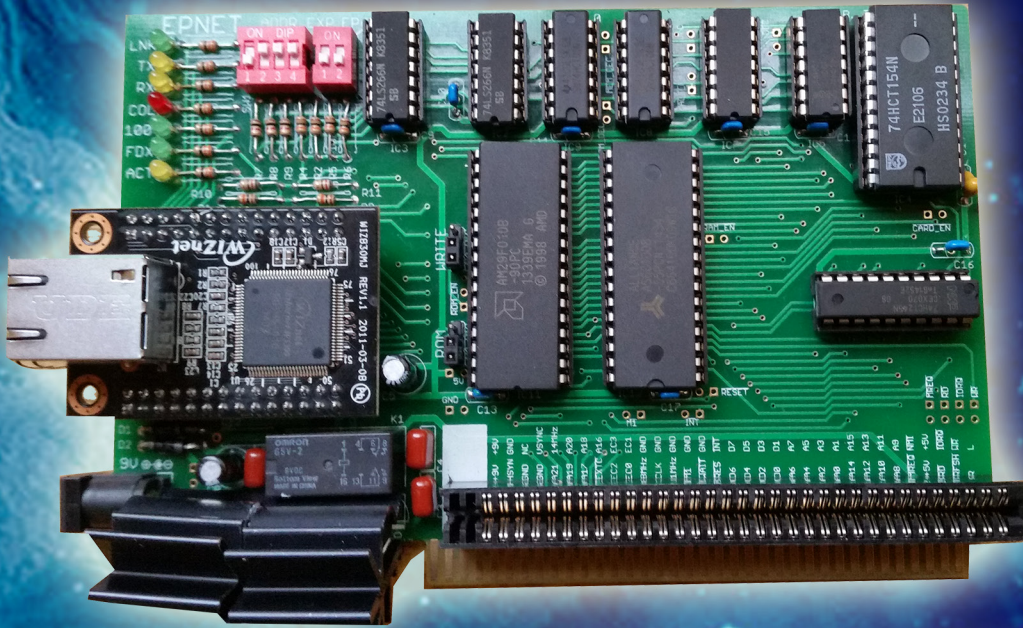


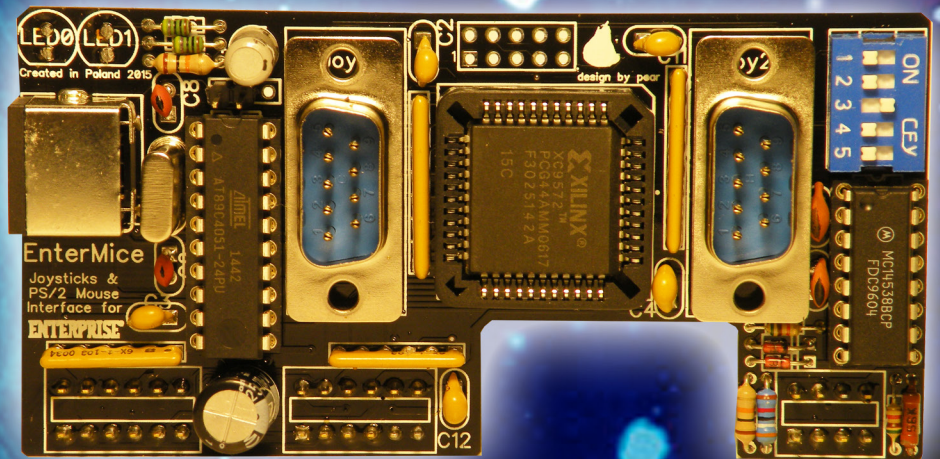
ENTERPRISE

Magazin az ENTERPRISE felhasználóknak

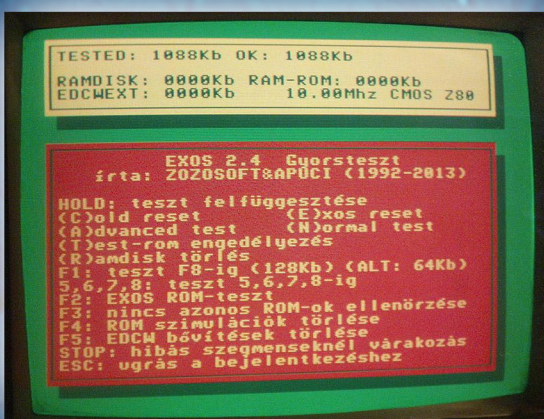
2016. november



EPNET kártya (by Bruce Tanner)



EnterMice (by Maciej Gruszecki)



10 Mhz Enterprise-on (by Zozosoft)

Határtalan lendülettel



Írta: **Matusa István (Tutus)**

Igen, jól látjátok, ismét megjelent az Enterpress Magazin!

Az Enterpress történetének folytatása érdekes. 2015. májusában hívtak fel **Zozo**, hogy lesz egy jubileumi klubtalálkozó. Ráadásul nem is akármilyen: **30 éves az Enterprise!** Erre az alkalomra külföldi vendégeket is hívtak, **Bruce Tannert** az egyik Enterprise szoftver fejlesztőt, valamint **Jörn Mika**, a SymbOS operációs rendszer készítője lett még meghívva németországból.

Megmondom őszintén, hogy a jubileumi klubnap előtt utoljára 2008-ban voltam jelen az Enterprise Forever fórumban, ami több év kihagyást jelentett. Zozo jelentkezése és a szülinapi rendezvény kapcsán igencsak beindultam. Kitaláltam, hogy nagyon gyorsan készíteni kell (kb. egy hetünk volt Zozoval...) egy **jubielumi Enterpress újságot** a találkozóra. Ez sikerült is, színes címlappal, kicsit kapkodva, de ott volt a nyomtatott magazin a klubnapon. A lendület ezután sem csökkent, elhatároztam, hogy a régebbi Enterpress Magazinokat digitalizálom elektronikus formátumban. Ez kicsit „nagy falat” lett, eddig az első két számot sikerült így megjelentetni. Folyamatosan dolgozom a többi szám digitalizálásán. Egy honlap is kellett az újságnak, ez is létrejött: <http://enterpress.news.hu>.

Természetesen úgy gondoltam, hogy új Enterpress Magazinok is megjelennek majd. Igaz, időszakosan, de korlátozott példányszámban nyomtatott formában és persze elektronikus formátumban is, melynek előnye, hogy

linkeket videókat is lehet a cikkekbe ágyazni és nincs akadálya, hogy több oldal legyen mint a nyomtatott formátum, valamint a tartalomban lehet keresni. Ha már több régi magazin lesz fent, ezeket archívumba lehet egyszerűsíteni és a teljes archívumban is lehet keresni (keresztkeresés).

No de beszéljünk arról is, hogy az általam nélkülözött évek alatt milyen fejlesztések történtek. Megmondom őszintén, csak kapkodtam a fejem, mikor Zozo próbálta velem behozatni az elmulasztott éveket... Szerencsére a Facebook beszélgetésünket vissza tudom nézni :).

- **Varga István (IstvanV)** fantasztikus PC emulátora, kép- és zenekonvertáló programjai,
- az éppen akkor futó EnterMice projekt melyet a spanyol srác, **gflorez** rábeszélésének köszönhetően **Pear** álmodott meg lengyelországból,
- **Prodatron** SymbOS grafikus operációs rendszere, melyet CPC-n fejlesztett, Enterprise gépen is működik,
- **Saint** az Egyesült Királyság színeiben belső, 1 MB-os memóriabővítő tervezett,
- a legnagyobb dobás pedig egyértelműen **SzörG** SD kártyája, melynek köszönhetően elfelejthetjük a kazettás magnót, floppyt és winchestert. Ezt a kis kártyát a bal oldali cartridge helyére dughatjuk
- **Persa Noel (geco)** CPC, Spectrum és TVC emulátorjai Enterprise-ra
- **Zozo** folyamatos munkái az EXDOS és EXOS rendszeren, valamint ő látja el egyedül a hazai és külföldi szervizhálózatot. Aproó, ha valaki el tudja neki intézni, hogy egy nap legalább 36 órából álljon, ő külön jutalomban részesül :).

De folytassuk az éppen aktuális fejlesztésekkel:

- **Bruce Tanner** EPNET kártyája, melyvel csatlakozhatunk az internetre

és megfelelő programokkal lehetőség lesz az alábbi szolgáltatásokra: FTP, Chat vagy akár szöveges alapú böngésző,

- **Zozosoft** FAT16-os EXDOS verziója, belső RAM/Flash/Turbo kártya,
- **Lénárt Gábor (lgb)** XEP128 EP emulátora PC-re, Linuxra és Mac OSX-re
- **Pear 3** kártyája, FlexiBridge, IDE-Compact Flash, RAM-Flash-Clock.
- Közben lefutott az **Enterprise Dev-Compo #1** programozási verseny, melyre sok kiváló program készült.
- Azóta világot látott még **Povi** 2048 játéka Enterprise-ra, valamint az újonc **Sdw** szuper demója, aki a gépét nemrégén vette.

Terveink között szerepelt a budapesti **Enterprise Klub** felélesztése. Ez már egy picit nehezebb feladatnak bizonyult. Korosztályunk már igencsak 40-50 év körül mozog, nyilván a család, munkahely és más problémák közbeszólnak és ez érthető. Ennek ellenére 2016. novemberétől reményeink szerint ismét havonta tudunk találkozni, jövőre pedig ismét tudunk külföldi vendégeket hívni a klubtalálkozókra. A Klubról részletes információt taláhattok majd a <http://www.enterpriseklub.hu> oldalon, melyet 2017. január 1-től indítunk ba. Ezzel egyidőben egy klubtagsági rendszert vezetünk be, mely a tervek szerint három különböző előfizetést tartalmaz majd.

Ha ezt a tempót tudja tartani az Enterprise Forever kis közössége, akkor még sok örömben lehet része az Enterprise számítógépek tulajdonosainak!

Ha szeretné támogatni az ENTERPRESS Magazin megjelenését, itt megetheded:

<http://enterpress.news.hu>
(az oldal alján „TÁMOGATÁS”)

EPNET



Írta: Bruce Tanner
(BruceTanner)

Háttér

Harminc évvel ezelőtt az Intelligent Software nevű angliai vállalatnál dolgoztam, melyet az Enterprise Computers alkalmazott tervező cégként. Itt készítettem el az IS-BASIC-et és az IS-FORTH-t, és az EXDOS valamint az IS-DOS egyik felét. Az IS és az Enterprise Computers jól ismert összeomlása után Robert Madge, én és még három IS alkalmazott (az egyikük Martin Lea volt, aki az EXOS-t, továbbá az EXDOS és az IS-DOS másik felét megalkotta) létrehozta a Madge Networks nevű céget, ahol a következő tíz évben különböző hálózati termékeken dolgoztam. A munkám az alacsony szintű assembler szoftverektől kezdve egészen a Windows alkalmazásokig sok mindent érintett. Többé nem igazán foglalkoztunk az Enterprise-szal, hiszen amennyire tudtuk, ez már halott ügy volt. A továbbiakban teljes erőbedobással dolgoztunk feltörekvő technológiákon alapuló új és izgalmas termékeken, számítógépes hálózatokon és PC-ken. Huszonöt évvel később történt, hogy megpillantottam egy eladó Enterprise-t az eBay-en, ráadásul drágább volt, mint a maga korában! Nem tartott soká és szereznem egyet. Illetve fedeztem az "Enterprise Forever" fórumot. A legmeglepőbb az volt, hogy addigra Magyarország huszonöt évnyi Enterprise történelmet tudhatott maga mögött, amiről én semmit nem tudtam!

Azonnal ellenállhatatlan készíttetést éreztem néhány szoftver letöltésére az "új" Enterprise-omra. Persze mindjárt rá is döbbsentem, hogy magnó nélkül - ami nekem nem volt - nem volt rá mód, hogy bármilyen programot is kipróbálhassak rajta.

Az én háttérrel (először az Enterprise, aztán a hálózatok) logikusnak tűnt, hogy én vagyok a megfelelő személy egy hálózati kártya megtervezéséhez. Ráadásul véletlenül épp akkoriban alakítottam át az otthoni garázsomat elektronikai műhelyé, amit már évek óta tervezgettem. Már csak az első, műhelyhez passzoló feladatot kellett megtalálni ...

A dizájn

Minden hardver tervezése egy sor kompromisszumot és megalkuvást is jelent. Az első és legfontosabb számomra az volt, hogy ne essek a „sokat akar a szarka, de nem bírja a farka” hibájába, így a projektnek jó esélye legyen a kész és működőképes állapot elérésére. Szakmám szerint nem va-

Egy másik limitáció abból az érzésből fakad, hogy kissé problémásnak vélttem túl sok csúcstechnikát illeszteni egy Z80 alapú rendszerhez. Manapság a processzorok 1-2 GHz órajellel működnek, szemben a Z80 4 MHz-ével, közel ezerszer gyorsabban. 1984-ben 64 Kbyte egész sok tárhelyet jelentett, manapság a zsebemben lapuló telefonban is 16 Gbyte van - több mint a százezerszerese!

A harmadik korlát pedig az, hogy ez alapvetően egy hobbi projekt. Szerettem volna aránylag egyszerű forrasztási eljárást használni, és kerülni az apró, felület szerelt alkatrészeket, melyek szereléséhez különleges technika szükséges. Hőmérséklet kontrollált forró levegős kemence és egyéb speciális eszközök.

Így elhatároztam, hogy a tervem alapján véve az EXDOS kártyáéra fog em-



Az EPNET kártya közvetlenül a gépbe dugva

gyok hardverfejlesztő. Viszont egész szakmai karrieremet hardver közeli munkákkal (driver-ek írása), illetve új hardver elemek piacra dobása kapcsán felmerülő kérdésekkel és feladatokkal töltöttem el. Tinédzserként magam terveztem és marattam saját NYÁK-jaimat, persze a jelen feladat komplexitása jóval felülmúlja azokét. CAD programokat láttam már használat közben, viszont én magam ezelőtt még nem használtam ilyesmit, ezért eltartott egy ideig megtanulni.

lékeztetni. Ez utóbbi tartalmazza a címdekódoló logikát és a ROM-ot, valamint vezérel egy IC-t, amely a lemezmeghajtó elérésének bonyolult feladatát végzi. Saját kártyám ehhez hasonlóan tartalmazna egy címdekódoló logikát, ROM-ot (és RAM-ot), és egy IC-t, vagy egy modult, amely a hálózat elérésének komplex feladatát végzi.

