

SPECCYALISTA VILÁG

A SPECCYALTA BARÁTI KÖR LAPJA



TV-BASIC különkiadás 3. rész

ZX Evolution 2. rész

Játékújdonságok

Játszd újra Fred!

Klavia-túra

SCenE esszencia

Assembly ovi 6. rész

Matthew Smith story

FPGA Jocó titkos naplója 1. rész



2016. 2. SZÁM

TARTALOM

BEKÖSZÖNTŐ.....	3
SPECTRUMOLÓGIA	
SPECTRUM FEJLESZTŐI KÖRNYEZETEK - 2. RÉSZ.....	4
OLVASOKKOLÓ	
EGYESÜLETI ÉLET, AHOGY AZT PGYURI ELKÉPZELTE.....	6
ARCKÉPCSARNOK	
MATTHEW SMITH.....	7
KLÓNOK HÁBORÚJA	
AMIT SOSEM MERTÉL MEGKÉRDEZNI A ZX EVOLUTIONRÓL 2. RÉSZ.....	10
LOAD " "	
JÁTÉKÚJDONSÁGOK.....	9
JÁTSZD ÚJRA FRED!.....	14
BOLDOG KARÁCSONYT! - HORACE.....	15
ZX81 KLASSZIKUSOK - 3. RÉSZ.....	16
PROGRAMOZÁSTECHNIKA - BASIC	
ZX SPECTRUM TV-BASIC KÜLÖNKIADÁS - 3. RÉSZ.....	17
PROGRAMOZÁSTECHNIKA - ASSEMBLY OVI	
HOGYAN ÍRJUNK JÁTÉKOT ZX SPECTRUMRA - 6. RÉSZ.....	22
SCENE ESSZENCIA	
ACROSS THE EDGE.....	27
HARDVER SIMOGATÓ	
KLAVIA-TÚRA: BILLENTYÜZET SAGA.....	28
HARDVER ÖTLETEK - ZX SPECTRUM	
RÉSZLETEK FPGA JOCÓ TITKOS NAPLÓJÁBÓL - 1. RÉSZ.....	32
SZUPER BOK.....	37
REJTVÉNY.....	38

BEKÖSZÖNTŐ

Kedves olvasó!

Végre itt a várva várt negyedik szám! Bár szerettünk volna most is korábban megjelenni, de sokat akart a szarka... Emiatt is döntöttünk amellett, hogy idén inkább negyedévente, rövidebb 16-24 oldalas számokkal jelentkezünk majd.

Most is mint mindig, igyekeztünk az egykori Spectrum Világ szellemiségét megőrizni és minden olvasónknak adni valami számára érdekes olvasmányt. Valójában ezért is ez a kis csúszás, megpróbáltuk összevágni, hogy minden témából szolgálhassunk valami érdekességgel.

A szerkesztőségünk címére ugyan most is kevés visszajelzés érkezett, de ezt a vonalat igyekszünk továbbra is képviselni.

A hardverészeknek továbbra is szeretnénk kedvezni, ezért a berkeinken belül folyó munkákba ismét engedünk némi betekintést, amihez most is Jocó nyújtott hathatós segítséget.

A Spectrumos témák mellett már ez előző számokban is elkezdtünk szemezgetni a ZX81-gyel foglalkozó írásokból, nem hagyunk fel ezzel most sem. Az elkövetkező számokban viszont talán sikerül bővítenünk a sort a Spectrum Világból megszokott Enterprise vonallal.

Reméljük a "lap" továbbra is elnyeri tetszésedet és lesznek, akik szívesen bekapcsolódnak a további munkába akár csak egy-egy cikk erejéig. Ugyanakkor mindenképp várjuk véleményeteket, ötleteiteket, melyekkel jobbra, érdekesebbé tehetjük a mi kis kiadványunkat.

Ezúton szeretném ismét megköszönni a szerkesztőségünk szerzőinek és munkatársainak azt a sok munkát, melyet befektettek ezen újabb szám elkészítésébe.

2016. december 31.

Kardos Balázs (Balee)

IMPRESSZUM

Főszerkesztő: Kardos Balázs (Balee)

Szerkesztés, tördelés: Kardos Balázs (Balee)

Felelős szerkesztő: Egri Imre (Zimi)

Grafika: Molnár Péter (Mopi)

Rovatvezetők: Böszörményi Zoltán (Zboszor), Buzogány Csaba (Makranc), Mezei Róbert (M/ZX), Tanács Imre (Kapitány), Kardos Balázs (Balee), Lakatos Péter (Latyi.ca), László József (FPGA Jocó), Pgyuri, Taletovics Dávid (G.O.D)

Szerzők: Barabás Péter (z0d), Böszörményi Zoltán (Zboszor), Buzogány Csaba (Makranc), Lakatos Péter (Latyi.ca), László József (FPGA Jocó), Mezei Róbert (M/ZX), Náray József (Njosef) Pgyuri, Samu József (Sam. Joe), Sárfi Krisztián (Kiki), Székely Gábor (HA5AWS), Taletovics Dávid (G.O.D)

Szerkesztőség e-mail címe: specyalista.vilag.szerkesztoseg@sinclair.hu

Kiadó: Specyalista Baráti Kör <http://sinclair.hu>

2016. december

SPECTRUMOLÓGIA

SPECTRUM FEJLESZTŐI KÖRNYEZETEK - 2. RÉSZ

Az első számunkban olvashattatok egy összefoglaló írást arról, hogy miként látjuk a korabeli fejlesztéseket az eddig előkerült bizonyítékok alapján. Folytatjuk áskálódásunkat először is a Z80-as host gépek kategóriájában, elsőként vegyük górcső alá a TRS-80-at.

A processzor ugyanúgy Z80-as, a memóriaszerkezete is hasonló (állítólag), így ugyanarra a memóriacímre lehetett fordítani a programot, nem kellett relokálni, ráadásul a TRS-80-as billentyűzete sokkal professzionálisabb és strapabíróbb volt.

Matthew Smith

Nem kellett sokat kutakodni, hogy kiderüljön Matthew Smith első számítógépe valóban egy TRS-80 model 1 volt. A Tandy bolt előtt lógva ismerkedett meg más hasonló érdeklődésű sráccokkal és ebben a korban kezdte el írni első játékait is, először TRS-80-ra, majd VIC-20-ra és csak később Spectrumra is. Spectrumot a Bug-Byte szoftvercégtől kapott.

(Egy kis érdekesség, hogy ki tudja miért, de átfestette a Spectrumot metálzöldre. Szóval gyűjtők figyelem, ha valakinek az útjába kerülne, akkor az Matthew Smith-é volt!!!)

Első játékát – melyet kiadott a Bug Byte – 16 éves korában írta, de hozzátette, hogy „Elég nagy bukás volt ez a Bug Byte-nak. Azért engem ez nem igazán tört le. Azon a nyáron (1982) leültem megírni a Manic Minert. Rájöttem, hogy a Styx azért volt olyan gyenge, mert Spectrumon írtam, ahelyett, hogy a TRS-80-amat használtam volna fejlesztésre.”

Mesélt arról is, hogy a római nyaralásáról hazatérve, az ott vásárolt piros jegyzetfüzetbe vázolt Manic Miner tervek alapján 6-8 hét alatt, éjszakánként programozta le a játékot és hogy a TRS-80-at használta fejlesztőgépként, miután kibővítette mindenféle komponensekkel. Emiatt csak éjszaka

tudott dolgozni, mert amikor valaki bekapcsolta a teafőzőt, akkor a gép elszállt. Egy interjújából kiderül, hogy egy 5 Mbájtos merevlemezzel is büszkélkedhetett a TRS-80-a.



A Manic Miner kiadása után egyszer így nyilatkozott: „Így már sokkal könnyebb azt a játékot írni, amin most éppen dolgozom, a Jet Set Willyt, főleg miután megjavítottam a TRS-80 model IV-esem hardver hibáját.”

Valahol megemlítette, hogy miután a fejlesztések a TRS-80-on történtek, írt egy loader rutint, ami be tudta olvasni a Tandy szalagokat Spectrumon. Majd maga szolgáltatatta mindezek élő bizonyítékát, mikor elárulta, hogy ez a rutin benne is maradt a memória tetején és rákerült a végleges szalagra a Jet Set Willy-ben.

Utánanéztünk és lássunk csodát tényleg ott van:

„TRSHDX (C) Copyright 1982 by LSI”

Források:

<http://www.redkeyreddoor.com/?p=75> (ez mintha nem létezne azóta)
<http://jswremakes.emuunlim.com/Authors/Smith/mystery.htm>
<http://jswremakes.emuunlim.com/Authors/...ystery.htm>
<http://www.carlylesmith.karoo.net/spectrum/matsmith/>
<http://www.youtube.com/watch?v=wxUNPfrXZqE>

David Jones

A Magic Knight sorozat készítője is egy TRS-80-on fejlesztette Spectrum programjait. A konfigurációja egy TRS-80 model III volt 48K-ra



bővíve 2db 5 1/4"-os lemez meghajtóval (180K/oldal) és egy 15 Mbájtos merevlemezzel kiegészítve (ami fizikailag akkora volt, mint egy mai modern kisebb PC). Jones a WoS fórumában értekezett erről, ahol azt is megemlítette, hogy mennyibe is fájt ez akkoriban:

16k TRS-80 Model III	700 font
48K-ra bővítés +	
2db 5 1/4"-es floppy	700 font
5MB merevlemez	1200 font
15MB merevlemez	2000 font

“Végül rájöttem, hogy sokkal jobban megéri, ha inkább a 15M-ásat választom, így az lett.”

