Ve hře je, jak bylo nesměle naznačeno hlavním úkolem přežít. K přežití musíte ničit vše co se okolo vás hýbe. Můžete otáčet lodí a pohybovat jí po obrazovce (pokud se dostanete až k borderu, loď se ocitne někde jinde na obrazovce). Při nastavení větší rychlosti a mála životů je hra pekelně obtížná a můžu garantovat, že u ní propotíte nejen pletené papuče od babičky ale i trenky od Rákosky.

Hra je určena pro všechna Spectra, včetně 16K. Její grafika není tak propracovaná jako na

mnohobitech, o to lepší mi ale přijde její hratelnost a můžu říct, že jsem u ní strávil hezkých pár hodin i po té, co jsem dopsal většinu této recenze.



Fat Ware



Aragorn

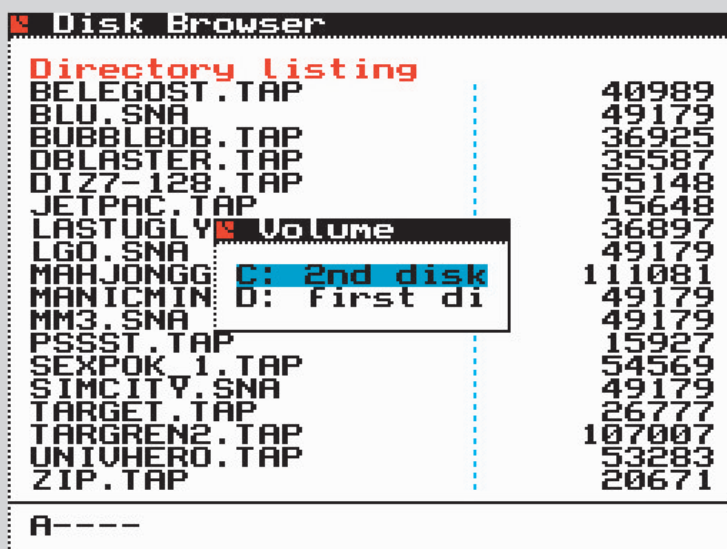
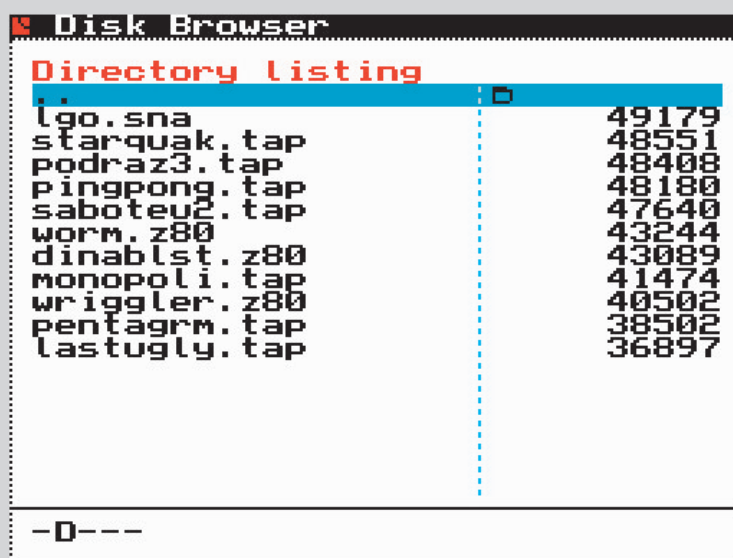


FatWare je nový systém od Bazeho pro Divide, který se nahraje (naflashuje) do EEPROM a umožňuje přímo pracovat s IDE disky, na kterých je zformátován logický disk s FAT16. Z toho plyne omezení, že jeden disk může mít 2GB, ale zase jich může být na jednom zařízení více.

Stávající verze umí zatím pouze čtení, a to se soubory TAP, SNA a SCR.

Do systému vstoupíte přes tlačítko NMI, kde v menu vyberete soubor. K TAPkám pak můžete přistupovat přes BASICovský příkaz LOAD " ", soubory SNA se přímo spustí, a SCR jsou buď klasické obrázky, nebo Dithvide, které pak systém zobrazí.

<http://baze.au.com/divide/>



Kaplicon

(29.7.-31.7. 2005)

Kde?

Kaplice. Kaplici najdete v jižních čechách směrem od Českých Budějovic na Dolní Dvořiště. Kousek mapy snad pomůže se zorientovat.

V Kaplici se ptejte po Základní škole, ve které se bude letos konat. Pokud se dobře pamatuji, tak je to tak 300m od náměstí směrem k hlavnímu tahu (E55). Dřívější ročníky (2003 a dříve) byly v Domě dětí a mládeže, který je zrušený.