Egy WIZNet nevű cégnek vannak olyan moduljai, melyek képesek elvégezni a hálózati kapcsolódás alacsonyabb szintű feladatait. A modul rendelkezik

a konkrét hálózati csatlakozóval és a szükséges elektronikával, valamint egy IC-vel, mely tartalmaz egy processzort, RAM-ot és ROM-ot benne a szoftverrel, ami az alacsony szintű protokollok kezeléséhez szükséges, mint amilyen az ARP, az ICMP, az IP és a TCP. Továbbá, a modul közvetlenül összekapcsolható a Z80 adat- és címbuszával, I/O vagy memóriába ágyazott módon való működést lehetővé téve.

Úgy tűnt, ez ideális! Lehetőségem volt választani, egyrészt ott volt a régebbi 8 bites w5100-at tartalmazó modul 32K RAM-mal, illetve az újabb, 16 bites w5300-at alkalmazó 128K RAM-mal. Féltem tőle, hogy a régebbi modell (mely már tíz éve jelent meg a piacon) hamarosan beszerezhetetlenné válik. Az újabb könnyebben elérhetőnek bizonyult az Egyesült Királyságban és Magyarországon is és valamivel még olcsóbb is, mint a régi. A választás tehát egyszerűnek tűnt. Egy hátránya volt a dolognak, a wi-fi nem támogatott. Ez egyike volt azoknak a bizonyos tervezési megalkuvásoknak, amivel együtt tudok élni, remélve, hogy ezt később sem fogom megbánni.

A kártyához szükség volt még RAM-ra, a program tárolására szolgáló ROM-ra (FLASH), a hálózati kapcsolat állapotának visszajelzésére szolgáló LED-ekre, illetve mindenféle, ezen elemeket összekapcsoló logikára.

Az egyik ötlet a tervezésnél az volt, hogy az EPNET egy teljesen kibővített gépen és egy módosítatlan alap Enterprise 64-en is használható legyen. Ezért egy függőleges NYÁK mellett döntöttem, az alján lévő élcsatlakozóval, ezáltal egy buszbővítőbe, de az alapgépbe is csatlakoztatható lesz. Így tehát az EPNET közvetlenül az Enterprise-hoz csatlakozna, rendhagyó módon függőlegesen tájolva, így viszont busz bővítővel is használható lenne.

A szoftver oldalon a fő célom, hogy képes legyen betölteni és kimenteni programokat a hálózaton keresztül. Természetesen jó lenne egyéb dolgokat is implementálni, mint például e-mailek olvasásának lehetőségét, vagy akár web oldalak szöveges módon való megjelenítését.

Jelenlegi állapot

Az Eagle CAD segítségével terveztem a NYÁK-ot, melyet a "Seed Studio"

nevű kínai vállalkozással gyártattam le öt példányban. Jelenleg egyet építettem meg, és 12 Kbyte-nyi szoftvert írtam hozzá (tisztán assembly-ben). Tervezem a többi négy elkészítését is, amelyet így tesztelésre, illetve szoftver fejlesztés elősegítésére (pl.: SymbOS rendszerhez) el tudok majd küldeni különböző embereknek.

A szoftver három különböző EXOS eszköz implementál három protokollhoz, melyek az FTP: a HTTP: és a TCP:

Az FTP: eszköz lehetővé teszi fájlok és programok olvasását és írását egy FTP szerver viszonylatában, illetve egy sor FTP parancs végrehajtását mindenféle fájl funkciók eléréséhez. Például az :FTP DIR kilistázza a könyvtár tartalmát, az :FTP DEL töröl egy fájlt, az :FTP MD létrehoz egy könyvtárat, az :FTP CD pedig megváltoztatja az aktuális könyvtárat. Igen egyszerű egy (pl ingyenes) FTP szerver futtatni pl egy Windows-os vagy Linux-os PC-n, és ez talán akkor a legegyszerűbb módszer arra, hogy LAN-on át az ember programokat töltsön be, vagy mentsen ki.

A HTTP: eszköz lehetővé teszi fájlok és programok beolvasását egy HTTP (azaz: webes) szerverről és talán ez lenne a legjobb módja annak, hogy egy sor Enterprise programot tegyünk elérhetővé mások számára. Megint csak könnyű beszerezni egy ingyenes web szervert, ami Windows vagy Linux alatt fut. Ellentétben FTP-vel, a HTTP protokoll nem tesz lehetővé más fájl műveleteket például egy fájl törlését, vagy egy könyvtár tartalmának megjelenítését.

A TCP: eszköz önmagában nem alkalmazható fájl szinten, de igény szerint, például akár egy e-mailt olvasó is implementálható segítségével. Elegendő csupán egy csatornát megnyitni a „TCP:<hoszt>:port>” névvel, ezek után a csatorna simán olvasható és írható.

Az EPNET tartalmaz néhány ":" parancsot. Ezek röviden a következők:

:NET DIAG

Ha probléma adódna, a NET DIAG inicializálja az EPNET-et és a w5300-at, lépésről-lépésre diagnosztikai üzenetek kíséretében. Továbbá elvégez néhány extra ellenőrzést is, mint például a w5300 memóriájának tesztelése.

:NET STATUS

Kiírja az EPNET-hez rendelt egyedi MAC- és IP-címet, stb. A MAC-

cím fix és be van építve az EPNET-be, az IP-cím és a kapcsolódó paraméterek (alhálózati maszk és az alapértelmezett átjáró IP-címe) a rendszer indításkor kerül lekérdezésre a DHCP-protokoll segítségével.

:PING<hoszt> vagy :NET PING <hoszt>

A hálózati kapcsolat tesztelésére használható egy másik a hálózaton lévő számítógép viszonylatában. Egy "echo request"-et küld el a megadott <hoszt> címre, majd válaszra vár, és aztán ezt ismétli meg öt alkalommal. A :PING esetleg ütközhet a jól ismert BASIC hangadó parancsot elvégző kiterjesztéssel, melyet szintén :PING -nek hívnak. Ebben az esetben az EPNET :PING parancsa ebben a formában nem érhető el.

:FTP LOGIN<hoszt>

Bejelentkezik egy távoli FTP kiszolgálóra, amire a további EXOS FTP: eszköz és :FTP parancsok vonatkozni fognak. Például a LOAD "FTP.myfile.bas" BASIC-ben betölt egy programot, az :FTP DIR kilistázza az aktuális könyvtár tartalmát a kérdéses szerver viszonylatában.

:FTP LOGOUT

Kijelentkezés az aktuális FTP szerverről, amire korábban a bejelentkezés történt.

:FTP STATUS

Az aktuális FTP szerver állapotát jeleníti meg.

:FTP DIR vagy :FTP DIR <dir> vagy :FTP DIR <file>

Az aktuális (vagy a megadott) könyvtár tartalmát jeleníti meg az aktuális FTP kiszolgálón.

:FTP DEL <file>

Töröl egy fájlt az aktuális FTP kiszolgálóról.

:FTP MD <dir> és :FTP RD <dir>

Létrehoz, illetve töröl egy alkönyvtárat.

:FTP CD vagy :FTP CD <dir>

Megjeleníti vagy megváltoztatja az aktuális könyvtárat az aktuális FTP kiszolgálón.

Ahogy látható, az összes FTP parancs úgy lett definiálva, hogy amennyi-

Az EPNET kártya az EP buszbővítőre dugva



hibáknak a javításáról szólna, viszont számtalan kérdést kaptam már, hogy van-e lehetőség általános memóriabővítésre alkalmas RAM elhelyezésére a kártyán. Valójában egy 512K-s RAM IC fizikailag nem foglal több helyet, és ráadásul manapság csak egy hangányit drágább, mint a 128K-s verzió. Viszont a memóriadekódoló áramkör emiatt változna. Arra is rájöttem, hogy igen egyszerű lenne egy Compact Flash csatlakozót biztosítani adat tárolási célra. Annak lehetősége, hogy egyetlen kártya képes több memóriát, háttértár és

re csak lehet, megfeleljen az ezekkel egyenértékű EXDOSparancsoknak.

:NET TIME

Aktualizálja az Enterprise által használt dátumot és időt a hálózatról, az NTP protokoll használatával.

:NET TRACE

"Nyomkövető" módba kapcsol, mely hatására protokoll diagnosztikai információk kerülnek kijelzésre. Például az FTP és HTTP szerverek válaszai kiírása kerülnek, ezzel segítve a hibakeresést. A "raw" mód szintén bekapcsolható, ekkor minden hálózati csomag küldésénél és vételénél megjelennek az azokban lévő byte-ok, ám ez igen sok információ képernyőre írását fogja jelenteni!

Problémák

Jelenleg azzal van némi problémám, hogy a Z80 jelei "zajosnak" tűnnek. A továbblépés előtt ennek a problémának a mélyére kell majd ásnom. Van néhány hiba magában a w5300-ban, amiket meg kell kerülnöm, de remélhetőleg már az utolsónál tartok! Az EXOS okoz némi fejfájást az általa limitált fájlnev hossz és a fájlnevben megengedett karakterek kapcsán. Például a szokásos megoldás egy IP-cím és port (hoszt:port) megadására a 192.168.0.1:80 lenne, ámde az itt szükséges kettőspontot az EXOS nem

engedi meg. Ennek a problémának kikerülésére még ki kell találnom valamit.

A szoftver 80%-ban már kész egy első verzióhoz - egyértelműen kielégíti a kitűzött elsődleges célokat, azaz a hálózaton át való program betöltés (LOAD) és mentés (SAVE) lehetőségét, azonban van még néhány elvarratlan szál, ami kapcsán rendet kell tennem.

Tervek

A célom egy stabil hardver megtervezése, aztán egy nagyobb tétel NYÁK legyártatása. Ezek után megépítem ezeket, majd felkínálom eladásra a NYÁK, plusz az alkatrészek költségén.

Az EPNET-en csak szabadidőmben dolgozom, ezért aggódok, hogy képes leszek-e teljesíteni az igényeket. A kínai Seeed Studio összeszerelést is vállal, melynek utána fogok még nézni, de ez természetesen növelné az árat.

A kapcsolási rajz, a NYÁK tervek és az összes szoftver forráskódja elérhető a GitHub-on a github.com/BruceTanner/EPNET címen, így bárki megépítheti és/vagy módosíthatja a saját EPNET-ét.

Távlati lehetőségek

A jelenlegi NYÁK-on pár kisebb problémát még javítanom kell. Ideális esetben a második verzió csak ezeknek a

hálózat elérés lehetőséget is biztosítani akár egy Enterprise 64-nek is, nehéz ellenállni.

Jelenleg az összes memóriadekódolás diszkrét 74HCT logikával történik, hogy megfeleljen az EPNET-el kapcsolatos céljaimnak, miszerint ez EPNET kb. az EXDOS kártyához hasonló legyen, amolyan „hobby barát” megoldásokkal. Azonban lehet, hogy egy méreteiben sokkal kisebb és rugalmasabb megoldást kapnék, ha modern, programozható CPLD típusú eszközt használnék. Hosszabb távon jó lenne, ha az EXDOS is a kártyán kapna helyet, és integrálva lenne a hálózattal. Ezt úgy képezem el, hogy egy adott FTP szerver elérhető lenne meghajtóként, tehát minden EXDOS parancs (mint például a COPY és a DIR) működne helyi és hálózati fájlokkal is. Ez azonban néhány igen komoly módosítást igényelne az EXDOS-ban.

Elérhetőség

Az **Enterprise Forever** fórumon (www.enterprise-forever.com) található az EPNET téma, az angol "hardver" (Hardware) szekcióban. A felhasználói nevem **BruceTanner** (jelen sorok írásának idején a "tagok" link második oldalán), és szívesen váltok üzeneteket. Sajnos nem beszélek magyarul, de a Google fordító általában használható eredményt ad.

EnterMice



Írta: **Maciej Gruszecki (pear)**

Az EnterMice PS/2 egér és joystick illesztőt lengyelországból Maciej Gruszecki (pear) tervezte Enterprise 64/128-as számítógéphez. Pear sokat köszönhet a spanyol GFloreznek (ő ösztönözte az EnterMice tervezésére), valamint Németh Zoltánnak (Zozo), aki az első tesztelője volt a kártyának.

Az EnterMice nagyon szép, finom egérmozgásra képes, tesztelve lett hagyományos golyós, és optikai egérrel is. Működik akár 10MHz-es géppel is, az eredeti módosítás nélküli MOUSE.XR-el!

Mindkét joystickhoz van teljes értékű (azaz autofire kompatibilis) joy átalakító a kártyán, valamint PS/2 egér port.

Beépített mikrokontroller alakítja át az egeret MSX módra, azaz a végeredmény hasonló lesz az eredeti NEOS egérhez. A különbség az, hogy kihasználva a controll port lehetőségeit, másik bemenetre lett téve az egér, így egyszerre használható a joy és az egér, valamint nem örülnek meg a külső joyt használó programok (Például az EPDOS) az egértől. Az egér használható lesz joy módban is, ekkor átveszi a Joy1 helyét. Az egér akár menet közben is csatlakoztatható, ezt még a PC se tudja PS/2 porton...

Általános leírás:

Ne csatlakoztassuk az interface-t, amikor a számítógép be van kapcsolva!

Ha csatlakoztatás után bekapcsolásakor a számítógép nem megfelelően indul vagy szokatlanul viselkedik, azonnal ki kell kapcsolni!

Ellenőrizzük, hogy az interface helyesen van-e csatlakoztatva és próbáljuk újra!