Így a teljes rendszer 3400 font volt egy fekete-fehér, 48K-ás számítógépért, ami a Spectrumhoz képest fél gőzzel ment. De ahogy meséli a lényeges különbséget a meghajtók jelentették.

„Nem is tudom, hogy lehet a Spectrumon direktben kódolni. Az én fejlesztői rendszeremben a forráskód egy másik számítógépen volt, így nem volt semmi rizikó, hogy a hibás kódom miatt minden megsemmisüljön.”

Egy interfésszel kötötte össze a Spectrumját a TRS-80 printer portjával, így küldte át a lefordított kódot a Spectrum memóriájába. Erről szintén mesél a WoS fórumában emígyen:

„Az illesztőt nem én építettem, John Campbell-től (Campbell Systems) kaptam. Ő írt egy Pac-Man klónt, ami évekig szerepelt a top listákon 84 táján. A játékaimhoz Ray Owen készítette a grafikákat, az apja Albert Owen ismerte Johnt, aki megkérte, hogy támogasson meg engem egy ilyen illesztővel és a hozzávaló szoftverrel. Az illesztő alapja egy 8255 PIO volt kiegészítve pár TTL kapuval. Úgy emlékszem, hogy a Spectrum bővítő buszára csatlakozott, a másik vége pedig a TRS-80 printer portjára.”

A szoftver

A szoftver igen egyszerű volt. A TRS-80 512 bájt fejlécezt küldött először, ami 256 darab 0 majd 256 darab 255 volt. Ezután következtek a bájtok egymás után némi késleltetéssel közöttük. A Spectrum megvárta a fejlécezt, majd ha megjött, elkezdte tölteni a bájtokat szépen sorban. A Spectrum oldalon nyilvánvalóan nagyobb késleltetéssel, miután majdnem kétszer gyorsabb volt, mint a TRS-80.

Források:

<https://www.worldofspectrum.org/forums/discussion/45715/>

Psion

Amikor elkezdtek fejleszteni a Horizons Tape-et, amit aztán minden ZX Spectrumhoz



mellékeltek, vásároltak egy TRS-80-at lemez-egységekkel és a fejlesztést ezen végezték. Emellett építettek egy saját illesztőt, amivel összekötötték a Spectrummal, de erről pontosabb információ nem látott napvilágot ezidáig.

Csak érdekességként említjük meg, hogy a Psionnál hamar rájöttek arra, hogy nem kifizetődő mindig egy új fejlesztő környezet felállítása, amikor egy új hardver jelenik meg a piacon, ezért kicsit visszaforgattak a nyereségből és vettek pár VAX 750 VMS-t. A QL-hez készült programok már ezen készültek, de erről talán majd később értekezünk.

Forrás: The Home Computer Advanced Course - Issue 20

Summa summarum

A kezdeti feltételezéseinket nyilván sikerült igazolnunk a megtalált példák alapján, azonban az is nyilvánvaló, hogy további példák után is érdemes lehet kutatni. David Jones-féle történetből mindeképp feltárult előttünk egy újabb nyom, aminek feltétlen utána kellene járnunk, nevezetesen John Campbell tevékenységének illetve programjainak.

Ugyanakkor folytatjuk kutatásunkat a további Z80 alapú host gépek elemezgetésével.

**Kardos Balázs (Balee) és
Lakatos Péter (Latyi.ca)**
(Folytatjuk...)

TRS-80 SPECIFIKÁCIÓK

Model III

CPU: Z-80A 8-bit Processzor (2.03 MHz)
ROM: 4K ROM (Level I) vagy 14K ROM (Model III BASIC – (c) Tandy és Microsoft)
RAM: 4K – 48K
Megjelenítés: 12" FF monitor; text 64x16; grafika 128x48
Billentyűzet: Teljes méretű írógép jellegű, 65 gomb numerikus billentyűzettel
I/O: Kazettás magnó illesztő (Level I: 250 Baud; Level II: 500/1500 Baud) számítógép vezérelt
Portok: Párhuzamos port; RS-232-C (soros) port
Opcionális: Külső, belső floppy meghajtók; Hi-Res grafikus kártya

Model IV (változások)

CPU: Z-80A 8-bit Processzor (4 MHz)
Billentyűzet: teljes méretű írógép jellegű, 70 gomb numerikus billentyűzettel + control, caps lock és 3 programozható funkció billentyű (F1, F2, F3)
Megjelenítés: text 80x24 (Model 4 módban), 64x16 (Model III módban)
Nyelv: Microsoft 5.0 BASIC
Hang: BASIC-ből hozzáférhető; "keyclick"
Floppy: 1 vagy 2 beépített DD 184K 5-1/4"

OLVASOKKOLÓ

EGYESÜLETI ÉLET, AHOGY AZT PGYURI ELKÉPZELTE

... még 2009-ben

ZX Spectrum Személyi Számítógép
Hagyományörző Klub Egyesület



- évi 8 rendezvény, belépés
16, 48 évesek 32 % kedvezménnyel,
48+ korúak 16 %-al,
128-asok ingyen ...
- saját újság a ZX v(n)agyok ...
- közgyűlések, szavazások a sinclair.hu oldal
színeiről ...
- kapcsolatfelvétel az ULAnbatorban működő
társzervezetekkel ...
- Horace és a Pókok állandó ízeltlábú kiállítás
rendszeres megtekintése ...
- Manic Miner bányalátogatás és túra ...
- Jet Set Willy szelektív üveg- és kupakgyűjtő
akció ...
- Radírgumigombos Varró kör életre hívása ...
- kicsi gyerek részére hangyaboly építő szakkör az
Ant Attack térkép alapján a "Sandy White"
homokvárban ...
- ugróriskola az ifjú és tehetséges Jumping Jack-
eknek ...
- saját kutya, akit Bézik-nek hívnak (törzskönyvi
neve Beginners All-purpose Symbolic
Instruction Code, röviden BASIC,
ejtsd Bézik) ...

Így aztán, aki tud "Bézik üli!", adhat
neki - de kizárólag szintaktikailag
helyes - parancsokat úgy, mint



Bézik! RUN !

Bézik! LOAD "kaja" !

Bézik! INPUT keksz !



Bézik! GO TO helyedre !

Ha butaságot csinál,

Nonsense in Bézik !



de talán neki legkedvesebb játék a

Bézik! CAT 1 !



szerencsésebbek CAT 8 !



Pgyuri

Morzsa 1984-ből...

**Számítógépes
játékok lányoknak**

A személyi számítógépekkel forgalmazott játékok népszerűek gyermekeink körében. Úgy tűnik, hogy a lányos szülők hátrányosabb helyzetben levőnek érezhetik magukat, hiszen a játékok – nagyon kevés kivételtől eltekintve – fiúknak szólnak, sőt szemléletük erőszakos. Az amerikai piacon megjelent az első olyan játéksorozat – az ismert Addison-Wesley kiadó forgalmazásában –, amely kifejezetten lányoknak szól. Ezt a szoftvert nők is írták. A sorozat játékeinak olyan címei vannak, mint a Préri Jenny-je vagy a Clair a barlanglakó. Ezekben a játékokban természeti csapásokat kell „túlélni”, előfordulnak vadállatok is, de itt ezeket nem „lelőni” kell, hanem el kell futni előlük.

ARCKÉPCSARNOK

MATTHEW SMITH

Csillag születik

Matthew Smith neve valószínűleg egy spectrumosnak sem ismeretlen. Ő írta – többek közt – a Manic Minert és a Jet Set Willyt. A Manic Miner, ami az egyik legikonikusabb Spectrumos játék, aminek számos remake-je és nem hivatalos kiegészítője látott napvilágot az elmúlt 30 évben. No, de kezdjük az elején...

Matthew 1966-ban született Londonban. Első számítógépét karácsonyra kapta 1979-ben, ami egy Tandy TRS-80 volt. Erre írt is egy Delta Tower One című Galaxian klónt, amiből szerinte kb. 50 példányt adott el a kiadó, ez jónak számított akkor (1980-at írunk!). Matthew csak este tudott dolgozni a gépen, mert a TRS-80 mindig elszállt, ha valaki bekapcsolta a teafőzőt.

Második kiadott játéka a Monster Muncher volt, ami VIC-20-ra készült BASIC-ben.

A Miner Willy franchise

1983-ban a Bug-Byte cég megbízta Matthew-t 3 spectrumos játék írásával. A cég adott neki egy ZX Spectrumot, amin fejleszthetett. Az első játék a Styx volt, 16K-s Spectrumon futott és 5,95 GBP-be került. A játék célja eljutni a kijáráshoz, miközben szörnyek próbálnak elkapni minket, akiket le lehet lőni. A játék nagyon sikeres volt, Smith-nek harminzezer angol fontot hozott, a Bug-Byte pedig neki adta a Spectrumot is.

Ezután szabadságra ment Olaszországba, ahova magával vitt egy piros jegyzetfüzetet. Ebbe a füzetbe rajzolta a Manic Miner pályáit és terveit. Hazatérése után 8 héttel elkészült a játék, aminek

grafikáját és zenéjét is ő készítette. Ez volt az első spectrumos játék, aminek folyamatos zenéje volt, nem csak effektek. Matthew egyik kedvenc játéka a Donkey Kong volt, amit játéktérmekekben játszott.