Doporučuji vám přijet autobusem nebo autem (autobusem se dostanete až do města), vlak staví celých 5 kilometrů od města, autobus odtamtud zpravidla žádný, ale stopem se do města dostanete (ovšem musíte stopovat sami). Spojení si můžete vyhledat i na netu (www.vlak-bus.cz). Mapka znázorní lépe:

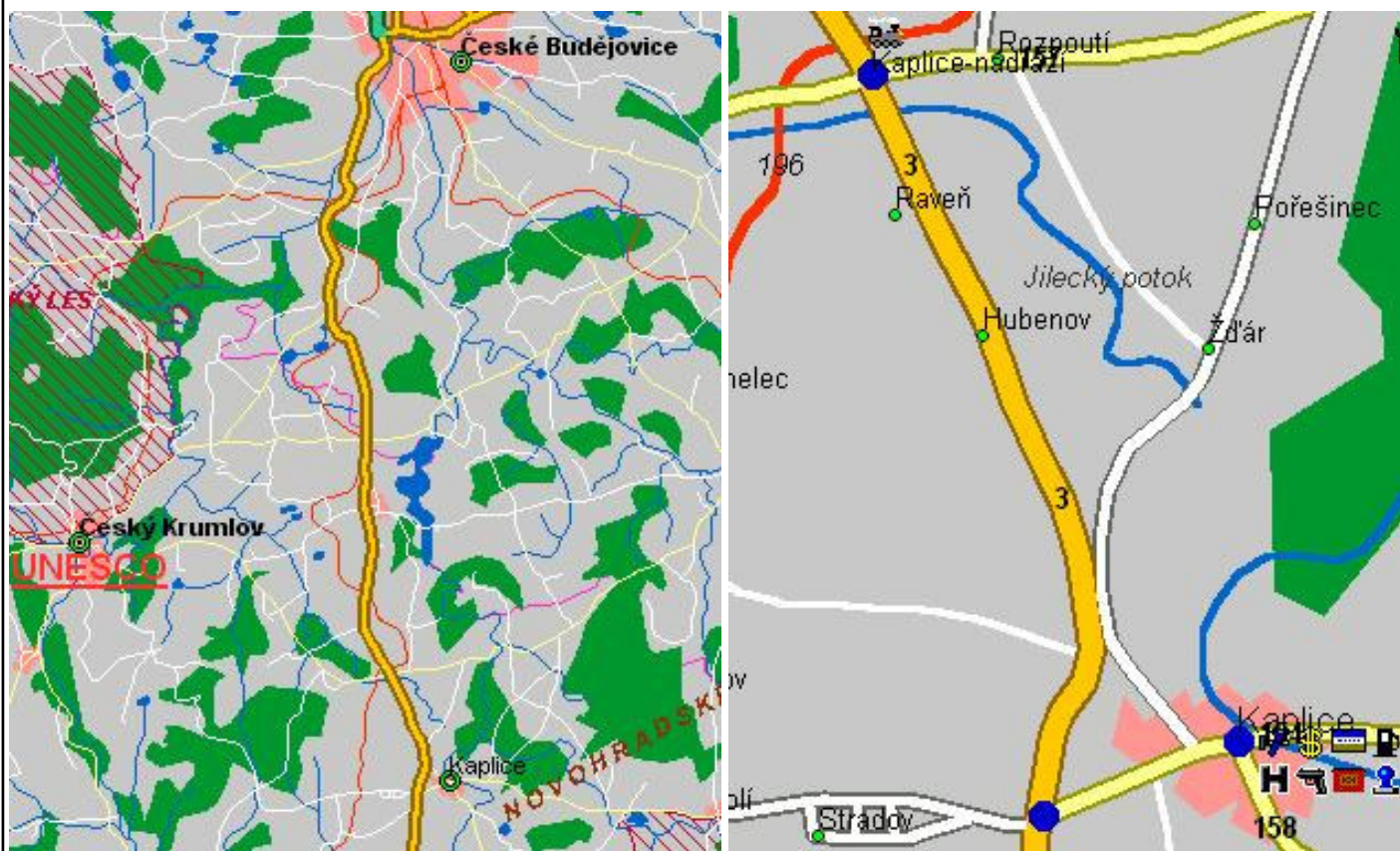
Kdy?

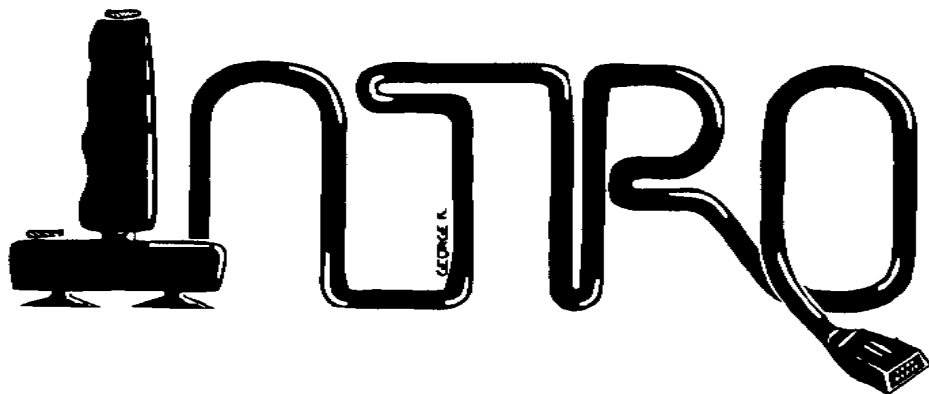
Kaplicon 2004 se koná 29.7.-31.7. 2005 (tj. pátek-neděle o posledním weekendu v červenci).