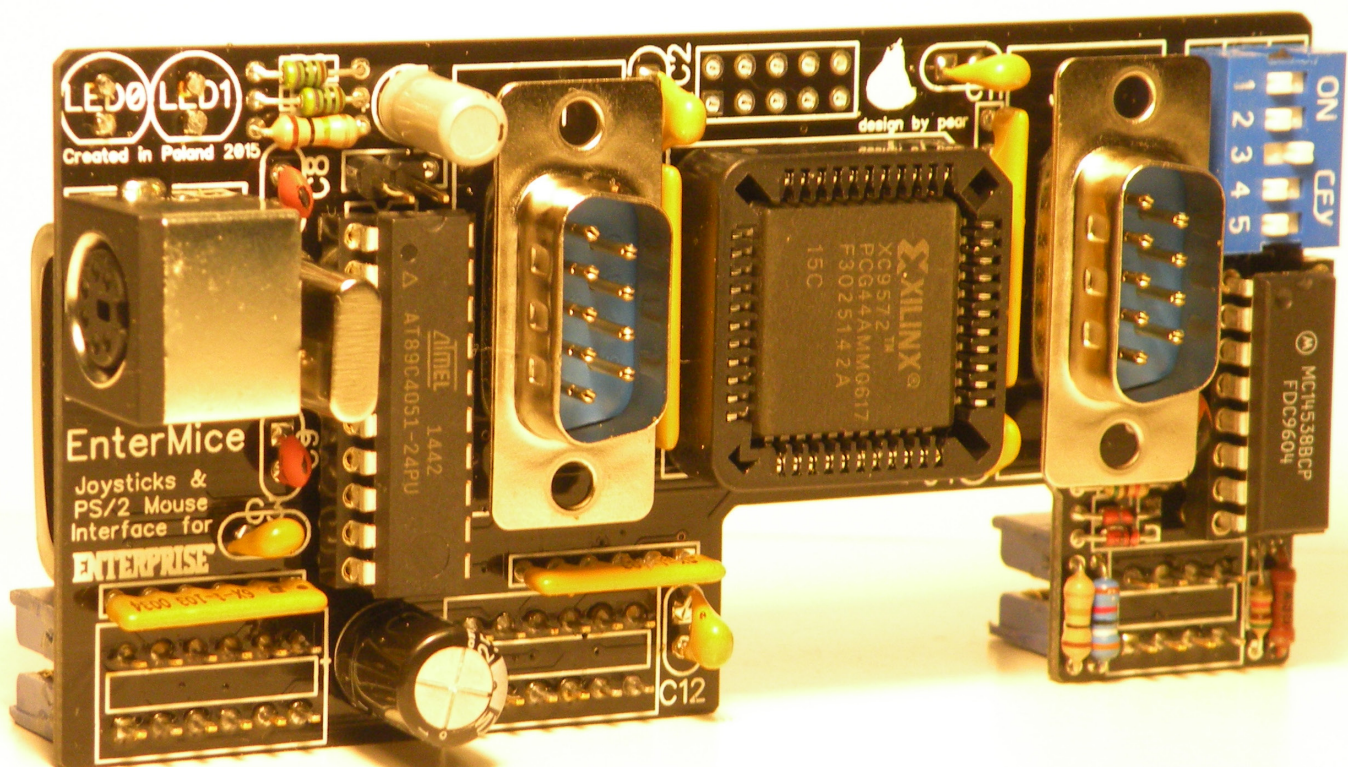
Bár az egér és a botkormányok csatlakoztathatók és kihúzhatóak akkor is, amikor a számítógép be van kapcsolva, ez nem ajánlott. Biztonságosabb mindenféle csatlakoztatást kikapcsolt állapotban elvégezni.

Miután a kártyát csatlakoztattuk, kapcsoljuk be a gépet! A kártya egy öntesztet hajt végre, ezt jelzi a piros és zöld led háromszori felvillanása (feltéve ha a kártyán lévő DIP-kapcsoló 3-as kapcsolója ON állásban van).

Driver

Az EnterMice kártya drivere a mouse.xr mely innen tölthető le:

<https://enterpriseforever.com/programming/universal-mouse-driver/?action=dlattach;attach=14686>



Működési módok

A működési módokat a kártyán elhelyezett 5 db DIP kapcsolóval állíthatjuk.

DS1	DS2	Működési módok
ON	ON	EnterMice egérmód
OFF	ON	BoxSoft kompatibilis egérmód. Az egér bal és jobb gombja fordított állásban fog működni.
OFF	OFF	EnterMice joystick mód (Az egérmozgás az 1-es joystick portra van irányítva, így ez a port nem használható)
ON	OFF	nem használt

DS3	LED státusz
OFF	LED státusz kikapcsolva
ON	LED státusz bekapcsolva

Joystick emulációs mód beállítása

DS4	Érzékenyég
OFF	Normál
ON	Magas

DS5	Diagonal correction
OFF	Tiltás
ON	Engedélyezés

Az egérrel jelenleg használható felhasználói programok

A Paintbox egérvezérlős verziója

- Az Enterprise Graphical Interface egérvezérlős verziója
- SymbOS Boxsoft és most már EnterMice módban is
- Az egérvezérlővel Basic programok használhatják az egeret.
- EDC Windows Gflorez által módosított alkalmazása
- SPEmu128 (geco Spectrum Emulátora Enterprise-ra)
- ep128emu 2.0.10 beta verziója

Az egérrel jelenleg használható játékprogramok

Paszíánsz (Solitaire)
Swap
Chess Master 2000

XEP128 Emulátor

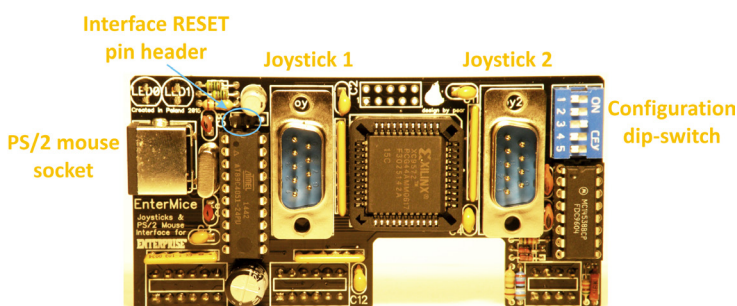
Ha nincs EnterMice adaptered, akkor sem kell kimaradnod az egeres Enterprise használatának élményéből! LGB kiváló XEP128 emulátora biztosítja az élményt PC egéren keresztül.

Egy kis basic program a teszteléshez:

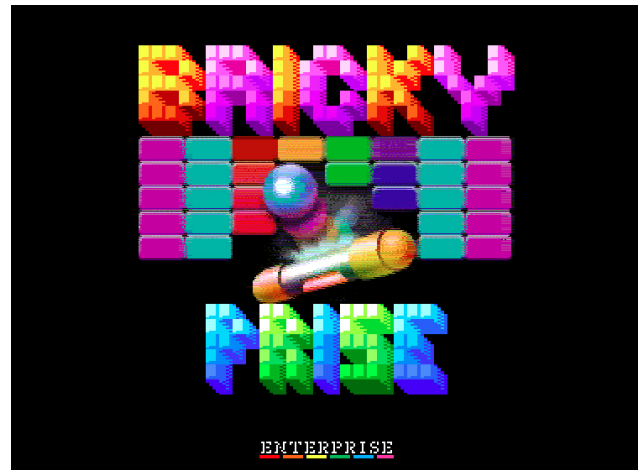
```

100 PROGRAM „ms_test.bas”
110 NUMERIC X,V,MAX_Y”,X_COUNT,LASTX,LASTY
120 GRAPHICS
130 CALL MOUSE_SETUP
130 OPEN £1:”mouse:” ! Open a channel to the Mouse
140 DO
15D CALL MOUSE_POS
160 LOOK 1:FIRE ! Check the fire button
170 IF FIRE THEN
130 PRINT £1:”o”; ! Turn the pointer off
190 PLOT LASTX,LASTY,X,Y
200 PRINT 1:”O”; ! Turn the pointer on
210 ELSE
110 PLOT X,Y,
230 END IF
240 LET LASTX=X:LET LASTY=Y
250 LOOP
250 END
270 DEF MOUSE_SETUP
280 LET CHAR_Y=20 ! No. of characters screen high
290 LET X_COUNT=2 ! Pixel calc no
300 SET 180,101 ! Video channel to put
mouse pointer on
310 SET 181,30 ! Position of co-ordinates on status line
320 SET 182,1 ! Show co-ordinates
330 SET 183,255” ! Colour of pointer
310 LET MAX_Y=CHAR_Y*36-2 ! Convert char_y
to co-ordinates
350 END DEF

```



ENTERPRISE PROGRAM Bricky Prise - Persa Noel (Geco)



AZ **ENTERPRISE** FOREVER BEMUTATJA:

Van valami programod?
Legyen Enterprise programod!

ENTERPRISE

DevCompo #1

programozási verseny

LIFE BEGINS AT 30

A verseny kezdete: 2015. november 16., 0:00:00
Leadási határidő, és szavazás kezdete: 2016. május 16., 23:59:59
Szavazás vége: 2016. május 22., 23:59:59
Eredményhirdetés: 2016. május 23.

Kategóriák

- Enterprise program
- Konverzió
- Egyéb

A kategóriák nyertesei az alábbi jutalomban részesülnek:
Kategóriák 1. helyezettje: minimum 150 Euro
A kategóriák 2. és 3. helyezettjei is értékes jutalmakat kapnak.
A kategóriák első három helyezettjei csak **100% BAN KÉSZ** program esetén kaphatják meg a díjakat!

Segítségért nyugodtan fordulj az Enterprise fórum közösségéhez!
<http://www.enterpriseforever.com>

Részvételi feltételek

Egy rövid videó a programról, akár a YouTube-ra feltöltve.
A linket még lehet osztani az Enterprise fórumon, <http://www.enterpriseforever.com>

- pár screenshot a menüről,
- a program működés közben,
- maga a program.

Ezeket mind fel lehet tölteni majd az Enterprise fórum megfelelő topikjába, ha nincs kedved regisztrálni, vagy esetleg valami probléma merül fel a regisztrálás során, küldheted az alábbi e-mailre is: persa75@freemail.hu és mi feltöltjük, csak ebben az esetben küldj pár infót legalább az alkotókról.
Legalább 80%-ban kész programmal vehetsz részt a viadalon.
Csak olyan programmal lehet nevezni, amely nem volt eddig publikus Enterprise gépen!

Ha konverzióval indulsz, akkor annak tartalmaznia kell valami Enterprise specifikus dolgot is.
Más programokból felhasznált dolgok megengedettek, csak ezt jelezd a programban.
Egy programozó/csoport több programmal is indulhat.

A programnak bármilyen hardver vagy szoftver bővítéssel rendelkező Enterprise gépen futnia kell, ami megfelel a program alapkövetelményeinek.
A program futásával kapcsolatos további követelmények és a specifikus dolgok részletes leírásai az alábbi linken, valamint az Enterprise fórumon találhatóak!

Visszaszámlálás indul! 42 ... 30 ...
Írj programot a 30 éves ENTERPRISE-ra!



KONVERZIÓ Lirus - Németh Zoltán (Zozosoft)



Az Enterprise DevCompo#1 végeredménye:
Here is the final result of Enterprise DevCompo#1:

Basic:	
Wall Defender játék:	7,789473684
Bomber 3in1:	7,1
FlapFlap játék:	6,263157895
Jumprise játék:	6,210526316
Snaki:	6,105263158
conversion:	
LIRUS:	8,526315789
Filler:	8,368421053
Pacman:	7,75
Chase HQ:	7,631578947
ZenLoops:	7,473684211
Enterprise program:	
Bricky Prise:	8,894736842
Team Hat Trick aka Six Men One Puck:	8,631578947
Flora:	6,526315789

Mindenkinek köszönjük szépen a részvételt, külön köszönet gOblinishnek, aki nem EP-sként csatlakozott, és 3 nagyon jó programot köszönhetünk neki.
Thanks to everybody for joining, special thanks to gOblinish who met the Enterprise the first time and made 3 very good programs for us.

BASIC Wall Defender - Baráth Endre (Endi)



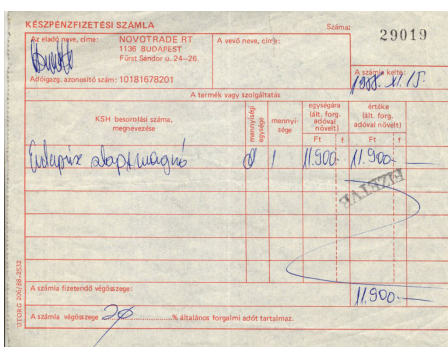
**A GYŐZTESEKNEK
GRATULÁLUNK!**

Hogyan lettem Enterprise programozó?

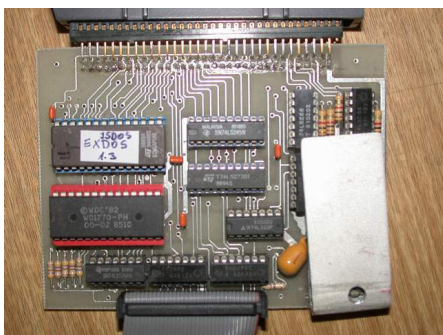


Írta: Povázszy Zoltán (Poví)

Az Enterprise számítógépünket 1988 novemberében vettük. itt a számlája :-)



2001-ben, vagy 2002-ben vettem egy EXDOS-kártyát Bihari Misitől 7000 Ft-ért. Itt egy fénykép róla.



Ekkor egyetemista voltam az ELTE-n, geográfus szakon. A meteorológiai tanszék könyvtárából kivettem a Sztrókay Kálmán féle Z80 Assembly könyvet HT1080Z-re, ebből tanultam meg a Z80 assembly-t. Illetve volt egy Tetris Spectrum-ra, (Spectris) ami a 2002-es MiniGame Compo-n 8. helyezett lett, 1kB-os kategóriában, és ennek megvolt a dokumentált asm forráskódja.