Viszonylag drága szórakozás volt, ezért a Manic Minert kicsit a Donkey Kong mintájára csinálta meg, illetve egy 1982-es Atari játék, a Miner 2049'er is inspirálta. A fejlesztés Tandy-n történt, ami szintén Z80 alapú, viszont rendelkezett 5 Mb-ajt merevlemezzel, ami felgyorsította a programozást. Valószínűnek tartom, hogy a gumi billentyűzet hiánya is inkább előny volt. Egy érdekesség, a Manic Minert később Matthew cége, a Software Projects is kiadta. Ezt a Bug-Byte-os szerződés egy apró

hibája miatt tehette meg, a játékot 'átvette' a cégtől. Elég sok platformra portolták, többek közt Amigára, Commodore 64-re illetve később megjelent Gameboy Advance-ra is. Egy interjúban a szerző elmondta, hogy a keresett pénzt ész nélkül elszórta. Nem hibázhatom, 17 évesen kevés ember hoz jó pénzügyi döntéseket. Néhány éve azt nyilatkozta, hogy a mai eszével házat venne annyi pénzből.

Míg a Manic Miner fejlesztését nagyon élvezte, a Jet Set Willy – az ő szavait idézve – egy pokol volt, menedzser lihegett a nyakába és nem volt elég idő a tesztelésre. A bolti verziót nem is lehet végigjátszani több programhiba miatt, ezekre később a Bug-Byte POKE-okat adott ki. A fejlesztési idő kb. háromszorosa volt a Manic Minerének, pedig a grafikát már saját programjával rajzolta a játékhoz. Egy 20 évvel későbbi interjúban Matthew elmondta,



hogy még soha nem játszotta végig a Jet Set Willyt. A Manic Minerhez hasonlóan, ebben a játékban is folyamatos zene hallható, a menüben Beethoven Holdfény szonátája, a játékban pedig a "Hegedűs a háztetőn" darab, "Ha én gazdag lennék" című muzsikája szól. A játék 1984-ben jelent meg és a Manic Minernél is nagyobb sikert aratott. Több hónapig uralta a spectrumos eladásokat.

Matthew eredeti terve az volt, hogy a végén Mariát ágyba kell vinni, aztán Maria megöli Willy-t. Saját bevallása szerint "alkohol és egyéb ártalmas szerek" befolyása alatt írta a játékot. A fejlesztés befejeztével két hónap szabadságra ment, mire visszatért Liverpoolba, már többen a játék folytatásáról beszéltek. Ez 1985-ben el is készült a Software Projects kiadásában – de nem ő írta meg végül – a neve Jet Set Willy II: The Final Frontier lett. Ez nem teljesen új játék lett, inkább a Jet Set Willy kiegészítője - ma valószínűleg DLC-nek hívnánk.

Rögös utakon

A Jet Set Willy után egy The Mega Tree nevű játékon kezdett el dolgozni, amit Commodore 64-re írt, ekkor már saját cégénél. Ez nem vonzott elegendő tömeget és Matthew üzleti partnerei türelmetlenek voltak, ezért 3 hónappal a fejlesztés kezdete után leállt a projekt.

1987-ben több játékmagazinban is hirdetések jelentek meg egy új játékról, az Attack of the Mutant Zombie Flesh Eating Chickens From Mars-ról, amit szintén Matthew írt, de soha nem adott ki. Állítólag azért, mert nem volt megelégedve a programmal, miután elkészült.

1988-ban bezárta Software Projects cégét, aztán eltűnt a színről, részben anyagi okok miatt. 1995-ig Wallasey-ben élt, ahol felnőtt, egy ideig kórházban is volt, a pszichiátrián kezelték. Ennek több oka is volt, saját bevallása szerint kiégett. Nem volt nagy titok az sem, hogy nem vetette meg a drogokat. A Manic

Miner fejlesztése alatt is használt kábítószereket, némelyik szörny ennek az eredménye.

Az internet terjedésével páran feltették a kérdést, hogy mi történt Matthew Smith-szel. Sok pletykát lehetett olvasni, hogy éppen hol van és mit csinál. Páran biztosan emlékeznek a "Where is Matthew Smith" oldalra.

1995-ben Hollandiába utazott, hogy helyretegye az életét. A



kompig stoppal jutott el.

Egy kommunában lakott és többféle munkát is elvállalt, de egyik sem játékfejlesztés volt.

Dolgozott élelmiszer-gyárban illetve szupermarketekben virágokat is pakolt.

1997-ben kiutasították az országból, mert nem

újította meg a tartozkodási engedélyét. Miután visszatért Angliába és felfedezte az Internetet, meglepődve tapasztalta, hogy sokakat érdekel, mi is lett vele.

Matthew Smith újrátöltve

1999-ben egy Runecraft nevű cégnél kapott állást. A cég egyetlen játéka a Scrabble volt Game Boy Color-ra mielőtt csődbe mentek volna.

2010-ben az Elite Systems újra kiadta – több Spectrumos játékkal együtt – a Manic Minert és a Jet Set Willyt iOS-re, Androidra és a Kindle tabletekre is. Később megjelent Windows Phone-ra is.

2013-ban az Elite Systems bejelentette, hogy Matthew Smith-szel együtt egy új játékon dolgoznak. Sajnos ennél többet nem árultak el és azóta sincs több infó róla. A mostánában készült interjúkból sajnos az látszik, hogy az élet megviselte Matthew-t. Nem lett happy end a korai sikerekből. Reméljük még hallunk felőle és visszatér egy felejthetetlen alkotással.

"Five years ago I was a washout. Ten years ago, I was history... Now I'm a legend." - Matthew Smith

"Öt éve egy katasztrófa voltam. Tíz éve történelem ... Most viszont legenda vagyok." - Matthew Smith

Barabás Péter (z0d)

Mysterious Dimensions (Hooy)



A Hooy-Program egy nemzetközi formáció, amit a lengyel Yerzmyey alapított 1989-1990-ben. A program csak 128K-s gépeken játszható, cserébe egész idő alatt szól Yerzmyey hangulatos zenéje. Ő találta ki a játékot, rajzolta a grafikát, sőt ő írta a játék AGD kódjait is. Segítségére a cseh Hellboj volt, ő kódolta ezt a gyűjtögetős platformjátékot. A Mysterious Dimensions egy aranyos kis mese. Történet nincs, egy kisfiút kell irányítani négy különféle terepen, virágos réteken, egy várban, az erdőben és a barlangban. Mindegyik helyszínt 11 képernyőn keresztül járhatod be, és mindegyiken más-más értékeket kell begyűjtened, csak csillogjon! A környezet nagyon szépen megrajzolt, kihasználja a Spectrum teljes színekészletét. A játék könnyedén végigjátszható egyhuzamban, köszönhetően a 99 életnek.

Davey Sludge újabb örült ötlete, aminek természetesen köze van kedvenc témájához, ugyanis a sztori szerint a Munkaügyi Központ kihelyez téged közmunkára a városi csatornahálózatba. Heti 39 órát kell dolgoznod 70 fontért cserébe. Na, most szó szerint bűdös a munka! A csatorna lakói, azok az áldott jó lelkek, szétszaggatták a csatornahálózat egy-egy részét, hogy kedvükre dagonyázhassanak a posványban. A te feladatod a csatorna darabjait a helyére lökdödni, ez a célhely pedig a képernyő tetején van. A csatornadarabokat boxkesztyűvel tudod feljebb ütögetni. Amint sikerül egy egyenes csatornaszakaszt létrehoznod, mehetsz a következő szétesett darabhoz. Közben természetesen a kanális lakóközössége próbál téged megakadályozni munkád elvégzésében.

Sewer Rage (The Death Squad)



Strange Kitchen (textvoyage)



A furcsa konyha egy Boriel ZX Basicben írt lövöldözős játék. Egyszerű recepteket kell elkészíteni, elpusztítva az ételhez közelítő nemkívánatos összetevőket! 15 egyre nehezedő szinten kell helytállnod, egyre fenyegetőbb és gyorsabb élelmiszerek ellen. Minden szint előtt láthatod, hogy milyen ételt kell elkészítened, és ennek milyen összetevői vannak. Csakis ezeket az alkatrészeket szabad és kell is átengedned a védelmi vonaladon. Minden szinten addig kell ellenállnod, míg el nem készül az étel, ennek a folyamatnak az állapotát a jobb felső sarokban lévő kijelzőn követheted. 100%-nál léphetsz a következő szintre. És, ahogy textvoyage, a szerző írta, ha úgy gondolsz, hogy ez így nincs rendben, légy hálás, hogy nincs melletted Gordon Ramsey, és nem üvöltözik veled!

KLÓNOK HÁBORÚJA

AMIT SOSEM MERTÉL MEGKÉRDEZNI A ZX EVOLUTIONRÓL 2. RÉSZ

Az első részben a ZX Evolution több lehetőségéről csak pár mondatot ejtettünk. Következzenek most ezek bővebb leírásai. Az Evo Reset Service (ERS) főmenüjén van egy „G” opció, amellyel kiválaszthatjuk, hogy reset után a rendszer milyen üzemmódban induljon el. Ezek a lehetőségek a már ismert „Evo Service” (azaz ERS), a „GLUK Service”, az „EVO ProfROM” és a „Custom ROM”. Az utóbbit nem ismertetjük, az mindig attól függ, hogy milyen ROM-ot éget be a felhasználó.