K dispozici je asi 6 míst na spaní (luxusně v posteli), takže těm co se ohlásí jako první, stačí jen spacák, případně deka. Ohlasit se můžete do formuláře na zxm.speccy.cz nebo přímo mě do pošty, nebo SMS či brknout na 607 819 905 (Aragorn).

Takže počítáme s Vámi.

IJK, Sweet Factory a Aragorn





Vaše dopisy v /dev/null končí

Nazdar.

Používám RealSpectrum najmä na hranie hier v D80. V robení D80 nie som žiaden začiatovník, ale predsa som narazil na jeden problém: neviem ako skopírovať hry z originálneho Didaktiku Kompakt na D80 formát, ide konkrétne o originálne hry od Ultrasoftu a sú na originálnych distribučných diskoch. Pri normálnom Read Into Image File mi píše 100 percent chybných sektorov a keď som skúšal inak kopírovať (zapísať prázdny image na d80 a potom skopírovať súbory), hra mi zhorela na kontrole originálu (vypíše že hra nie je na originálnej diskete a že si musím "kúpiť" originál)

Neviete mi pomôcť?

Dík.

Matus alias FRGT/10

Celý problém spočíva v tom, že Ultrasoft používal formát o 10 sektorech na stopu. Emulátor Real Spectrum má vždy defaultne nastaveno 80x09 ("D80 STD"). Proto těch 100% chybných sektorů při pokusu o přenos. Abys tedy data bezproblémů přenesl, musíš nastavit 80x10 ("D80 800K"). Pokud však byla disketa formátována postaru MDOSem 1.0 (tj. chybnou formátovací rutinou, co má v ROM), může se stát, že budeš mít ještě problém s index pulzem nebo špatným číslováním druhé strany diskety. Vše by se o5 projevilo vadnými sektory. V takovém případě je tedy nejjednodušší přímo na Speccym soubory z originální diskety zarchivovat jako D_0 (pomocí MDOS File Commanderu) a na PC si z nich pak nechat udělat image D80 (pomocí MFC d_0 Converteru v3.1 nebo vyššího). Ten PC konvertor totiž dokáže detekovat D_0 soubory, které byly vytvořeny z originální deskety od Ultrasoftu a image vytvoří už i s tou přibližlou ochranou, takže žádná hláška typu "kup si originál".

Měl bych dotaz ohledně kopírování formátu D_0 na disketu. Mám disketu 720K PC formát, a když kopíruju ve Windows v Total Commanderu Xmagazín na disketu, abych mohl provést následně import v MFC, napíše mi po 56 souborech, že je disketa plná, i když je tam skoro polovina volného místa. Kde dělám chybu? Je to otázka clusteru? Nebo to musím kopírovat nadvakrát?

Dík za radu.

Chip

Na PC použij MFC d_0 Converter v3.1 (nebo vyšší). Ten umí konvertovat d_0 přímo do d80 image, takže žádný přenos přes deskety není nutný. Ale zpět k otázce, proč udělá Total Commander 56 fajlů a končí. Za tohle můžou naše "skvělé" Windows! Resp. Billova podpora dlouhých jmen. Každý soubor má na disketě vyhrazeny minimálně 2 (a to i když je jméno klasického formátu 8+3 !?) položky v adresáři a 112/2 je tedy 56. Pokud bys ale kopíroval (klidně pod Win) dosovskou M602, problém by se vůbec neobjevil. Zajímavé, že? :)

Nemáte někdo nějakou tabulku úplně všech instrukcí Z80, i nedokumentovaných, s časováním?

Omikron

Máme! Samozřejmě ti doporučím tu svou (<http://mts.web.wo.cz>), ale Busy by mi jistě neodpustil, kdybych zde ne-napsal i odkaz na jeho kodex (<http://busy.host.sk>).

Dementi

Představte si, že tentokrát jsme vůbec mezi sebou dementy hledat nemuseli. Oni se ozvali a označili sami.

Vše to odstartoval Sweet tvrzením

"Takže od teď, kdo nema divide a hard-disk, tak je megakokot a megadement" no a pak už stačilo jen počkat...

DiZZy: "Ja jsem ovsem ultragigakokot, protože divide sice mam, ale stale jeste nezprovozneny."

TDM: "...ja jsem zcela jiste vetsi ultragigakokot, nez bych sam cekal - stavebnici mam jiz nekolikaty rok nesestavenou."

Písmenková je nejlepší

Písmenkovou polévku doporučuji 4 z 5 spektristů.



until
the end
of

ATARI