Gondoltam, hogy átírom EP-re a SPV-ben megjelent cikk alapján, közben meg megtanulom a Z80 assembly-t. Így is lett, Zozo segítségével pár évvel később meg EXOS-kompatibilissá lett téve :-)

Ennek a sikerén felbuzdulva egy évvel később már én is neveztem a MiniGame

Compo-ra, 2004-ben egy 50 pályás Sokoban-t csináltam 4kB-ban, 2005-ben pedig egy 1kB-os Nibbles-t, kilenc pályával.

A Sokoban 16 színes grafikus képernyőn fut, Paintbrush-ban rajzoltam az „ikonokat”, a Nibbles pedig karakteres képernyőn fut.

Zene egyik játékban sincs, a Nibbles-be később beleraktam ugyanazt a zenét, ami EDC-nek a Nibbles-ében is volt, így már 3kB-ra hízott a játék :-)

A legkomolyabb játékomat, 2005 karácsonyán kezdtem el, és 2006-ban lett kész, ez az Atomix. Ez is 16 színű grafikát használ, a kezdőképernyő kivételével (ami a PC-s változattól lett átvéve), pixelre megegyezik a C64 változattal. Zene sajnos itt sincs.

A pályaadatok a PC-s változattól van kiszedve, ott egy külön fájlba voltak tárolva, így könnyű volt kinyerni. A grafikát pedig PC-n futtatott C64 emulátorból szedtem, majd a pálya-akoltó elemeket paint-ben kimentve írtam egy Pascal progit, ami átírta EP-formátumúra. Vagyis végig kellett játszaniom C64 emun a játékot, hogy minden pálya grafikai eleme meglegyen (nem értek a C64 grafikához, ezért nem a fájlból szedtem ki).

Felhasználói progik

Altair BASIC - erről a logout-ról olvashatsz részletesebben:
http://logout.hu/cikk/negyven_eves_az_altair_basic_a_microsoft_elfo_term/bevezetes.html

Brainfuck fordító

```
Brainfuck Compiler
PovSoft Brainfuck Compiler v1.4
Available commands:
Load source file
Compile
Run
Help
Quit

>L
File name: hello.bf
006FH bytes loaded.

>C
Compiled to 4000-405C

>R
Hello World!

>
```

http://povi.uw.hu/brainfuck_compiler_z80.html

Életjáték

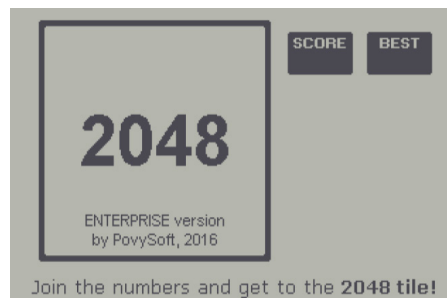


http://ep.lgb.hu/jsep/demo/?-disk=http%3A%2F%2Fpovi.uw.hu%2Fgame_of_life%2Fgame.com&diskhack=load&autostart=yes&mem=128
 karakteres képernyőn fut, van néhány előre beépített kezdőminta (F2-F7, F8 : kilépés, F1: random minta)

SCELBAL

A SCELBI egy Intel 8008 alapú gép volt, amire kiadtak egy BASIC-et, ez volt a SCELBAL. A program assembly forráskódja (full dokumentált) egy 50\$-os könyvben jelent meg, így ez volt az első open source progik :-)

Mivel a Z80 nagyon hasonló az Intel 8008-hoz (minden intel 8008 utasításnak van Z80 megfelelője), ezért nagyon egyszerű volt emut írni hozzá. Az emulált 8008 regiszterei a Z80 vesszős regisztereiben vannak eltárolva.



2048

A 2048 számítógépes és mobilos játék sikerén felbuzdulva arra gondoltam, hogy megírom a 2048 című játékot Enterprise gépre is. A játék gépi kódban készült.

10 Mhz Enterprise-on



Írta: Németh Zoltán
(Zozosoft)

Van olyan Enterprise program ami-nél van értelme a turbó funkciónak?

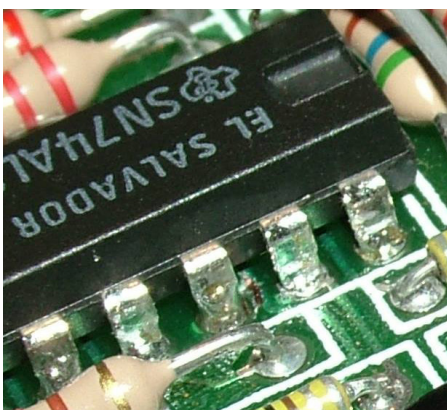
A felsorolást kezdhethük az IS-BASIC-kel. Meg aztán assembler, stb. Rengeteg játék is van, ahol jól jön egy kis plusz tempó. Külön érdekesség a Race Ace ahol csak a te autód gyorsul, a gépe nem (azok a videó megszakításhoz vannak szinkronizálva). Az utóbbi években készült programoknál már előfordulnak turbós extrák, például több fps-t kirajzoló mód, vagy jobb digi hang lejátszás. És persze ott van a SymbOS is mint multitasking Z80 operációs rendszer.

Melegszik a Z80 tőle, vagy úgy kezdődik a mod hogy CPU csere?

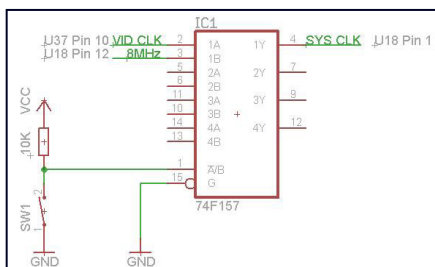
Ha 10MHz-et akarsz, ahhoz mindenképpen kell a csere. 6-ot elbír még a legtöbb Z80A, 7.12-öt is még sok, főleg a Zilog gyártmányúak. Az új CMOS Z80 meg nem melegszik. De nem Kínából kell venni...

Technikai megvalósítás

Le kell vágni a 8MHz-et a rendszeróra jel vezetékéről, ehhez az U18 12-es lábánál kell vágni:



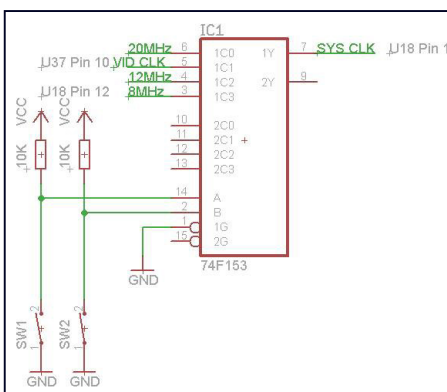
A legegyszerűbb megoldás a videó órajel használata, így nem kell plusz órajel generátor. Ez esetben jön ki a 7.12 MHz-es órajel. Ebben az esetben egy 74F157 lesz a kapcsoló IC. (De persze külön órajel generátorral lehet más órajel is.) 7.12-es tuningnál elsőként be szoktam kötni direktben a videó órajelet az U18 1-es lábra, megnézni, hogy az adott Z80 bírja-e, kell-e csere.



Mit csinál mikor nem bírja? Szemeltel vagy el se indul?

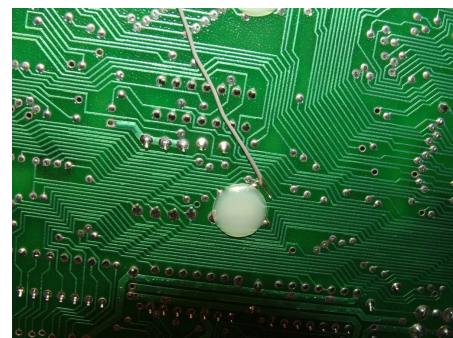
El se indul (ez esetben a bekapcsoláskori memóriaszemét lehet a képernyőn), vagy hamar lefagy.

4 sebességes turbó-hoz kell két plusz órajel generátor (kreatívan elszórva :-), és ez esetben 74F153 a kapcsoló IC.

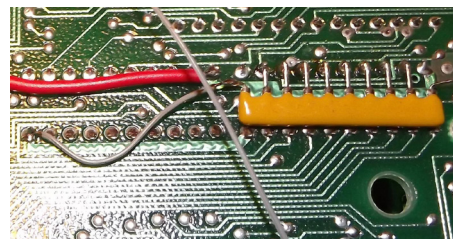


További lényeges dolog: alaplapi memória időzítéshez R12-t rövidzárba cserélni, C7-et pedig kivenni. A felső 64K panelt 10MHz-hez ki kell cserélni modern SRAM-os bővítőre, az eredeti DRAM-okkal 7.12MHz-ig el lehet jutni, R150 és R151 értékét kell csökkenteni, esetleg rövidebbre zárni.

Ha EXDOS kártyát is akarunk használni, akkor gondoskodni kell róla, hogy a busz csatlakozóra az eredeti 8MHz menjen ki, a képen látható helyen kell vágni, és az U18 12-es lábáról odavezetni a 8MHz-et (szürke drót).

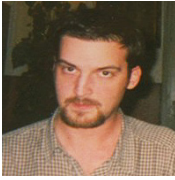


További apróság, ami egyes alapgépeken is előjön: az EP-ben nincs felhúzó ellenállás az adatbuszon, így tulajdonképpen semmi nem biztosítja, hogy nem használt címről FFh jöjjön. Ez okozhatja fantom hibás RAM szegmensek megjelenését a RAM tesztben, és ami nagyobb gond: egyes játékok lefagyását (nem teljesen kitöltött IM2 táblát használóknak). Ez némely alapgépen is előjön, de turbó és/vagy CMOS Z80 használatával sokkal nagyobb az esélye. Ezért célszerű egy 4.7K ellenálláshidat berakni felhúzó ellenállásnak. Erre legcélszerűbb hely az U7 2-9-es lábai.



A 7.12Mhz az még megy az eredeti RAM-mal, és sok esetben az eredeti Z80A procival (Zilogok bírni szokták, NEC, SGS az általában csak 6MHz-ig). Sőt anno még kapcsoló IC se volt, csak parasztosan ráraktam az órajel drótra a kapcsolót és kész. Ez esetben tényleg 1 vágás és 3 drót volt a turbó, és még 1 vágás meg még 1 drót az EXDOS miatt.

Hogyan lettem Enterprise programozó? #2



Írta: Persa Noel
(geco)

Persa Noel (Geco nick nevű), szoftverfejlesztésben meglehetősen aktív fórumtagunkat kérdeztem az Enterprise-hoz fűződő viszonyáról és az általa készített AMSTRAD CPC (2006), TVC (2008), VIC-20 (2009), ZX81 (2010), ZxSpectrum 48K (2014), ZxSpectrum 128 (2015) Emulátorokról.

1987 karácsonyára vették meg a szülők az Enterprise 128-at családi használatra, valahogy még is úgy alakult, hogy a használója én lettem. Később kiderült az is, hogy nagymamám is meglepetésre készült. Vett nekem egy Commodore 64-et, csak ő később szerette volna ellőni a puskaport. Így az Enterprise 128-at követően háromnegyed évvel megkaptam a C64-et is. A C64-en játszottam sokkal többet, de az Enterprise-t is nyúztuk rendszeresen, és hiába volt több, szebb játékom C64-re, az EP lett a szívem csücske. Ezt a mai napig sem tudom megmagyarázni, lehet a kulcsín az oka, vagy az, hogy a Basic-jében lehetett programozni, és nagyon sok dolgot megvalósítani. Már akkor is úgy éreztem, hogy sokkal több van benne, mint amit addig láttam (a nagyrészt speccy átíratok révén).

A Basic után, meg szerettem volna tanulni gépi kódban programozni, de a kisvárosban nem volt olyan Enterprise-os ismerősöm (sőt Z80-as se), aki ebben segítséget tudott volna nyújtani. Az elérhető irodalom pedig kimerült a Spectrum világ folyóiratban. Igaz az EXOS műszaki leírás megvolt, de akkor nem igazán értem fel ésszel. Első hamvába holt próbálkozásom a Spectrum világban leírt Mooncrest konvertálás volt, bármiféle előképzettség nélkül. Ráadásul Spectrum világos cikksorozat egyik része hiányzott.

Ezután egy hosszabb szünetet következett, 2002 környékén kezdtem el újra, akkor már jobban felfegyverkezve, az összes Enterpress magazint letöltve estem neki az újabb próbálkozásnak.

A 90-es évek végén összefutottam a neten az Amstrad CPC elnevezésű, addig számomra teljesen ismeretlen csodával. Ráadásul véletlenül észrevettem, hogy a CPC-s, és az EP-s Airwolf kódja majdnem bitre ugyanaz, elkezdett foglalkoztatni a dolog.

Elő is vettem az AlienAttack című játékot, mivel ez egy ROM hívós program volt, elkezdtem saját rutinokat barkácsolni, és azokra lecserélni a CPC ROM rutinokat, ha jól emlékszem a hangnál akadtam el, és ekkor osonó pályán eljutottam az Exolonhoz, amibe alaposan beleenyúltam, mert így utólag azt mondom, hogy az átírás legkönnyebb

kategóriájába esik. A kezdeti örömet a konverziós siker fölött Zozó „alaposan letörte”, felhívta a figyelmemet arra, hogy a program nem EXOS kompatibilis, természetesen én a favigó módszerrel estem neki a konvertálásnak, csak úgy, mint az átírók 90%-a régen.

Azt se tudtam mit takar az EXOS kompatibilis, Zozó megosztotta a betöltőjét, segítséget nyújtva a program javításához, azóta is hálás vagyok neki, hogy felhívta a figyelmem rá, és mentőövet is dobott.



Először csak annyi volt a motivációm, hogy a tonnányi Speccy átírat mellett legyen elérhető több szebb program is, ami az EP képességeit is jobban kihasználja. Aztán az idő múlásával megpróbáltam némi extrát is belevinni az átíratokba. Szebbé tenni azokat mint az eredeti, amiben nagy segítséget nyújtottak István által fejlesztett EPIMGCONV, és EPCOMPRESS/DTF utilityk.