GLUK Reset Service



A GLUK Reset Service, az orosz Wikipédia leírása szerint egy 1996-ban kifejlesztett menürendszer és inicializáló kód, amelyet a Pentagon nevű klónhoz fejlesztettek ki. Erre azért volt lehetőség, mert a Pentagon 64KB ROM-jából az egyik 16KB-os rész nem volt addig használva. A többi három 16K-s részében a 48K-s és az 128K-s ZX Spectrum ROM-ok, valamint a TR-DOS foglaltak helyet. A GLUK fő erőssége, hogy – eltérően pl. a 48K-s ROM-tól, ami memóriatesztet is végrehajt –, reset-eléskor nem törli a RAM tartalmát, ezért programkód debuggolására, illetve feltörésére kiválóan alkalmas. Ezért, bár egy menürendszert is kapott, valójában a programozók, és nem a felhasználók kényelmére készült. A Mr. GLUK kódnevű programozó a TR-DOS rendszert kiegészítette IDE felületű eszközök kezelésével is, ez lett a 6.x-es TR-DOS sorozat. A ZX Evo ROM-jában szereplő változata a 6.12E, amely képes RAM-

diszket is kezelni. 2000 óta a GLUK Reset Service-t AloneCoder fejleszti tovább.

Az IDE eszközök kezelése miatt már a floppy mellett képes merevlemezről és CD-ről is programot indítani, bootolni.

A menüpontok, bár kisbetűvel vannak írva, a bennük előforduló egy-egy nagybetű jelzi azt a gombot, amellyel a menüpont közvetlenül is aktiválható, a szokásos fel/le kiválasztás és Enter gombok mellett. A menüben a „P.c” illetve „p.C.512” menüpontok a már említett Perfect Commander kétféle verzióját takarják.

A 48K BASIC mód egy módosított/javított ROM-ot tartalmaz, amelyet a megváltozott copyright üzenet is jelez: „© 2006 Sinclair Research Ltd”

A 128K BASIC mód indításakor a megszokott 128K menüt kapjuk, de itt is megtalálható a TR-DOS menüpont.

Amennyiben a RAM-ba be van töltve előzetesen az XAS assembler, vagy az STS monitor illetve debugger program, akkor ezekkel a menüpontokkal azt indíthatjuk el. A #51 illetve #57 az az érték, amit a \$7FFD portra ki kell írni, így a 128K-s gépen a megfelelő RAM lapot belapozva, arra kell betölteni az XAS illetve STS programokat. A rendszer onnan fogja indítani.

EVO ProfROM

A ProfROM egy professzionális monitorprogramot és további menüből elérhető, főleg diszkkal kapcsolatos lehetőségeket tartalmaz. Eredetileg a SMUC nevű IDE bővítőkártya részeként fejlesztette ki a Scorpion klón gyártó cég. Ez a bővítő két IDE felületű eszközt (merevlemez, CF kártya, CD-ROM, stb.) képes kezelni, opcionálisan valós idejű óra IC-t, illetve tartalmaz mintegy 2KB NVRAM-ot a beállítások megőrzésére. A SMUC egy ZXBUS felületű bővítőkártya, amely többek között a ZX Evolution-al is kompatibilis, és újabb ERS ROM verziók kezelik is az SD-kártya illetve a beépített Nemo IDE mellett. A ProfROM egy módosított változata került be a ZX Evo ProfROM menüje alá. A módosítás lényegi része, hogy kezeli az ZX Evolution-ban lévő Nemo IDE nevű IDE csatolót is. Az EVO ProfROM a 48K-s és 128K-s ROM-ok mellett



továbbfejlesztett ROM-okat is tartalmaz. Ezt mutatja az alábbi menü is. Nagy változás, hogy a TR-DOS már működik 128K-s BASIC-el is.



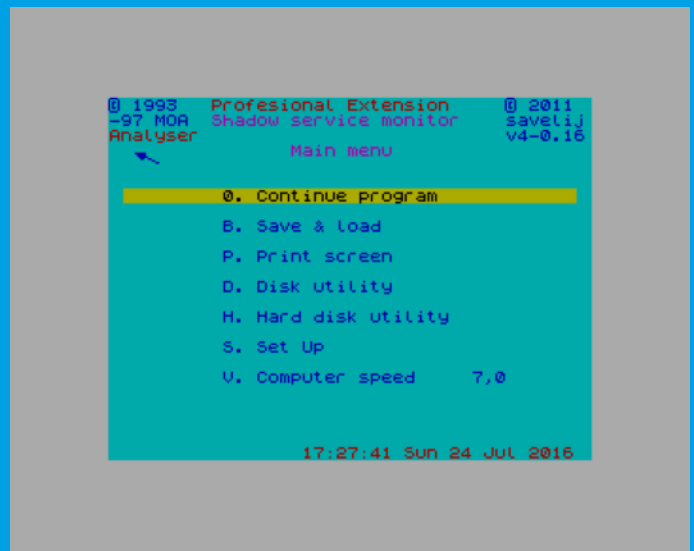
ProfROM NMI menü

Az NMI gomb megnyomására egy komplex menürendszerbe lépünk.

Ha csak egy szalagról betöltött játék aktuális állapotát (snapshot) akarjuk lemezre menteni, akkor elegendő a „Save & load” - „Magic button” menüt használni. Ez az eredeti Betadisk illetve Betadisk 128 bővítővel azonos formátumban menti le a memória tartalmát és a CPU állapotát. Ha 128K-s gép játékot mentettünk, akkor több fájl is keletkezik. Betölteni a GOTO "filename"CODE paranccsal tudjuk. A teljes leírást a <http://www.worldofspectrum.org> oldalról a Betadisk 128 kézikönyvből tudhatunk meg.

Ezen kívül a megfelelő almenükben például floppy-t formázhatunk, merevlemezt particionálhatunk és

formázhatunk. Nem kompatibilis a FAT32 fájlrendszerrel és nem képes túl nagy partíciókat sem kezelni. Pl. egy 120GB-os SSD egyetlen FAT32 partícióját kb 48GB-osnak látja, és nem képes megnyitni a rajta lévő fájlokat. Amennyiben viszont van szabad hely a lemezen, akkor képes saját partíciót létrehozni 0x53-as partíció típussal. Nem valódi fájlrendszer, csak 640KB-os TRD lemezképek sorozatát tartalmazza. A partíció létrehozása után MFS típussal jelenik meg a ProfROM listájában. Ezután egy virtuális lemezképet felcsatolhatunk a



ProfROM menüjéből az A: ... D: lemezcímkék alá, és például TR-DOS floppylemez tartalmát másolhatjuk rá a TR-DOS alól. Sajnos olyan másolóprogram (egyelőre?) nem létezik, amely a ZX Evolution-on futna, és megosztott diszk vagy CF kártya esetén a FAT32-re másolt TRD lemezképeket közvetlenül tud a ProfROM MFS partícióján egy virtuális lemezképre rámásol vagy fordítva.

IS-DOS

Az orosz Iskra Soft fejlesztése volt eredetileg, egy kereskedelmi DOS rendszer volt az orosz ZX Spectrum klónokhoz. Nem TR-DOS formátumú floppy lemezeket használ, és saját partíció típust. Így csak akkor használható, ha valódi floppy drive és opcionálisan merevlemez is csatlakoztatva van. A merevlemez fájlrendszere nem kompatibilis az MS-DOS FAT12/FAT16/FAT32 rendszereivel, valamint lényegében csak orosz nyelvű leírás található meg erről a DOS rendszerről, emiatt kipróbálása nehézkes és a NedoPC csoport elfoglaltsága miatt nem is kaptam több információt.

A TSconf konfiguráció

Az ukrán Val Chefranoff (TS-Labs) fejlesztése. Szintén ő módosította az Unreal Speccy emulátort úgy, hogy ennek a firmware-nek a lehetőségeit is emulálja.

A TSconf egy átfogó átírása a ZX Evolution FPGA kódjának, valamint kapott egy saját 64KB-os ROM kódot is. Az FPGA képességei a következők:

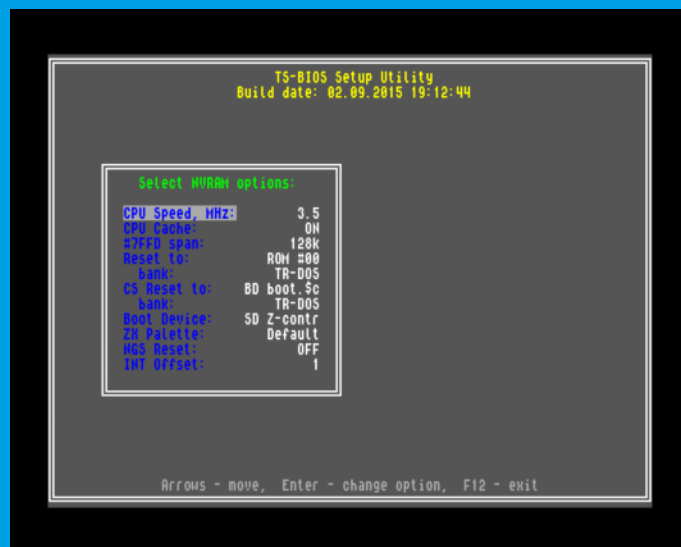
1. Pentagon időzítések (az Evo BaseConf firmware Spectrum 48K és 128K időzítéseit még nem portolták, ahogy az ULA+ lehetőségeket sem)
2. 1MB RAM engedélyezése a Spectrum felé, a többbit saját céljaira használja
3. CPU cache megvalósítása. A ZX Evo 3.5, 7 és 14MHz-en képes működtetni a Z80 CPU-t, ebből a legnagyobb sebességnél már az alaplapon alkalmazott RAM sebessége miatt wait state-ekre kényszerül a CPU, effektív sebessége 11MHz körül alakul. Az FPGA sebessége ennél lényegesen nagyobb és hidat képez a CPU és a RAM között. A cache vezérlése 16KB-os szeletekben valósítható meg, a CPU négy memóriaszeletére egymástól függetlenül. A cache ellensúlyozza a RAM wait state-eket, és a CPU a tényleges 14MHz-es sebességet képes elérni.
4. DMA kezelése. A ZX Spectrumhoz sokféle DMA-t használó bővítés készült, például az MB02, amelyből később a Velesoft-féle DataGear is készült, ami nem tartalmaz diszkbővítést, csak a DMA-t különféle grafikai trükkökhöz. Az MB02-vel és a DataGear-rel ellentétben, amelyek a Zilog által a Z80-hoz készült DMA IC-t használják, a TSconf az FPGA-val valósítja meg a DMA képességeket. A DMA IC a CPU órajelével működik, és maga is wait state-ekkel fogja vissza a CPU működését. Ezzel szemben a TSconf DMA implementációja az FPGA 28MHz-es órajelével működik, amelyből a CPU órajele felezett, negyedelt illetve nyolcadolt osztásokkal áll elő. Mivel a DMA ezen a sebességen működik, a CPU órajel-ütései között képes elrejteni az adatátvitelt, ezzel a CPU számára teljesen transzparens. A DMA nem cache-koherens, vagyis ha a DMA másolás célja RAM, akkor arra a memóriaszeletre ki kell kapcsolni a cache-t. Ez a DMA implementáció egy DMA csatornát tartalmaz.
5. Programozható interrupt vezérlő. Egymástól függetlenül letiltható / engedélyezhető függőleges és vízszintes raszter-interruptok és DMA interrupt.
6. I/O porton keresztül kiolvasható a videóraszter pozíciója.