Játék konverziók közben jutott eszembe, hogy EP-re van jó pár Speccy szoftveres emulátor. Ezen felbuzdulva kezdtem el fejleszteni a CPC emulátort. Majd jött a folytatás, a Speccy konverziók közben támadt az ötlet, hogy lehetne egy hatékonyabb szoftveres Speccy emulátort írni, ami betöltés után konvertálja a kódot. Ez volt az alap ötletem. Semmi más konkrétum nem volt még akkor,



amikor elővettem a 48K-s Spectrum ROM forrását, és először csak működésképpé tettem EP-n. Ez ugye csak annyit tudott, mint a régi emulátorok. Ezután jött a Speccy betöltő rész módosítása, egyrészt választható Speccy/ EP EXOS betöltés, és a betöltés után ne kerüljön vissza a futás a szokásos ROM rutinra, hanem a kód ellenőrző rutin fusson le rá, ami megkeresi az előre definiált bájtsorozatokot, és cseréli azokat az emulált verziókra. Első elképzelés szerint az 0FEH, és 0FFH port olvasó, és író utasításokat.

Természetesen nem az összeset, mivel az in/out utasítások előtti, vagy utáni utasításokat is ellenőrzi a program, hogy nehogyan véletlenül adatot cseréljen, aminek az értéke megegyezik az in/out utasításokkal, ezért fordulhat elő, hogy némely programban nem működik a billentyűzet, mert nem implementált utasításkombináció található a programban. Ezzel már pár Speccy program el is indult, itt futottam bele az első nagyobb problémába. A Speccy játékok nagy része IM2-ben fut, és az általam elkövetett galádsággal csak az IM1-es programok működnek, hogy lehetne áthidalni ezt a problémát is? Az volt a gond fő okozója, hogy EP-n a megszakítási rutinban a megszakítási tárolót törölni kell, különben végtelen megszakítási ciklusba kerülünk. Ha ez nem lenne így, akkor az IM2 se

okozott volna fejtörést, nem is kellett volna foglalkozni vele. Hogy is lehetne ezt megoldani? Először arra gondoltam, hogy az IM2-es utasításokat cserélem le, de ez az ötlet megbukott.

Helyette az LD I,A utasítások lettek lecserélve, ennek segítségével lehetett iktatni az eredeti megszakítás elé az EP-n szükséges megszakítási tároló törlését. Viszont itt is jelentkezett némi probléma, volt olyan program, ami hamarabb került IM2-es megszakításba EP-n, minthogy az I regiszter be lett volna állítva, vagy épp elkészült volna az ugrótábla. Ezek után jött a következő Speccys érdekes megoldás, az I regiszterrel az ugró táblát a ROM-ba, vagy 0FFFH címre állítja jó pár program, mert speccyn az ugrótábla két utolsó eleme van csak használva. Ezzel újabb fejtörést okozva, de végül ezen problémákra is sikerült megoldást találni, így újabb programokkal bővült az emulátoron futtatható programok száma.

A következőkben a nem szokványos ROM hívással töltő programok okoztak fejtörést. Itt is némi keresgélés után sikerült olyan megoldást találnom, ami az ilyen betöltők hibáinak nagy részét orvosolta. A viszontagságos út végén elkészült egy olyan Spectrum emulátor, amin a Speccys szoftverek 40%-60%-a már fut.

TVC emulátorról annyit érdemes tudni, hogy most készül a továbbfejlesztett változata, ami ugyanolyan kódcsereléses változat lesz, mint a Speccy emulátor.

A VIC-20 emulátort Simon Owen írta még SAM Coupé-ra, azt konvertáltam át. Közben egy kis átalakítást is végeztem rajta. Így sebességben közeledett a SAM Coupés verzióhoz, ami 6MHz-en ketyeg. Az eredeti SAM verzióban nem volt fájlválasztási lehetőség. A futtatandó programot mindig be kellett fordítani az emulátorral, ezzel is kényelmesebb helyzetben vagyunk, mint a SAM felhasználók.

Lacika összehasonlító tesztje:

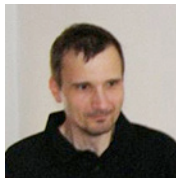
```
1 FOR I=1 TO 255
2 PRINT I
3 NEXT I
```

VIC-20	0:07
Enterprise (6MHz)	0:46
SAM Coupé	1:09
Enterprise (4MHz)	1:12
Spectrum	1:43

A ZX-81 emu is konverzió, 1997-ben adta ki YRS (nem tudom kiket takar) Spectrumra. A kódot disassembláltam, csináltam belőle egy fordítható kódot, majd EP-sítettem. Ez is átalakításokon esett át, pl a képernyő EP-n egy hardveres karakteres képernyő, ezért is jóval gyorsabb, mint a Spectrum és az itt leírt speccystologatósdi is meg lett szüntetve, ami úgyszintén sebesség növekedéssel járt (http://www.ep128.hu/Ep_Util/ZX81_Emulator.htm)

Ez a VIC20emuhoz hasonlóan az emulátor kategóriába tartozik, mivel a Z80 utasításkészlet is emulálva van, egy kicsikét furcsán hangzik, de így van. Az emulátor a DE regisztert használja PC-ként, ellenőrzi milyen utasítás van az adott címen, ráugrik az utasításnak megfelelő kódra, -s regiszter készletről vált a simára, végrehajtja a műveletet (a sima regiszterkészlet a zx81 által használt regiszterkészlet), majd visszavált a -s regiszterkészletre, és ellenőrzi a következő utasítást.

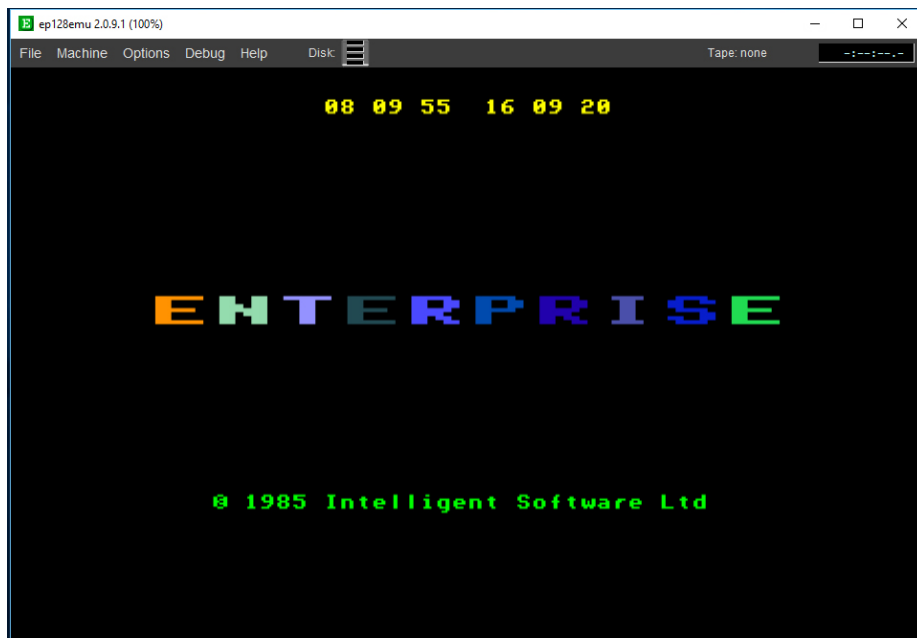
Ep128emu – Enterprise emulátor - IstvanV



Írta: Kiss Iászló
- ep128.hu

A „mezőnyhöz” később csatlakozott emulátor, de kinőve gyermekbetegségeit mára nagyon jól használható eszköz lett belőle! Kompatibilitása még az Ep32-t is felülmúlja, olyan érzés, mintha egy Enterprise-t rejtettünk volna a PC dobozába. A debugger része nagyon jó! A program a gyorsabb futás érdekében használja az OpenGL-t, de ha valakinek nincs telepítve a gépére, software-es módban is indíthatja a programot. Az OpenGL módban gyorsabb a kép frissítése, és néhány a „Display/Configure...” ablakban engedélyezhető effektus csak ebben a módban használható. Viszont a szoftveres mód minden gépen fut (az OpenGL esetén problémák lehetnek egyes video driverekkel), és valamivel pontosabban a színek (az OpenGL mód 16 bites textúrákat használ). A régebbi verzióktól eltérően a ROM fájlokat nem kell 16K-s darabokra szétvágni, így valamennyi letölthető ROM-ot tudunk az emulátorral használni! A program különlegessége, hogy emulálja a Zozosoft féle Spectrum emulátort és az IDE vezérlő kártyát! A 2.0.8-as verziótól pedig már Spectrum és CPC464 emulátort is tartalmaz!

A program elsősorban az image file-ok használatát támogatja, de lemezes rendszerben lehetőségünk van a PC floppy-meghajtóját is használni (hasonlóan az Ep32-höz). Ehhez a disk image kiválasztásánál (OPTIONS / DISK / CONFIGURE, de egyszerűbb az ALT+D billentyűk használata) file



névnek „\\.\A:”-t kell beírni. Ezt nem kell begépelni, az újabb verzióknak van külön gombja az A:, B: meghajtóknak, melynek hatása ugyanez. Az emulátorral „virtuális merevlemezeket” (Virtual Hard Disk) is használhatunk. A „disk” könyvtárban található egy „ide126m.vhd.bz2” file, amely kicsomagolva 4x31.5 MB üres (de formázott) HDD image file-nak használható.

Magnós rendszerben TAP, Spectrumos TAP/TZX, vagy WAV fájlokat tudunk betölteni. TAP készítésre fel lehet használni a ‚tapeedit’ programot is, mely az Ep128emu2 csomag része, és Ep128emu formátumban ment, amit az emulátor jobban támogat, mint az EPTE/TAPir file-okat. A program az EPTE formátumú .tap file-okat is el tudja olvasni, bár így nem minden funkció működik (nem lehet felvenni, és az olvasási pozíciót változtatni is csak korlátozottan - a TAP elejére - lehet). A tapeedit tud EPTE formátumú .tap file-okat olvas-

ni, amelyek így ep128emu formátumra konvertálhatók!

Még egy lehetőség file betöltésre: a file-rendszeren egy könyvtár közvetlenül elérhető (írható és olvasható) az epfileio.rom modul használatával. Ehhez a Machine configuration-ben engedélyezni kell a virtual file I/O-t, és betölteni a fent említett ROM file-t (pl. az EP_128k_Tape_FileIO.cfg konfiguráció használatával, amely a 10 szegmensre tölti). Az OPTIONS / SET WORKING DIRECTORY segítségével beállítható a könyvtár, amely az EP programok számára - a FILE: EXOS eszközzel - láthatóvá válik. A FILE: alapértelmezetté tehető a :def_dev_ file parancs használatával (az újabb verziók ezt megteszik), így a több részből álló programok megtalálják a többi file-t is. Üres file név esetén a file a GUI segítségével választható. Ha valaki olyan konfigurációt használ, amelyben EXDOS és epfileio is van, de a lemezt használja gyakrabban, akkor a :def_dev_disk paranccsal a

	CPU frequency	Video clock frequency	Sound clock frequency
4 MHz	4000000 Hz	889846 Hz	500000 Hz
6 MHz	6000000 Hz	889846 Hz	750000 Hz
7,12 MHz	7118768 Hz	889846 Hz	889846 Hz

floppy tehető alapértelmezett meghajtóvá. (Vagy átrendezni a ROM-okat, hogy az EXDOS legyen alacsonyabb szegmensen.)

Tipp: ha turbósított gépet szeretnének emulálni, a pontos időzítések érdekében az alábbi módon kell beállítani a frekvenciákat a MACHINE / CONFIGURE menüpont GENERAL lapján:

(Ezek az értékek 14237536 Hz-es NICK órajelet tételeznek fel.)

Nem szükséges azonban minden átkapcsolásnál megadni az értékeket! A MACHINE / QUICK CONFIGURATION menüpont alatt elmenthetünk két tetszőleges beállítást (SAVE CONFIG 1 / SAVE CONFIG 2), melyek között a Page Down / Page Up billentyűkkel (vagy a menü a LOAD CONFIG 1 / LOAD CONFIG 2 menüpontjaival) váltogathatunk.

Érdeklőség, hogy a videobeállításokkal játszva «emulálhatunk» korabeli monochrome monitort is a következő beállításokkal

(OPTIONS / DISPLAY / CONFIGURE):
saturation = 0
red contrast / blue contrast = 0.5
red brightness / blue brightness = -0.5 (vagy -0.25 világosabb és kisebb telítettségű zöld színhez)

green gamma = 2.0

green contrast = 1.1 („ízlés szerint” 0,95-1,1 közötti értékekkel lehet próbálkozni)

quality = 3 (csak az Onen GL-t használó verzióban működik)

A program legfrisseb verzióját letölthetjük innen:

<http://ep128emu.enterpriseforever.com/>

Hasznos tudnivalók

Emulátorban az eredeti 1.0-ás, 1.3-as EXDOS verziók nem nagyon voltak használhatóak, mivel az ember PC-n leginkább HD-s lemezt használ, amit igaz, hogy az emulált „EXDOS kártya” simán olvas, de ezek a verziók Not a DOS disk hibát írnak ki. Ezért leginkább az alaposan átbütykölt Turbo EXDOS volt használatban. Ami nem szép ha valaki egy teljesen eredeti gépet akar emulálni, meg lehet, hogy a jelen esetben teljesen felesleges Turbo rutinok lassítják is a működést. Zozosoft megcsinálta, az eredeti ROM-okból is az olyan változatot, ami elfogadja logikailag a HD-s formátumot (sőt ha már, akkor az ED-t is! (2.88-as meghajtó...)) Ez két bájt módosítást jelentett, a 1.0-

ás verzióban 2743H, 277CH címen, 1.3-asokban 2073H, 20ACH címen. A csomagban van eredeti 1.0, ez csak angol, 1.3-as eredeti, ez angol-német, módosított 1.3-as, amibe már benne van az ISDOS is, és a német helyett magyar hibaüzeneteket tud, ebből van HUN és HFONT karakterkészlethez való is.