7. Tile és sprite engine. Független és smooth scrolling tile plane-ek. 64x64 pixel méretű sprite-ok kezelése, soronként maximum 42, de a a sprite engine minden vízszintes raszterinterruptnál történő újraprogramozásával a teljes képernyőn korlátlan számú sprite megjelenítése lehetséges.
8. Az eredeti 256x192 az „attribute clash”-ról jól ismert videómód mellett 360x288, 320x240 és 320x200-as felbontások 256 és 16 színű palettás üzemmódok, és 80x25-ös karakteres megjelenítés is lehetséges.

A TSconf ROM-jában a szokásos 48K, 128K Spectrum ROM-ok és a TR-DOS mellett helyett kapott egy BIOS is, amely az alapbeállításokért felelős.

A beállítások:

1. CPU órajel, a ZX Evolution alaplapon elérhető 3.5, 7 és 14MHz lehetőségekkel
2. CPU cache ki- vagy bekapcsolása
3. A Z80 számára elérhető belapozható



memóriamennyiség: 128K, 128K auto, 512K és 1024K. A 128K egy zárolt mód amely csak 128KB lapozását engedélyezi. A „128K auto” a használt OUT (C),A illetve OUT (#FD),A gépi utasításokat detektálja, és a #7FFD portra írt 5-6. biteket engedélyezi, illetve maszkolja, ezzel 512KB illetve 128KB csak lapozását engedélyezve. Az 512K illetve 1024K módok nem detektálják az OUT utasítás típusát, a megfelelő mennyiségű memória lapozását engedélyezve.

4. „Reset to” - az F12 illetve reset esetén mi induljon el. A lehetséges beállítások: ROM #00 (a TSconf „Custom ROM”), ROM #04 (a második 64KB-os ROM rész az 512KB-ból), RAM #F8 (kísérleti ROM-ok tesztelésére szolgáló 64KB-os RAM rész, az F8-FB számozású lapok tartalma), „BD boot.\$c” - a boot device FAT32 fájlrendszer gyökerében elhelyezkedő „boot.\$c” Hobeta fájl, „BD

sys.rom” - a boot device gyökerében lévő „sys.rom” fájl, ami 64KB-os ROM tartalomként lesz értelmezve.

5. **Bank:** Amennyiben ROM típust (ROM #00, ROM #04, RAM #F8 ill. „BD sys.rom”) választottunk, akkor sorrendben a négy darab 16KB-os ROM rész a 64KB-os ROM-ból a következők: SYS (#0), TR-DOS (#1), „Basic 48” (#2) és „Basic 128” (#3)
6. **CS Reset to:** Caps Lock + F12 illetve Caps Lock + reset esetén mi induljon el. A lehetőségek ugyanazok, mint a „Reset to” esetén, saját ROM bank beállítással.
7. **Boot device:** kiválasztható a boot eszköz, úgy mint „SD Z-contr” (a ZX Evolution beépített SD kártya-olvasója), „IDE Nemo M” és „IDE Nemo S” (a ZX Evolution beépített IDE csatolójára csatlakoztatott master illetve slave eszköz), RS-232 (soros kábel), „IDE Smuc M” és „IDE Smuc S” (a korábban már említett SMUC bővítés master illetve slave eszköze)
8. **ZX Palette:** az alapértelmezett színpaletta
9. **NGS Reset:** reset esetén a csatlakoztatott NeoGS hangkártyát is reset-elje-e.
10. **INT offset:** a kanonikus Pentagon videó időzítésétől való eltérés CPU órajelekben, alapértelmezetten 1.

A TSconf is a PS/2 billentyűzeten keresztül éri el az extra funkciókat, mint az BaseConf. Ezek a funkciók viszont máshol érhetők el, mint ahogy a BaseConf-nál megismertük, általában a (Windows alá készült billentyűzeteknél látható) „Menü” gomb és egy másik billentyű kombinációjával.

Print Screen (SysRq): NMI

F12: meleg reset

Ctrl-Alt-Del: hideg reset

Ctrl-Alt-F11: „Easter Egg” mód, egy régi TV-játékot jelenít meg.

Ctrl-Alt-F12: a BaseConf és TSconf közötti váltás (bár az AVR kódot a TSconf adja, így a billentyűzetkezelést is, ha beégettük a TSconf-ot)

Scroll Lock: VGA / RGB módváltás

Menu + F1: floppy swap (A: ... D: váltások, ami az ERS főképernyőn érhető el)

Menu + F2: a beeper, tape IN és tape OUT közötti váltás

Menu + F3: 50Hz / 60Hz közötti váltás

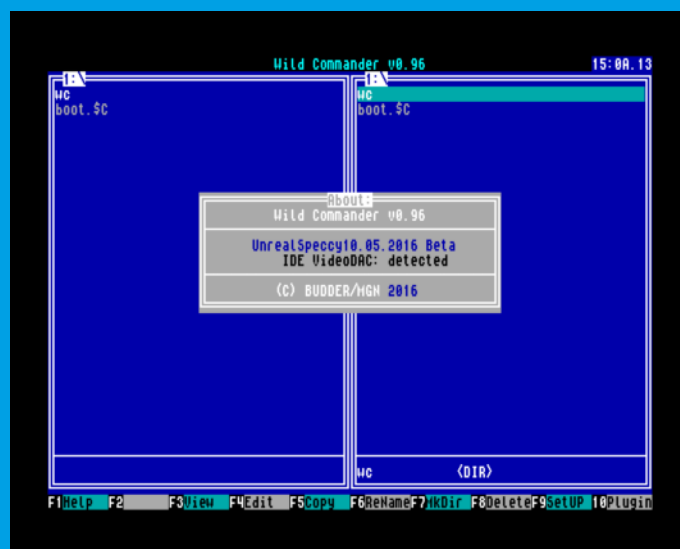
Menu + F4: videó pozitív / negatív polaritás váltás. Monitortól függ, hogy melyiket célszerű használni.

KeyPad +, -, *: PS/2 egér felbontásának beállítása

A TSconf fő kezelőfelülete egy „boot.\$c” boot fájlban és különböző modulokkal megvalósított „Commander” jellegű fájlkezelő, a Wild Commander. Modulokon keresztül van megvalósítva többek között a TAP és TRD fájlok felcsatolása, betöltése, és a

különböző snapshot (Z80, SNA, SPG) fájlok vagy a \$C típusú Hobeta fájlok indítása is. Van továbbá szövegfájl és BMP néző, vagy MP3 lejátszó, ha a NeoGS hangkártyánk is van. Ezen cikk szerzője apró mértékben hozzájárult a Wild Commander fejlesztéséhez, a ROM-Writer modult bővítettem ki a RAM #F8 lapjának kezelésével. Ezt főleg amiatt, hogy pl. Interface II ROM játékokat ki lehessen próbálni anélkül, hogy a Flash ROM korlátozott mennyiségű törlési ciklusait elhasználnánk, akár az ERS „Custom ROM” felülírási lehetőségével, akár a Wild Commander ROM-Writer moduljának használatával.

A Wild Commander önmagában is megérne egy cikket, az érdeklődő programozó számára gyakorlatilag egy operációs rendszer szolgáltatásait nyújtja, beleértve a minimálisan elvárható fájlkezelést is. Akár játékok vagy segédprogramok is megírhatók Wild Commander modulként.



A fentebb leírt lehetőségekkel (új videó módok, CPU cache, blitter DMA, és a röviden említett NeoGS hangkártya) gyakorlatilag egy Z80 alapú Amiga-gyilkos gépet kapunk. ZX Evolution a TSconf firmware-rel egy olyan ZX Spectrum kompatibilis gép, amely mintha egy alternatív univerzumra nyitott volna kaput. Ha 1985-86-ban lett volna erre lehetőség, akkor a 16-bites forradalom talán meg sem történik, és Sir Clive sem rokkann bele pénzügyileg a C5-be, és mindannyian CP/M-et használnánk grafikus felülettel.

Böszörményi Zoltán (Zboszor)

(Folytatjuk ...)

LOAD " "

JÁTSZD ÚJRA FRED!

Itt jön Fred újra! Nagyszerű lehetőség elővenni ismét ezt a fantasztikus játékot! Biztos mindenki emlékszik erre a barátságos fickóra, aki kincseket gyűjt piramisról-piramisra járva, miközben szellemek és múmiák próbálják megakadályozni benne.