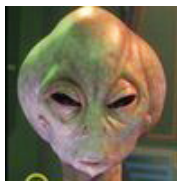
Disk image fájlok

Az Ep128emu2 emulátort a legkönnyebb disk image-file-okkal használni. A mai PC-ben nem is feltétlenül van floppy meghajtó, ebben az esetben nagyon nincs is választási lehetőségünk. Az alább előre összeválogatott „floppy lemezeket” tölthetünk le, melyeket nem csak az Ep128emu-val tudunk használni, hanem az Ep32-vel is. Ebben az esetben viszont célszerű átnevezni a file-ok kiterjesztését .IMZ-re, amit az Ep32 is „lát”. Bosszantó tulajdonsága sajnos az Ep32-nek, hogy minden reset után ki kell választanunk a használni kívánt disk-image-et. Ha a letöltött „lemezeket” „igazi” Enterprise-on szeretnénk használni, könnyen, gyorsan kiírhatjuk őket floppyra is pl. az **FDD Image** programmal.



ENTERPRISE BÖGRE rendelhető az alábbi e-mail címen:
inkedpixelshop@gmail.com

Én, az EP, és emulátor írási kalandjaim



Írta: **Lénárt Gábor**
(lgb)

Előszó

Ez a cikk még régebben íródott, ezért a vége felé némi „aktualizáló” rész következik. Az eredeti szándékom a cikk teljes újírása volt (ami idő hiányában elmaradt), de talán nem is baj, hogy így történt; így legalább folyamatosan jobban látszódnak a történések.

Még kétezer környékén történt, hogy egy szokásos „webes retro körutam” (értsd: céltalan böngészés a Neten „retro computing” témában) során belefutottam az Enterprise nevű masinába. Mivel még nem is hallottam erről a gépről (habár később eszembe jutott, hogy egy általános iskolás osztálytársamnak volt ilyenje, egyszer láttam pár pillanatra), elkezdtem utána olvasni. Azonnal lebilincselőnek tünnek elegáns hardware és software megoldásai, legalábbis legtöbb nyolc bites társához képest. Természetesen ki is akartam próbálni, aminek nálam megszokott módszere (valódi gép hiányában) az emulátor keresése volt. Rá kellett jönnöm, hogy itt akad egy kis probléma: ugyan rátaláltam az „Enter” nevű emulátorra, és talán az „EP32”-re is (nem emlékszem már), egyik se működött az általam használt operációs rendszer (Linux) alatt, Windows-om meg soha nem is volt (a wine – ami egy Windows emulátor szerű entitás – emlékeim szerint nem igazán boldogult akkoriban egyikkel sem). Pár napig bütykölgettem a forráskódjukat, de rájöttem, jobban járok (és többet is tanulok) ha én magam próbálom meg emulátort írni – magamnak. Hamar rákaptam a dologra, és megtetszett az emulátor írás, annak ellenére, hogy voltaképpen használható eredményig el sem jutottam (az Enterprise logó után mindig kifagyott). Viszont ebben saját volt minden, a (félkész) Z80 emuláció is, nulláról írva (és a Z80-at is ekkor ismertem meg közelebbről, addig csak távolról szemezgettem vele). Az egész csak Linux alatt futott, és nem is adtam ki soha (a neve amúgy Xepem lett volna, azaz „X EnterPrise EMulátor”).

A fentiekből látható, hogy kicsit kilógok az EP-s közösségből, nekem ugyanis nem volt régen Enterprise-om (C64-em volt anno), mint a legtöbbünknek, akiket jelenleg érdekel a téma.

Sok minden történt ezután, rátaláltam a magyar Enterprise-közösségre, a fórumra, lett igazi Enterprise-szöveg is (szégyenkezve bevallom, hogy egy-két gyors próbálkozáson kívül még nem igazán használtam – ezen hamarosan változtatni fogok!). Ami viszont

történetünkben legfontosabb, lett ep128emu is (az is elképzelhető, hogy csak én táltam rá viszonylag későn ...), ami futott natív Linux alatt is. Szeretném tisztázni: az ep128emu véleményem szerint a legjobb és leghasznosabb Enterprise emulátor. Az én „próbálkozásaimról” inkább élményszintű beszámolót szeretnék itt írni, nem állítom, hogy az én munkáim annyira komolyak és használhatóak (messze nem!), mint amilyen az ep128emu! Általános célú emulátornak mindenképpen az ep128emu-t ajánlom mindenkinek! Az „első” kiadatlan emulátoromra (Xepem) nem is volt szükség többé, hiszen színre lépett az ep128emu. Ami ráadásul működött is, ami nem kis előny :)

Már 2013-ban történt, hogy a fórumot böngészgetve rátaláltam a „web emu” témára. Itt egy felvetés böngészőben futó emulátorról szól, igaz az eredeti kérdés szerint plug-in vagy extension formájában. Az jutott eszembe, hogy utána kéne nézni a JavaScript lehetőségeinek, hiszen ez a technika „komolyabb” célokra való felhasználása felfutóban volt, és pont akkoriban botlottam bele az JSspeccy nevű JavaScript alapú ZX Spectrum emulátorba is. A dolog már csak azért is érdekelt, mert szerettem volna a JavaScript-et kicsit jobban megismerni mint amennyire a szokásos pár soros „weboldal dekoráló” megoldásokból az ember ismerheti. Megnéztem az JSspeccy forrását, és rá kellett, hogy jöjjenek: nem feltétlen tudnék ilyet írni magamtól elsőre (azóta már menne ...). Ezért azt a megoldást választottam, hogy lépésenként megpróbáltam azt átírni azzal a céllal, hogy a végén csak a Z80 emuláció maradjon, a többi része pedig már teljesen az én munkám lesz és persze EP-t emulál Spectrum helyett. Az első valamennyire működő megoldás meglepően hamar működőképes lett, bár itt még a kurzor is zöld volt – tehát némi Nick emulációs hiányosságok –, és egyéb „apró” problémák :) Az emulátor az JSep nevet kapta (hé, végre valami, aminek a nevébe nem teszek X-et!).

Az JSep számomra tehát egy tanulási folyamat is volt. Sajnos ez meg is látszik rajta: a mai JavaScript tudással már inkább sírógöröcsöt kapok azon, amilyen megoldásokat ott használtam. Igazából ez a fő gát is továbbfejlesztésének: egyszerűen nincs kedvem újra hozzányúlni ilyen kezdetleges kódhoz. Ezért teljes újírásra lenne szük-

ség (amit ugyan most már megtehetnék, csak éppen nem lenne sok értelme, lásd cikkem végefelé). Az emulátor jelenleg rengeteg dolgot nem tud, amit jó lenne, csak pár dolgot említve: nincs hang, pontatlan a Z80/Nick/VRAM összehangolás, nem lehet az emulátorral menteni, pontatlan Nick/vidéo emuláció (interlace, stb.), ésszerűen (az új helyesírás szerint most akkor így kell írni?) az emulátor nem használható saját oldalon / megoldásba ágyazva. Viszont van legalább két dolog, amit az JSep tud, viszont az ep128emu nem: emulálja a boxsoft-féle egerillesztőt (terveim szerint egyszer majd az EnterMice-t is), illetve az Am9511 APU-t (mai szóhasználatnál FPU lehetne – bár az APU emuláció sose lett komolyan tesztelve, software hiányában). Bizonyos limitációk a webes technikából fakadnak: többen kértek „beállító menüket”, illetve „floppy disk image

```

IS-BASIC          program 0
ok
:help xep
Helper ROM: XEP version 0.3 (Xep128
EMU)
Xep128 0.3 (C)2015,2016 LGB Gabor
Lenart
Built on: travis@travis-lgb on Darwin
13.4.0
Tue Jul 26 06:59:44 UTC 2016
GIT:
7e1a6255bea6bfbd6230bfd860a430b0e5b
fa
Compiler: clang 4.2.1 Compatible Apple
LLVM 6.0 (clang-600.0.54)

Commands: AUDIO CD CLOSE CPU DDN DIR
DISASMC DJ EMU EXIT EXOS HELPC?J LPT
MEMDUMPCIMJ MOUSE PAUSE PORTS PRIMO RAM
REGSCRJ ROMNAME SDL SETDATE SHOWKEYS
TESTARGS

For help on a command: (:XEP) HELP CMD
ok

```

XEP 128 Enterprise emulátor Windowsra, Linuxra és Mac OSX-re és web felületen

csatolást”. Az a baj, hogy ők azt nem veszik figyelembe, miszerint ez egy webes emulátor, célja a web környezetbe való integrálás, helyi gépről nem fog tudni csatolni semmit (utólagos kiegészítés: erre ma már lenne lehetőség, ám továbbra sem kényelmes és célszerű pár szempontból), mindent weben át kell neki adni, vagy URL (ahonnan letölti) vagy URL paraméter (config beállítások stb.) formájában. Az JSep talán hasznos lehet egy-egy software gyors bemutatására, főleg olyanoknak, akik nem feltétlen szeretnének egy emulátort telepíteni saját gépükre. Akkor még elfogadható az arra beállított környezet webes link formájában. Dióhéjban: egy webes emulátor célja a webes környezetben való futás, ami nem túl meglepő :) Ez sok esetben kissé másfajta megoldásokat szül, mint amit a natív alkalmazások esetén (pl. ep128emu) megszokunk. Például el tudok képzelni egy weboldalon némi Enterprise software gyűjteményt, ahol egy klikkel ki is lehet próbálni az adott programot azonnal weben, mielőtt az ember letöltené, stb.

Meglepő fordulat azonban, hogy az JSep megoldása más esetben is hasznos lehet (főképpen hasznos lesz, de itt is egy alapos újrairás után lehetne számítani elfogadható eredményre): a különböző mobil eszközökre, okos telefonokra, és tabletekre gondolok most. Annak idején kipróbáltam szerzeményemet a saját okos telefonomon, az eredmény lehangoló volt, talán slideshow-nak lett volna csak alkalmas az eredmény. Viszont a mostani okos telefonommal és egy mostani böngészővel rajta egész meglepően jó eredményt képes produkálni! Ez egyrészt köszönhető persze a kérdéses eszközök hardware szintű fejlődésének, azonban a JavaScript teljesítménye is meredeken növekszik az elmúlt években, és szinte versenyt űznek a böngésző gyártók ebből. Lassan oda jutunk, hogy egy kissé heterogén mobil eszköz környezetben (Android? iOS? Windows Phone?) lehetőség nyílna egy egységes megoldás használatára (főleg a jelenlegi mobil specifikus web API-k és megoldásokkal, amivel JS-ben akár az eszköz elmozdulása és ilyesmi is kezelhetőek). Hozzátenném, az JSep „kézzel” írt JavaScript kód, nem az asm.js megoldás, amihez akar C fordító is van (pl.: emscripten), mely képes C kódot az asm.js nevű JavaScript részhalmozra „fordítani”. Ezekre a megoldásokra azt mondják, hogy már csak kétszer-másfélszer lassabb adott esetben mint a natív kód (ezt igazolni nem tudom, még nem igazán próbáltam). Természetesen a webes technikának van még hova fejlődniük, az JSep-nek meg különösen ;)

Az JSep másik „sikere” az elrettentés volt :) Történt ugyanis, hogy rátaláltam a SymbOS nevű remek kis multi-tasking Z80-ra írt operációs rendszerre, ami azonban akkoriban még nem futott Enterprise-on. Egy örült ötlettől vezérelve betöltöttem a CPC-s verziót egy CPC emulátorba, amit forráskód szinten átirítottam: saját formátumba mentve adott pillanatban a memória tartalmát és a Z80 állapotát. Aztán kissé módosítottam a saját EP emulátoromat, hogy ezt be tudja tölteni, majd pár órát eltöltöttem vele, hogy a CPC memória/CPU állapotnak „pillanatképe” hogyan futtatható tovább egy tők másik gépen, az Enterprise-on :) A módosított JSep emiatt kis részben CPC-t is emulált az EP mellett! A tervem az lett volna, hogy apránként patch-elgetem a mentett CPC memóriatartalmat addig, amíg a CPC hack-ek nélküli emulátorban is elfut, ami után csak némi betöltő kell hozzá: tehát a SymbOS portolásáról lett volna szó memóriakép alapján az Enterprise 128-ra :) Tudom, az ötlet elég furcsa. Annyira örült, hogy Prodatron (a SymbOS szerzője) válaszolt nekem, és kijelentette, hogy ez annyira hülyeség, hogy már jó :) Illetve „elrettenve” ettől, inkább portolta ő maga Enterprise-ra rendesen a SymbOS-t :) Az igazsághoz tartozik, hogy már régebben is volt az Enterprise-ra portolási terv, de – amennyire én tudom – ennél több addig nem igazán történt. Tehát ha közvetlenül nem is, közvetve hasznos volt az JSep a SymbOS Enterprise-ra való megjelenése kapcsán is :) Persze lehet, hogy Prodatron ezt másképp mesélné el :) Az olvasótól pe-

dig elnézést, de hadd örüljek néha egy kis sikernek én is, a sikerélmény mint motiváló tényező fontos kérdés abban, hogy az ember ne hagyja abba a dolgot. :)

Egy nosztalgikus napon (már 2015-ben vagyunk) eszembe jutott az első Enterprise emulátor projektem, az amit C-ben írtam, és csak Linux alatt futott (tehát a Xepem). Mivel az SDL2-vel (az SDL egy cross-platform „média réteg” ami elérhető pl. Linux, Windows, Android stb. rendszerekre is) szerettem volna (jobb) megismerkedni, elhatároztam, hogy megpróbálom a régi projektemet (már a forrásra se volt meg) „utánozni” SDL2 alapokon, ezúttal nem saját, hanem a z80ex projekt Z80 emulációját felhasználva (amit már régebben is felhasználtam pár kisebb saját projektnél, pl. a CP/M emulációs réteg a UNIX környezetben, és hasonló örültségeknél). Ezúttal a dolog jobban ment mint anno, ez talán annak is köszönhető, hogy a Z80 emuláció biztosan teljes, és jó volt :) Az új emulátor a Xep128 nevet kapta. Neve a UNIX-os eredetre vezethető vissza (GUI-s programok nevei – azaz X11 alatt futóak – régebben hagyományosan X-szel kezdődtek). Hasonlóan az JSep-hez, itt is a tanulás volt a fő szempont: SDL2 ezúttal. Őszintén, addig sem terveztem eljutni, mint az JSep-vel.

A Xep128 szintén csak Linux alatt futott, ám bennem volt a kisördög, hogy ha már SDL (ami cross-platform) mennyire lenne bizonyult Windows-os programot is gyártani belőle (persze ez most egyszerűsítés, az SDL2 mellett volt benne közvetlen Linux specifikus dolog is, amit előbb orvosolnom kellett). Ismételném: sose volt Windows-om, Windows alá meg pláne nem programoztam még semmit! A megoldás tehát az lett, hogy kipróbáltam a Mingw nevű cross-compiler-t, ami Linux alatt futva képes C kódból Windows-os exe-t gyártani. El lehet képzelni meglepetésemet, amikor fel óra melőval előállt egy exe, ami a wine alatt legalábbis elfutott. Ha már idáig eljutottam (ami az elején szintén csak kíváncsiság volt részemről: menne-e Windows alá is) elhatároztam, hogy a „duál platform” jelleget megtartom (azóta ez már nem csak „duál”), a Xep128-nak mennie kell Linux (vagy más hasonló UNIX) és Windows operációs rendszerek alatt egyaránt.

A fejlesztési szokásaimon (hobby szinten is, mert munkában amúgy is így volt mindig is) szintén próbáltam csiszolni: ezúttal már verziókövető rendszerbe tettem a dolgokat (git), és a Github-bal így egyben elérhetővé és tettem mások számára. Illetve, célul tűztem ki, hogy a Xep128 se „általános” emulátor legyen, hanem valamire szakosodjon: ahogy az JSep a web-re, úgy a Xep128 az „extrémebb” hardware elemek emulálására. Hogy miért? Mivel a Xep128 is legalább annyi (sőt több!) hiányossággal rendelkezik (frissítés: rendelkezett ...) mint az JSep, ezért csak ki kellett találnom valamit, hogy akkor miben jó legalább :) Igen, a Xep128 több tekintetben annyit sem tud (tudott) mint az JSep, csak VINT/1Hz interrupt van (volt) jelenleg például. Viszont a boxsoft féle egér

szintén megy (azóta EnterMice dolgok is). Ahogy ezt is, sok dolgot egyszerű volt megcsinálni, mert az JSep-be írt JavaScript kódot kellett „átfogalmazni” csupán C-be. Viszont új például az SD kártya cartridge támogatás, ami sem JSep-ben, sem pedig ep128emu-ban nincs (azóta ep128emu-ban is van). Jelenleg csak olvasni lehet vele (azóta már írni is), de ez már mutatja, hogy az JSep-hez képest megvan a natív implementáció előnye, ezt weben megoldani főleg egy nagy méretű disk image esetén elég nehézkes. A SymbOS Xep128-al működik, egérrel, SD kártyával is (JSep-vel is működik, persze ott nem SD-ről).

Hogy mi még „extrém hardware”, arra jó példa a Z180 emuláció a Xep128-ban :) Nem sok embernek van Z180-al szerelt Enterprise-sza, tudtommal egyetlen egynek :) Mégis, érdekes problémának tűnt ennek megoldása (még nincs kész, de arra már jó, hogy Z180 által nem ismert Z80 opcode-okat kiszúrja például). A ZX Spectrum hardware emulátor kártya emulálása :) szintén elég örült ötletnek tűnt, bár lehet, ez nem túl eredeti ötlet (ep128emu-ban is van, tudtommal). Azonban kétségkívül a legértékesebb az lenne, ha a félbehagyott munkát az EPNET emulációját folytatnám. Ha valaki még nem hallott volna róla, az EPNET Bruce Tanner barátunk (aki az eredeti IS-BASIC-et, és az EXOS egy részét is írta például) projektje, dióhéjban: „kössük az Enterprise-t az Ethernet LAN-unkra és az Internetre”. Ezt emulálni egy webes emulátor nem is tudná, hiszen nem képes http-n és hasonlókon kívül más, natív protokollal használni a hálózatot, jellege miatt. A Xep128 mivel natív alkalmazás, képes lehet erre. Szerény véleményem az, hogy az EPNET mindenképpen ütős kis hardware lesz majd, hiszen „Enterprise a neten” az valami új. Ennek emulálása szerintem ezért eléggé fontos feladat lehet. A SymbOS új verziójában már van hálózat támogatás (igaz még nem az EP-s verzióban), és Prodatron valószínűleg örömmel tenne EPNET támogatást is az Enterprise verzióba majd. A Xep128 EPNET emulációjával meg mindez valódi hardware nélkül is kipróbálható lenne (lenne, ha haladnék vele).

Az Xep128 egyik nagy hátránya jelenleg, hogy köszönőviszonyban sincs a felhasználóbarátság (legalábbis, ha a felhasználó nem command line fetisizista, mint én) fogalmával :) Az JSep esetén ez még elfogadható annak jellege miatt („web”), hogy nem lehet szabadon configolni, helyi file-okat betölteni ROM-ként, disk image-ként stb. Azonban a Xep128 egyik lényeges jellemvonása pont az, hogy natív applikáció. A problémát azonban nehezíti, hogy lényegében az SDL csak egy (SDL2 - vagy több) „ablakot” ad, amibe szabadon lehet „rajzolni”. Ha az ember beállító paneleket, stb. akar, azt neki kell megrajzolni kvázi képpontként (még egy szöveg kiírása is így megy). A másik lehetőség, hogy az SDL mellett az emulátort futtató operációs rendszer saját GUI elemeket kell felhasználni. Ez azonban azért problémás, mert így akkor megszűnne azonnal a csaknem teljesen közös Linux/Windows

kódbázis (ami javarészt csak az SDL-en át „érintkezik” az OS-sel, így nem lát különbséget, azt meg az SDL elintézi), és külön Linux és Windows specifikus implementációk szükségesek ennek kialakításához (azóta pl. a FILE: implementáció használ file kiválasztó ablakot Windows és Linux viszonylatában is). Más függőséget nem igazán szeretnék bevezetni, bár léteznek SDL-re épülő platform független GUI megvalósítások, ezek kinézetük sem túl esztétikus, meg hát méretük sem kifejezetten apró ...

Amit én érdekesnek találok, az a „nagy egyesítési elmélet”: valami „app store” szerű cucc Enterprise-ra, neten át, ahogy kell. Valódi hardware-en ott az EPNET ehhez, működik. JSep eseten HTTP/AJAX neten át szintén. Végül pedig Xep128 esetén az EPNET emulációval kb. az első esetben vagyunk.

Nagyon érdekes amúgy látni az emulátorok közötti forráskód szintű kapcsolatot. Az JSep Z80 emulációja még mindig az JSpeccy féle, ami azonban a FUSE-ből származik (a FUSE egy eredetileg UNIX-ra írt ZX Spectrum emulátor – Free Unix Spectrum Emulator). A Xep128 Z80 emulációja a z80ex nevű Z80 emulátor projektből jön (habár erősen módosítgatva lett általam), ami meglepő módon szintén a FUSE-ből nőtte ki magát. Erre amúgy csak később jöttem rá, mindenesetre érdekes. Az összes említett projekt software licence (GNU/GPL) amúgy ezeket a forráskód átvételeket lehetővé is teszi (betartva néhány szabályt, például, hogy biztosítani kell az ember saját munkájának a forrását is aztán, amiben felhasználja). A Xep128 kapcsán elégedett vagyok a z80ex-szel, viszont az JSep oldalán az újrainást tervezem, hogy az eredeti FUSE kódbázisból kiindulva hozok létre Z80 emulációt, ahogy az JSpeccy is tette.

Érdekes tény, hogy az „igazán gyors” JavaScript – úgy tűnik – az asm.js és hasonló technikák irányába fog elmenni, ami nem kedvez a „kézzel írt” JavaScript kódoknak, mint amilyen jelenleg az JSep is. A már említett emscripten (C-ből asm.js) fordító ráadásul pont SDL API-t „emulál”. Így elvileg a Xep128 kisebb-nagyobb módosításokkal alkalmas lehetne a Windows és Linux natív működés mellett emscripten-t használva webes emulátor „generálására” is (azóta alkalmas is). Az is igaz, hogy SDL viszont van Androidra is, tehát elvileg az is lehetséges, hogy a Xep128-at Android natív appnak fordítsa le valaki, némi munka árán. Ki tudja, lehet, hosszú távon jobban megéri majd egyetlen emulátort fejleszteni, és nem web / natív bontásban kettőt? Ehhez azonban az én tudásomnak is gyarapodnia kell még. Pont ez az egyik tanulsága is cikkemnek (legalábbis szerintem): ha az ember már úgy is meg akar tanulni valamit, miért ne tegye ezt egy olyan projekten át, ami egyben a hobbiával is kapcsolatos, de ami tudást aztán máshol is fel tud majd használni? A másik érdekes következtetés: most is, de főleg régebben sokan kérdezték miért nem használok Windows-t, mint bárki más, az emulátorral se lett volna gondom. Az lehet (és nyilván az ember személyes OS

választása nem csak ezen múlik), viszont akkor nem is akartam volna írni egyet, ezáltal meg sem ismertem volna a gépet annyira, a Z80 gépi kódot, stb. Lehet, csak egy gyors kaland lett volna „na, erre a gépre is ránézem” alapon (amire nyilván elég sok példát említhetnék, ahogy bárki más is – gondolom) az EP. Így viszont menet közben szerettem meg, mondhatni. Más hasonló témában is jártam már így amúgy, és nem csak a „régis számítógépek” kapcsán, érdemes néha nagyobb dolgokba is belevágni.

Végezetül, szeretnék köszönetet mondani mindenkinek a fórumon, két személyt külön kiemelve: Zozo, aki sok tanáccsal és információval látott el az EP kapcsán (illetve néha igényekkel is az emulátor kapcsán), és István (az ep128emu szerzője) aki több esetben is irányt mutatott az emulátor írás rejtelmeinek útvesztőjében. Meg persze mindenkinek, aki kipróbálta, véleményezte munkámat (itt megemlítve Gflorez fórum társunkat, aki lelkesen kipróbált mindent, „egerészésben” adott tanácsokat, sőt ikont is szerkesztett aztán az emulátorhoz). Külön köszönet az olvasónak, aki kisregényemet – itt és most – képes volt végigolvasni :)

Ám ezzel nincs vége ...

Eredeti cikknek szánt betűhalmazom (mely fentebb ért volna véget) bizony nem épp tegnapi, azóta sok víz folyt le a vén Dunán. Legérdekesebb hír azóta talán az, hogy István újra foglalkozik aktívan az ep128emu emulátorral, ami örvendetes hír minden EP-snek, és íranta érdeklődőnek! Hiszen, én, mint az Xep128 szerzője is be kell, hogy ismerjem: azért az ep128emu-t a Xep128 messze nem képes pótolni. Azóta az ep128emu-ba is bekerült egy-két „hiányzó” dolog (a Xep128-hoz képest), például az egér, és az SD kártya emuláció. Ez utóbbi esetben megint azt látom, hogy bár a Xep128 közvetlenül nem feltétlenül hasznos túl sok ember számára, ismét nem volt értelem nélküli a dolog, hiszen az ep128emu jelenleg a Xep128-tól „örökölte” az SD emulációs képességeit. Ilyesféléképpen talán hasznosabb is a munkám, mintha csak a talán kevesek által ismert vagy használt Xep128-ban lenne benne ez a – szerintem hasznos – képesség.

A Xep128 még a fenti történet előtt (és az eredeti cikk megírása után) is sokat fejlődött azért, csak nagy vonalakban: file kezelő szintű FILE: eszköz (ami azóta ep128emu-ba is megérkezett), natív Linux (GTK-3) és Windows file selector ablak (szóval némi GUI, ha több nem is), minden Dave interrupt implementált, SD kártya írás is megy már, „amatőr hang” már van, stb. Elmondhatjuk, hogy mára a Xep128 tud legalább annyit (vagy inkább többet), mint az JSep tudott, igaz ez még mindig nem egy ep128emu ettől még (messze nem).

Az JSep jelenleg „halott” projekt, olyan értelemben, hogy nem hiszem, hogy tovább fogom fejleszteni. Elvégre, egy emulátort is nehéz csiszolgatni, az ember ne akarjon két emulátort írni és folyamatosan fejlesz-

teni, főképp, ha nem „csereszabatos” – mivel egyik C a másik meg JavaScript nyelven íródott -, illetve ha ugyanannak a gépnek az emulációjáról van szó ráadásul. Ugyanakkor, szintén új keletűbb hír, hogy a Xep128 már lefordítható HTML5/JavaScript (a „web”) „platformra” is, amiről már fentebb írtam, viszont jelenleg már működő demó verzió is van ebben a tekintetben (ami így jó is „az JSep helyett”). Ez többek között azért jó, mert így elég egyetlen emulátort fejleszteni, és az számtalan platformra elérhető, legyen szó, Linux/UNIX (Raspberry Pi is ideértve), OSX, Windows, vagy akár már a „web” is. Talán legjobban pár mobil platform hiányzik még, nekem konkrétan az Android (táblaten biztos jól mutatna ... habár a webes verzió is esetleg produkálhat értékelhető eredményt egy ilyen eszközön is ma már). Engedtessek meg, hogy röviden megemlítek egy közéletű tévedést, ha már JavaScript: semmi köze a Java-hoz! Azon kívül, hogy valószínűleg marketing okokból hasonló elnevezést adtak neki. A JavaScript – főleg manapság – a „web programozási nyelve” – és natív eleme a böngészőknek, míg a Java-hoz külön plug-in és az JRE kell, és mint programnyelv is ég és föld a kettő, már eleve.