Endre bácsi szerette ezt a játékot és sokat játszott vele unokájával együtt. Egyszer elhatározta, hogy csinál benne pár apró fejlesztést, hogy izgalmasabb legyen. Egy nap megláttam ezt a verziót és azonnal arra gondoltam, be kellene mutatni. Endre bácsit nem hozta lázba a dolog, de odaadta és megengedte, hogy mindenki számára elérhető, publikus legyen.

Eredeti ZX Spectrum 48-ason fejtette vissza saját disassembler programjával és mindent fejben és papíron oldott meg, mint anno a hőskorban.

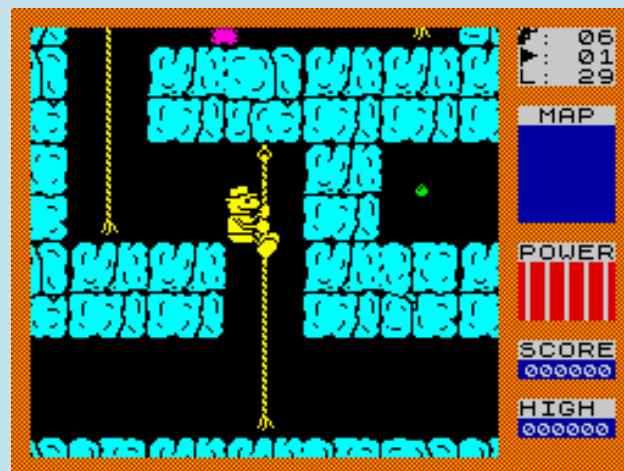
Mit is csinált?

Először is kijavított néhány hibát,

- ... aztán megtanította Fredet lőszert tenni a zsebébe,
- ... lehetőséget adott több életerőt gyűjteni,
- ... adott neki egy vöröskeresztes csomagot, ami egyszer megmentheti az életét,
- ... végül gyorsabbá és kisebbé tette a játékot jelentős változtatás nélkül.

Hogy kell ezt érteni?

- Nincs pontszám hiba (néha a pontszám felszaladt 150.000 pontra egy kincs felvételénél).
- Nincs csepp hiba (az eredetiben csak 1 csepp esett egy pozíción, de tudott volna 2 és 3 is esni).
- Több, mint 6 lőszerral lehet mászkálni.
- Maximális életerő 15-ről 40-re nőtt.
- Extra életerő bevetése, ha Fred energiája kevesebb, mint 5 egység.
- Kb. 3-szor gyorsabb játékmenet (egy hibás ellenőrző ciklus javításával).
- 4 Kb-ajt memória megtakarítás.



Továbbá az "M" lenyomására megáll a játék és ...
... "Enter"-re indul újra,
... "L"-re egy új piramist készít és újraindítja a szintet,
... "H"-ra beveti a vöröskeresztes csomagot (ha életeroje kevesebb, mint 5 egység).

Egyéb ötletek a jövőben?

Például egy véső és egy kalapács a falon átjutáshoz egyszer vagy kétszer ... esetleg pár rúd dinamit ...

Tisztelgés a készítőknak:

Fernando Rada

Paco Menendez

Carlos Granados

Endre bácsi emlékére,

Pgyuri

Letöltés:

Endre bácsi által módosított verzió letöltése.



BOLDOG KARÁCSONYT! - HORACE

Steve Broad 2013 decemberében jelentette meg első Horace főszereplésével készült játékát, a 'My rendition' of Horace to the Rescue-t. Fél éven belül megjelent első játéknak második epizódja és két Horace Miner rész is.

Majd hosszú, több mint két éves csend után állt elő a Merry Christmas from Horace-szal, valamint A Present from Horace címmel egy microdrive kiadással, ami Steve korábbi négy Horace-témájú játékát tartalmazza. Steve Horace játéka nem hivatalos verziók, nem az eredeti Melbourne House produkciók folytatásai, inkább egy különálló angol sorozatnak tekinthetők.

Sztori nincs, de valószínűsíthető, hogy Horace csak Mikulásnak segít ajándékokat gyűjteni, közben pedig egyre közelít az Északi-sark felé. Így adott a feladatod: irányítsd Horace-t, hogy össze tudja gyűjteni a karácsonyi ajándékokat. Használj billentyűzetet (Q, E, T, U, O - balra, W, R, Y, I, P - jobbra, A..Enter - tűz) vagy Kempston joystickot Horace mozgatásához. Közben próbáld elkerülni a szokásos ellenségeket és a hógolyókat, hópelyheket, és vigyázz, hogy ne zuhanj be a lyukakba. Lehetőséged van az ellenségek leütésére is, ha a megfelelő módon rájuk ugrasz.



A játék 49 egyre nehezedő szintet tartalmaz, öt különböző helyszínen. Először a vonaton, majd a háztetőkön, aztán a tehervonaton, végül a fagyos, majd a teljesen fagyott talajon kell teljesítened a szinteket, melyek mindegyike maximum 60 időegységig tart, és mindegyiken 10 ajándékot kell összeszedned. A még összegyűjtendő ajándékok számát és a hátralévő időt a jobb felső sarokban láthatod. A nehezebb szinteken találkozni fogsz az elfekkel, akik a képernyő közepére érve felrobbannak, négy hatalmas követ repítve két oldalra. A kövek elől egyedül a képernyő közepére menekülhetsz. Szerencsédre az ajándékok felvételéhez hasonlóan a Hungry Horace-ból megismert csengőket is felveheted, melyek különböző érdekes bónuszokat (pont, élet), vagy hasznos/haszontalan szolgáltatásokat (ellenségek kiiktatása, sérthetlenség, jeges felület) nyújthatnak.

Mezei Róbert (m/zx)

A játék innen letölthető:

<http://www.stevebroad.co.uk/>



PÓKOK A JÁTÉKHOZ

99 másodperces szintek:

POKE 35155, 57: POKE 35160, 57

Csak egy csomag kell a következő pályához:

POKE 35165, 48: POKE 35169, 60

Kezdő pálya: POKE 36695, x ahol x 1-49

245 élet: POKE 35150, 255

Örökéletet: POKE 38150, 0

Bónusznak egy örökidő: POKE 36052, 0

Gaál Csaba

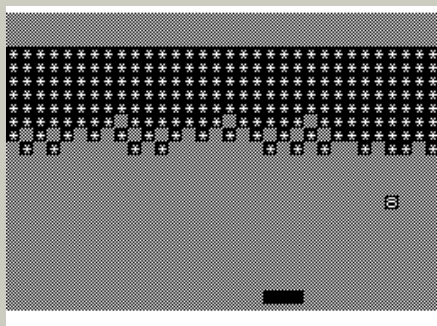
ZX81 KLASSZIKUSOK - 3. RÉSZ

Ebben a számban is szeretnénk folytatni a sort a fontosabb ZX81 opuszok áttekintésével. Gyanítom tartogathat meglepetést is ez a kis sorozat, de ha másra nem is, hát egy kis nosztalgiazásra remek lesz. Ebben a részben egy rendhagyó írást olvashattok Kiki billentyűi alól. (Balee)

BREAKOUT (1982 - CDS MICRO SYSTEMS)

A számítógépek elterjedése óta alapjátéknak számít az úgynevezett "falbontó" játék. Minden géptípusra megírták már tucatszor, manapság egy jobb WC-papír adagolóknak is szerves részét képezi valamelyik változata. Annak idején persze kihívásnak számíthatott a szűkös erőforrásokkal rendelkező, korabeli hardverekre egy élvezhető verzió elkészítése. Ezt a kihívást akceptálta a CDS Micro Systems nevű cég(?). (Annak idején minden Jóskapista nevű programozó hangzatos nevet választott magának, és egy jó névválasztás félig már garantálta a sikert (lásd Microsoft, hű de gonosz vagyok :)). Bár ez eddigi életem első általam látott ZX81-es gépre készült játéka, azt mondanám, inkább ne tette volna. A játék alapkonceptiója, hogy a képernyő felső részében elhelyezkedő téglákat az utolsó darabig eltüntessük. Ehhez

rendelkezésünkre áll egy, a képernyő alsó részben balra-jobbra mozgatható ütő, és a természet mesés rendjének következtében a képernyő bármely részében fellelhető labda. A labda ha téglát ér, az eltűnik és örülünk. Ha falat vagy ütőt ér, akkor visszapattan (szintén örülünk), ha alul elhagyja a játékteret akkor... Na, mi van akkor? Nos, senki? Ámátörök... Ilyen esetben elveszítünk egy életet (és kevésbé örülünk).



Betöltés után kiválaszthatjuk a nehézségi szintet 1-9-ig, ami a labda sebességét fogja befolyásolni. Az ütő irányítása az 1 és 0 billentyűkkel történik. Kezdszkor 9 életünk van és bazi sok téglánk. De tényleg. Egy nagyobb családi házat fel lehetne húzni belőle és még a kerítésre is maradna. Minden téglá belsejében egy csillag található. Hogy minek? Csak. Nyilván a programozónak volt raktáron egy marék

megmaradt hópehely karaktere a cég korábbi játékból, a White Snowing Simulator - közismertebb nevén WSS-ből, és gondolta ide jó lesz. A játékban minden szögletes. A labda is. Nem tudom, mekkora programozói bravúr lett volna egy O betűt mozgatni egy négyzet helyett, de legalább tökéletesen illik a labdapattanás matematikai modelljéhez.

Olyan ugyanis nincs. A labda mindig mindenhol 45 fokban pattan, és ez egy kissé kiszámíthatóvá teszi a játékot. Apró ötlet (de nem túl szerencsés), hogy életeink számát a labda belsejében láthatjuk. A labda és az ütő scrollozása karakter szintű, de gondolom ez hardveres limitáció lehet a ZX81 részéről. A küzdelem pontokért megy (minden téglá 1 pontot ér, megszámláltam), de ez csak a játék végén derül ki, mert játék közben nem kapunk információt arról, hogy megdöntöttük-e már a szomszéd Pistike rekordját. Van "High score" is. Ezt a program egy NAND memória modulban tárolja a ZX81 rejtett áramkörei között az idők végezetéig. Lehet, hogy túl magas elvárásaim voltak a programmal vagy a ZX81-el kapcsolatban, de azt hiszem, ha ezzel a játékkal találkoztam volna először, mikor még ismerkedtem a számítógépek világával, akkor inkább kőművesnek vagy pléhbojnyuszinak álltam volna annak idején, esetleg papnak (de semmiképp nem Gyurinak). Nem ismerem a ZX81 képességeit, de tekintettel arra, hogy a játék 16K-s gépre készült, kis törődéssel szerintem sokkal többet ki lehetett volna hozni belőle (többféle téglá, lehulló extrák, látható pontozás, stb.).

Sárfi Krisztián (Kiki)

Folytatjuk...

PROGRAMOZÁSTECHNIKA - BASIC

ZX SPECTRUM TV-BASIC KÜLÖNKIADÁS - 3. RÉSZ

Kedves Programozni Vágyó Olvasó!

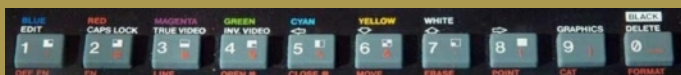


A soron következő részben a Spectrum BASIC "egyszerű" grafikai részébe bonyolódunk bele, amely távolról sem könnyű és közben alaposan elgondolkodtató.

Rajzolás karakterekkel

Sokan életükben először talán az első drágább színes televíziókon adásidőben nézhető képűjságban találtak szögletes jelekkel rajzolt egyszerű grafikákkal és bizony meg is irigyelhatték vagy ha más nem, maguk is szerettek volna olyat készíteni.

Lessük csak meg a billentyűzeten szereplő szép grafikus jeleket, amelyekkel oly romantikusan lehet labirintusokat rajzolni!



(ott, a számok mellett látható négyzetecskék azok)

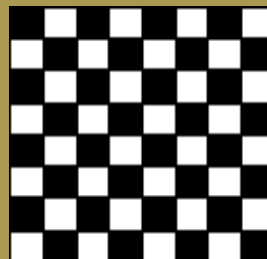
Nem túl kusza az elrendezésük? Kezdő programozó legyen, aki eligazodik köztük és rájön, hogy van inverz módjuk is, amelyhez a SHIFT-et kell nyomkodni!

Vajon miért nem követték egymást az összetartozó elemek? Nem lenne szebb a sorrend például így:



Majd meglepődve tapasztalhatjuk programíráskor, hogy a 8-ason szereplő grafikus karakter inverz rajzolata és a szóköz kinézetre egyforma (egy üres karakter),

de belsőre nagyon más, azaz karaktereik kódja eltérő ... miért nem kaphatott inkább kis ponthálót helyette?



Vagy nem jutott eszükbe vagy ha a programozónak nem tetszett valamiért, akkor a válasz egyszerű:

Mert erre találták ki a szabadon definiálható karaktereket, melyet röviden UDG-nek (User Defined Graphics) szólíthatunk!

A 17 darab, betű szerint A-tól U-ig szabadon definiálható karakter rendelkezésünkre áll, bár elég zavaróan alaphelyzetben a normál betűk képét tartalmazzák, már ha valaki GRAPHICS módba át tud váltani (Caps Shift és 9) és ott nyomkodja a betűket.

Miért nem adták oda azt a pár kimaradt betűt V-től Z-ig?

Mindössze 72 bájt memóriára lett volna szükség ehhez, ami lássuk be, nem a világ vége, miután még utána is több, mint 41000 bájt állt volna rendelkezésre egy BASIC programnak.

Aztán ha már a ROM-ban üresen maradt 1939 bájtnyi terület, akkor miért nem leptek meg minket pár aranyos, kedves vagy épp hasznos kis alakzattal, amely azonnal betöltésre került volna?

Egy emberke, golyó, téglasor vagy négyzet nem lett volna a világ vége...



De hogy az UDG karakterek programozása miért lett ennyire "egyszerű"?!?

Lássuk példának az emberkét:

```
POKE USR "A",24      (vagy BIN 00011000)
POKE USR "A"+1,24   (vagy BIN 00011000)
POKE USR "A"+2,16   (vagy BIN 00010000)
POKE USR "A"+3,126  (vagy BIN 01111110)
POKE USR "A"+4,24   (vagy BIN 00010000)
POKE USR "A"+5,40   (vagy BIN 00101000)
POKE USR "A"+6,68   (vagy BIN 01000100)
POKE USR "A"+7,68   (vagy BIN 01000100)
```

Így tehát majdnem úgy, mint a régi mesében ... adtak is, meg nem is, benne is volt, meg nem is, egyszerű is, meg nem is.

A karakterkészlet

Minden kis- és nagybetű, írásjel és egyéb karakter képe - egyezően az UDG definiálásával - tárolva van. 768 bájtot foglal el, mivel 96 különböző karakter létezik és minden karakter 8 bájtot igényel.

Bár a programozók szépen előkészítették a betűstílus cseréjének lehetőségét, valamiért a BASIC felhasználók számára nem nyitották meg ennek egyszerű csatornáját, holmi "pókolgatásra" kényszerítve minket. Ne maradjunk teljesen elzárva a karakterkészlet váltásától, a

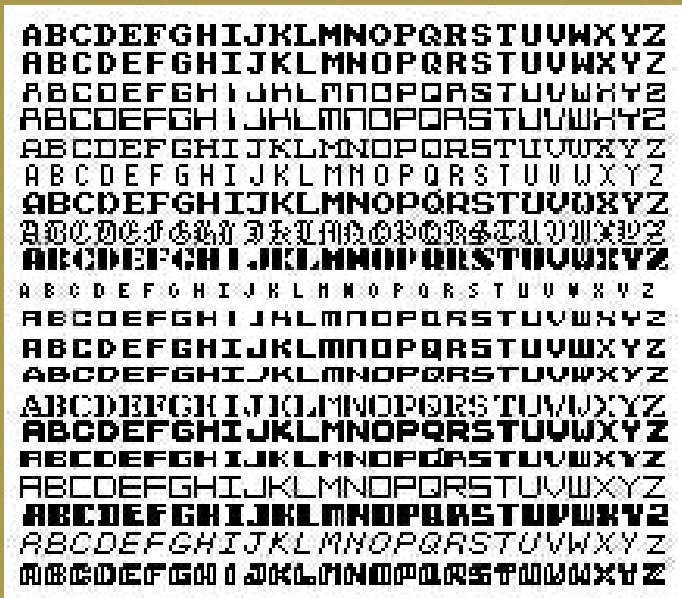
```
POKE 23606,X: POKE 23607,Y
```

utasításokkal megtehetjük, ahol "X" és "Y" a karakterkészlet címét adja meg egy kezdők számára nehezen emészthető képlettel, miszerint:

Jelölje C a karakterkészlet kezdő címét a memóriában és ekkor

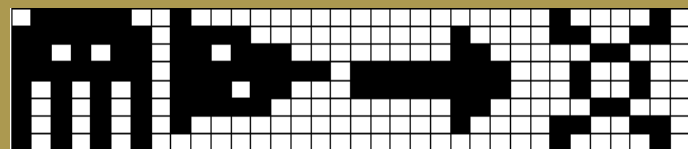
```
Y=INT (C/256) -1 és X=C-256*(Y+1)
```

Vajon miért nem tettek a ROM-ba (van hely bőven) még egy szép karakterkészletet, amelyet akár egy utasítással tettek volna elérhetővé, például GRAPHICS 0 alapkészlet, GRAPHICS 1 másik készlet.



(Kinálat bőven volt)

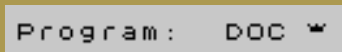
Arra a sok szabad helyre akár kettő is befért volna, akkor pedig GRAPHICS 2-re jöhetett volna 96 kedves, hasznos karakter a nyilaktól kezdve az úrhajón át a szörnyekig minden. Reklánilag igencsak lenyűgöző lett volna a nézelődők, ismerkedők számára.



Sok-sok nyitott kérdés, amelyekre választ nem lelhetünk.

Játék a betűkkel, azaz betűt a betűre!

Néhány játékprogramnál rácsodálkozhatunk, hogy milyen érdekes neve van. Töltsük be egyik kedvencünket, a Defender of the Crown-t:



és varázslóként tekinthetünk készítőikre, hiszen korona nincs is Spectrumon!

Igaz, ami igaz, ilyen betű nincs, de egy kis finom trükkel lehet:

```
PRINT "Y";CHR$ 8; OVER 1;"↑"
```

és kész is a kis korona.

Nem történik más, mint kiíródik egy nagy Y, utána visszalépünk rá (CHR\$ 8) és úgy írjuk rá a felnyilat, hogy összekombinálódjon vele (OVER 1).