Szintén színesíti elfoglaltságaim palettáját az Xemu projekt. Itt azt tűztem ki célul, hogy legyen egy „emulációs keretrendszer” amin belül többféle gépet is tudok emulálni, így nem kell mindenhez külön emulátor projekt, nulláról kezdve. Ennek keretein belül már elérhető a Commodore LCD (elégge ismeretlen, forgalomba nem került gép, amihez én írtam először működő emulátort még anno, igaz az szintén JavaScript-ben volt még), Commodore VIC 20, folytatása következik majd persze, mind emulált gépek számában, mind pedig feature-ök tekintetében, TVC és pl. Jupiter Ace készülében – tehát a jövő évezredben talán, magamat ismerve). Mivel a Xep128 sok tekintetben hasonló alapokra épül, a távlati tervem az, hogy ez is az Xemu projekt keretein belül él majd tovább előbb-utóbb. Ez negatív változást az esetleges Xep128 felhasználók (ha van ilyen egyáltalán ...) számára nem hoz, viszont megint csak a karbantarthatóság kérdéséről van szó. Természetesen az Xemu is tovább akarja vinni a sok platformos jelleget (a „webet” mint platformot is ideértve), ám ezúttal ehhez csatlakozik a „sok célúság” szempontja is, ahol „cél” alatt most azt értem, hogy emulálandó számítógép, mint cél. Persze, most itt nekünk ebből az egészből az EP a lényeg.

<http://xep128.lgb.hu/>
<http://jsep.lgb.hu/http://xep128.lgb.hu/web-demo/>
<https://github.com/lgb/lgb/xemu/wiki>
<http://c65.lgb.hu/web-xemu/xc65.html>

Az EPIMGCONV használata

A legegyszerűbb példa:

epimgconv foo.jpg foo.com

A video módot a -mode paraméterrel lehet választani, és a következő értékek használhatók:

* 0: 2 színű, normál függőleges felbontás (nem feltétlen fekete-fehér, hanem soronként optimális palettát próbál választani)

* 1: 4 színű, normál függőleges felbontás

* 2: 16 színű; 256 színre ditherelt képet konvertál 16 színre dither nélkül (ez az alapértelmezett), normál függőleges felbontás

* 3: 16 színű; a 2-es móddal azonos palettát generál, de az eredeti képet dithereli 16 színre, normál függőleges felbontás

* 4: 16 színű; ez a paletta keresést és a dithert próbálja kombinálni, de nagyon lassú, normál függőleges felbontás

* 5: 256 színű, normál függőleges felbontás

* 6: attribute mód, normál függőleges felbontás

* 10: 2 színű, interlace (nem feltétlen fekete-fehér)

* 11: 4 színű, interlace

* 12: 16 színű; 256 színre ditherelt képet konvertál 16 színre dither nélkül, interlace

* 13: 16 színű; a 2-es móddal azonos palettát generál, de az eredeti képet dithereli 16 színre, interlace

* 14: 16 színű; ez a paletta keresést és a dithert próbálja kombinálni, de nagyon lassú, interlace

* 15: 256 színű, interlace

* 16: attribute mód, interlace

A **-mode 0-6** tehát a normál függőleges felbontást, a **-mode 10-16** pedig az interlace-t alkalmazza.

A **--help** a program használatát és a paraméterek listáját írja ki rövid leírással.

A **-raw 1** használatával program helyett a fent leírt (nyers, külön program nélkül nem használható) formátum-



ban lehet létrehozni a konvertált képet (amit így az IVIEW programmal jeleníthetünk meg Enterpsie-on); így a méret is nagyobb lehet (ha a kimeneti formátum program, akkor csak max. 46 karakter * 288 sor lehet). Az **-outfmt n** használatával is nyers formában menthetjük a képet, ennek paraméterezése a következő:

-outfmt 0, -outfmt 1: ugyanaz, mint a **-raw 0** és **-raw 1**.

-outfmt n (11<=n<=19): a nyers formátum tömörítésének mértékét adhatjuk meg. A tömörített és a normál nyers formátum között az epcompress programmal lehet konvertálni, illetve az epcompress csomagban található uncompress bővítéssel Enterprise-on is kitömöríthetők az ilyen formátumú képek.

-outfmt 2: Agsys .crf formátumban menti a konvertált képet. Ezt csak 1-es módban (4 szín/sor, nem interlace) lehet használni, és az Agsys nem tud nagy méretű file-okat betölteni (a **-size 36 216** még működik, de például a **-size 40 240** már nem). Bár az Agsys támogatja a soronként változó palettát, a képet egyszerűbb lehet szerkeszteni a **-palres 0** vagy **-color0..color3** paraméterekkel fix palettát beállítva.

-outfmt 3: a ZozoTools VL és VS parancsok által használt formátumban írja a konvertált képet. Interlace és attribútum módokat nem támogat, és a file méret legfeljebb 16K lehet. Ez a formátum automatikusan **-palres 0-t** és EXOS-kompatibilis képméretet (legfeljebb 42x27 karakter, 9-el osztható magasság) állít be, akár csak az **-outfmt 4**.

-outfmt 4: PaintBox formátum; ez szintén nem támogatja az interlace módokat, és a file nem tárolja a FIX-BIAS-t, így azt vagy a konverter által kiírt értéknek megfelelően be kell állítani, vagy a konverziónál a **-bias 23** paramétert kell használni. A formátum automatikusan **-palres 0-t** és EXOS-kompatibilis képméretet (legfeljebb 42x27 karakter, 9-el osztható magasság) állít be, mint az **-outfmt 3** is.

-outfmt 5: Zaxial formátum, a Paint-Box-hoz hasonló korlátozásokkal, de tárolja a bias értékét. Attribute mód esetén a file csak az eredeti VLOAD paranccsal tölthető be, a Zaxial nem támogatja ezt a módot.

-outfmt 6: Fejléc nélküli formátumban lehet menteni csak az attribútum és pixel adatot. Ez automatiku-

san „-palres 0”-t állít be. Attribútum módban először az attribútumokat menti, aztán a pixel adatot. Elvileg interlace módban is lehetne használni, de ez egyelőre hibás, mert a második félkép pixel adata elcsúszik 8 byte-tal. Célja: egy hagyományos grafikával (pl. a 4 színű Wiggler) rendelkező programba lehessen PC-n megrajzolt grafikai elemekből közvetlenül felhasználható pixel adatbájt kupacot konvertálni. Pl: egy 16x16-os 4 színű BMP konvertálása: epimgconv_sse2 sprite.bmp sprite.raw -size 2 0 -dither 0 0 -nointerp 1 -mode 1 -outfmt 6

A méretet a **-size W H** állítja, **W** = szélesség karakterekben, **H** = magasság (interlace esetén félkép-) sorokban. Ha a **-size** után megadott egyik érték 0 vagy negatív, akkor azt kiszámolja a másiktól és a kép méretéből. Például a **-size 46 0** azt jelenti, hogy a szélesség mindig 46 karakter, a magasságot pedig automatikusan választja a kép mérete alapján. Így azonban a magasság túl nagy is lehetne, ezért negatív értéket megadva be lehet állítani egy maximális méretet. A **-size 46 -300** hatása tehát azonos a **-size 46 0**-al, de ha a magasság így nagyobb lenne, mint 300 sor, akkor az 300 sor lesz, és a szélességet csökkenti.

Alapértelmezés szerint a program a képet a legnagyobb méretre konvertálja, ami még éppen kifér (ha oldalt vagy felül és alul marad hely, akkor azt keretszínnel tölti fel), de a **-scale X Y** és **-offset X Y** segítségével lehetőség van a kép nagyítására és eltolására.

A **-scalemode 1-et** beállítva a képet nem úgy méretezi át, hogy a legnagyobb méretű legyen levágott részek nélkül, hanem a legkisebb üres (keretszínű) területek nélkül. Tehát ha a **-size** paraméterrel beállított szélesség és magasság aránya nem megfelelő, akkor nem lesznek keretszínnel feltöltött területek, hanem a kép egy részét levágja oldalt vagy alul és felül.

A **-color0 n ... -color7 m** (n, m: színkód száma) paraméterekkel fix palettaszíneket lehet megadni (egyelőre nem tudja, hogy ne használjon egyes színeket). Mivel a **-mode 0** vagy **-mode 10** nem direkt fekete-fehér képet generál, a fekete-fehér képet a **-color0 0 -color1 255** paraméterek megadásával lehet elérni. A színeket

#RGB formátumban is meg lehet adni, pl. a #000 a fekete és a #773 a fehér szín.

A **-bias** paraméterrel a fixbias állítható - így esetleg jobb minőséget is el lehet érni, mint az automatikusan választott értékekkel. A színeket #RGB formátumban is meg lehet adni, pl. a #331 a bias=31.

Ha a .com file túl nagy (>47.75K) lenne, akkor a program automatikusan tömöríti; így - szerencsés esetben - az EXOS már be tudja tölteni, igaz, sokkal lassabban indul el. A **-nocompress 1** használatával a „nagy” programok automatikus tömörítése letiltható.

A **-nointerp 1** kikapcsolja az interpolációt a kép átméretezése közben (pl. 32x32 -> 512x512 2 színű interlace formátumra konvertálva interpoláció nélkül; így minden pixel pontosan 16x16 méretű négyzet lett)

A **-palres 0** letiltja a soronként változó palettát (az alapértelmezés a -palres 1), így a paletta az egész képen ugyanaz lesz. Ez általában rontja a minőséget, ezért nem tűnhet túl hasznosnak, de így valamivel kisebb lehet a konvertált file, és esetleg más programokban is egyszerűen lehet használni.

.jpg, .png, .gif, .bmp, és .xpm formátumot eszik meg a program.

IstvanV

Tippek

A következő módszereket alkalmaztam és kombináltam:

- az attribute mód adja a legjobb eredményt, utána a 4 szín mód;
- olyan képeket válogattam ki, amelyek nem túl bonyolult struktúrát tartalmaznak;
- kivágtam a képből a legjobb részletét, de úgy, hogy azért a kompozíciója szép maradjon;
- olyan képeket kerestem, ahol kihasználódik, hogy vízszintesen elválasztott részekben elég sokféle szín legyen;
- sokféle és erős színeket tartalmazó képeket használtam.

Szóval ahol ezek a legjobban érvényesültek, azok lettek a legjobb konverziók.

Endi

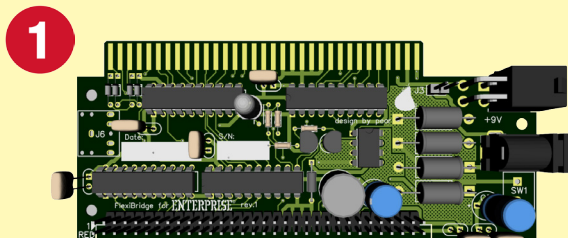
Ez 16 szín attributum mód, soronkénti palettával, és interlace-szel, fényképszerű képeknél ez vált be leginkább:

epimgconv_sse2 jpgname eppi-cname -size 42 280 -quality 9 -mode 16

Zozosoft



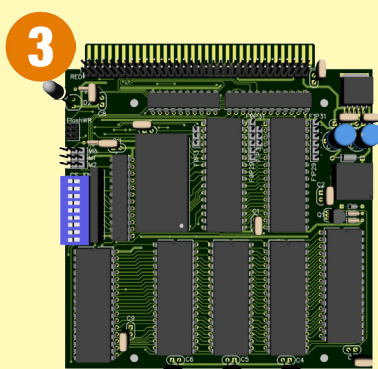
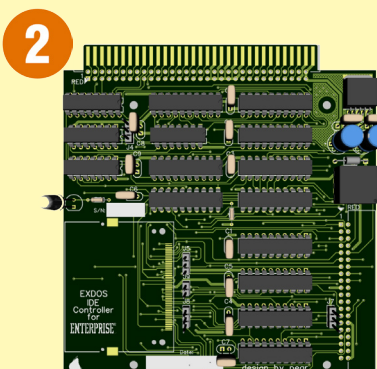
Pear (Maciej Gruszecki) készülő hardver újdonságai



1. A FlexiBridge új, átdolgozott változata.
2. IDE-Compact Flash kártya (winchester vezérlő, CF memóriakártya olvasó).
3. RAM-Flash-Clock kártya

A RAM-Flash-Clock kártya rövid leírása:

- tartalmazza majd Mészáros aGyula óra-kártyáját - Enterprise Clock & Calendar (DS12887),
- két foglalat FlashROM-nak és öt foglalat SRAM-nak előkészítve (mindegyik 512 KB),
- mindkét FlashROM cserélhető SRAM-ra (jumperekkel konfigurálható)
- 8 féleképpen programozható memória kiosztás (3 jumperrel, M0, M1, M2)
- minden chip-et ki lehet önállóan kapcsolni, beleértve az RTC-t is (dip kapcsolókkal).



ENTERPRISE KLUB

Minden hónap utolsó szombatján!

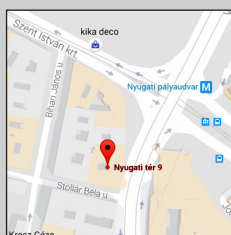
Helyszín:

Skála terem

Budapest (V. ker.)

Nyugati tér 9.

14 órától 19 óráig



Információ: www.enterpriseklub.hu

Ha te is szeretnél Az ENTERPRESS Magazin szerkesztője lenni, küldj cikket, játékleírást, játékismertetőt, vagy bármit amely az Enterprise számítógéppel kapcsolatos!

A cikkeket erre az e-mail címre küldheted:

info@enterpress.news.hu

ENTERPRISE FOREVER

<https://enterpriseforever.com>

ENTERPRESS Magazin - 2016. november

Főszerkesztő: Matusa István

Szerkesztőségi főmunkatárs: Németh Zoltán (Zozosoft)

A csapat: geco, Povi, Kiss László, SzörG, szipucsu, lgb

Design, nyomdai előkészítés: Matusa István

Weboldal: <http://enterpress.news.hu>

E-mail: info@enterpress.news.hu

A lap időszakosan - korlátozott példányszámban - nyomtatott formátumban és elektronikus formában is megjelenik.

ENTERPRESS e-magazinok:

<http://enterpress.news.hu/index.php/magazin>