Ezzel a kis programmal legenerálhatjuk az összes ilyen rajzocskát és néha bizony érdekes, sőt már-már rejtélyes dolgok jönnek elő.

```
10 FOR f=33 TO 143
20 CLS
30 FOR g=f+1 TO 143
40 PRINT CHR$ f;CHR$ 8; OVER
1;CHR$ g; OVER 0;" ";
50 NEXT g
60 PAUSE 0
70 NEXT f
```

Így hát nincs más teendőnk, mint hogy keresgélünk és ha megtaláltuk a nekünk szükséges rajzocskát, már el is kápráztathatjuk vele spectrumos barátainkat.

Pontok, vonalak, körök

Bármilyen hitetlenkedve is nézzük, bizony ott van a billentyűzeten a PLOT, DRAW és CIRCLE (pont-, vonal- és körrajzoló) utasítás, amelyeket C64-es barátaink igencsak irigyelnek, mert Spectrumon "kényelmesen" lehet rajzolni. Nincs külön karakteres és grafikus képernyő, nem kell POKE utasítások tömegét ismerni egy sima vonalhoz és amint sikeresen kiadjuk első rajzoló utasításunk, azonnal gyönyörködhetünk is annak szépségében.

Először idézzük vissza sorozatunk első részében írtakat:

```
PLOT, DRAW, CIRCLE utasítás mind csodaszép dolgokra képesek, de mit vétettünk nekik mi, BASIC rajongók, hogy csak 176 soron rajzolhatunk, pedig 192 sornyi grafika állt rendelkezésre és bizony semmilyen korlátja nem lett volna megengedni ezt a luxust számunkra.
```



Arra az alsó fehér részre nem lehet rajzolni, pedig semmi akadálya nem lett volna maga a Spectrum képes rá, de a BASIC-je nem. A ROM-ban található grafikai résszel is menne a teljes képernyő használata, de szándékosan lett csökkentve.

A PLOT utasításban nem rejlik semmi varázslat, bár a koordináták megadása eleinte meg-megtréfálja a kezdőket, mert ...

... egyrészt a koordináta rendszer nem a PRINT utasításnál megszokott bal felső sarokból indul, hanem az alsóból (ez végülis nem probléma, mert a matematikai koordináta rendszer is így néz ki, de nehézséget okoz, ha egy vonal végéhez egy szöveget szeretnénk írni, mert nem kis munka kiszámolni a megfelelő karakteres koordinátákat);

... másrészt semmi nem segít abban, hogy vízszintes, függőleges vagy függőleges, vízszintes a helyes koordináta sorrend (ez kis idő múlva már nem jelent problémát).

A DRAW utasítás már izgalmasabb, mert egyenesen kívül a korábbi, karácsonyfás képen is látható görbék, ívek húzására is képes! Persze ehhez kisebb matematikai zseninek kell lenni, hiszen az ív irányának és "hasának" megadása PI segítségével történik.

Kissé érhetetlen, hogy ha:

egy negyed körívhez $\text{PI}/2$ -öt írhatunk,
egy fél körívhez PI -t,
egy háromnegyed körívhez $3*\text{PI}/2$ -öt,

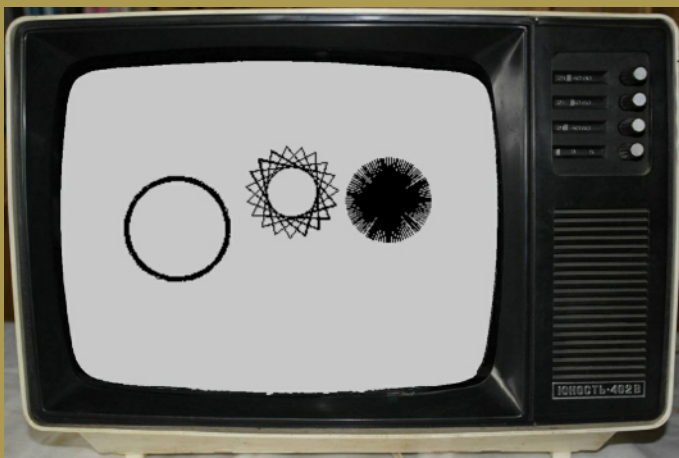
akkor a teljes körhöz miért nem írhatjuk oda a $2*\text{PI}$ -t (odairhatjuk,

de kör helyett egy egyenes vonalat kapunk), ám nem érdemes feszegetni, hiszen kör rajzolásához adták nekünk a CIRCLE utasítást.

Ezzel ugyan kényelmesebb rajzolni, hiszen a középpontot és a sugarat kell megadni, de ha a DRAW tudta volna a kört, akkor teljesen feleslegessé vált volna a CIRCLE és helyére más, hasznos utasítás kerülhetett volna. Egyben ne is beszéljünk arról, hogy a kör rajzolása rendkívül lassú, mert minden pont helyét szögvényekkel számolja ki ahelyett, hogy okosan csak negyedívet számolt volna és azt tükrözte volna 4 irányba a középpontra (amint azt későbbi BASIC bővítések alkalmazták).

Az igazi programozónak eszébe se jut értelmetlen adatokat megadni a rajzoló utasításokban, mert úgy gondolja, hibaüzenetet kap, de a DRAW utasítás meglepheti. Próbáljunk az ív megadásához $2 \cdot \text{PI}$ -nél nagyobb értékeket írni és váratlan csodákban lesz részünk!

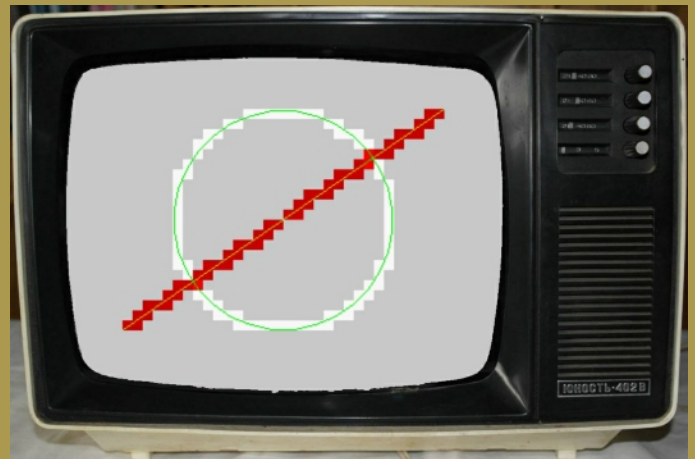
Ki gondolná, hogy az alábbi rajzokhoz mindössze 3 darab DRAW utasítás (előtte 3 darab PLOT) elegendő (meg egy kis türelem)!



Ó, de a színek! Ha már rajzolunk, használhatunk színeket is?

Megint csak "igen is, meg nem is" a megfelelő válasz. Hogy miért? Mert a ZX Spectrum színei már nem annyira szabadok a képernyőn, mint pontocskái.

Tegyük a kiinduló pontot a bal alsó sarokba - `PLOT 0,0` - majd húzzunk egy piros alapon sárga vonalat a képernyőn átlósan - `DRAW PAPER 2; INK 6; 255,175` - majd egy fényes zöld körrel zárjuk művünket a képernyő közepén - `CIRCLE BRIGHT 1; INK 4; 127,87,87` - és csodálkozzunk egy nagyot:



Hát ez bizony már nem lett olyan szép. Sajnos a rajzolást csak akkor élvezhetjük igazán, ha a háttér színét nem változtatjuk és a fényességen sem állítunk. A képen látható "négyzetesedést" az okozza, hogy egy 8×8 pontból álló területen belül csak 2 szín alkalmazható (háttér és tinta), így ha oda egyetlen egy pontot is kiteszünk, akkor az egész terület átveszi annak szín adatait (végülis milyen butaság kitenni egy pontot, amelynek a háttér színe eltérő, hiszen egy PONTOT akartunk kitenni, annak meg tinta színe van).

Ha már színekre léptünk, soroljuk fel, milyen utasítások segítenek benne minket:

BORDER Beállítja a keret színét. Kezdként meglepődve tapasztalhatjuk, hogy "csak" 8 színünk van és nem használható a fényesség állítása (lásd később).

PAPER A hasznos terület háttérének színét változtathatjuk meg. Itt már igazán nem 8, hanem 15 szín áll rendelkezésre, mert a fényesség szinte új

színeket varázsol. Ne a tévéken keressük a hibát, de van fekete és fényes fekete, ám látható különbség nincs köztük pedig programunk számára nem egyformák!

INK A kiírások, pontok színét jelölhetjük ki. Sajnos a fényesség csak a háttér színével együtt értelmezett, így a kombinációk száma meglehetősen véges.

BRIGHT Fényesség ki- és bekapcsolása.

FLASH Villogás ki- és bekapcsolása.

Sokat gondolkozhatott a tervező annak idején a villogáson és valószínűleg a műszakiaknak is sok fejtörést okozott megvalósítása, nem beszélve a tévé "kínlódásáról", ahogy megpróbálta követni.

Elsőre nagyon hasznosnak tűnhet egy villogó kiírás - villogás alatt értjük, hogy másodpercenként felcserélődik a szín az adott területen - de értelmes alkalmazása szinte nincs és nem is volt. Sokkal többre mentünk volna a szembántó villogás helyett akár egy újabb fényességi szinttel, így növelve az alkalmazható színek számát - aki villogást akart volna, az tudott volna csinálni programjával is, de ne legyünk telhetetlenek, örüljünk és kerüljük alkalmazását.

Azt pedig már csak az igazi programozók próbálják ki, hogy fekete háttérre villogva fényes feketével kiírják a "FOLYTATJUK" feliratot!

```
10 PRINT PAPER 0; INK 0;  
BRIGHT 1; FLASH 1;  
"FOLYTATJUK!": GO TO 10
```

Pgyuri

IDÉZET

Bruce Everiss elég tömören megfogalmazta, hogy mennyire nem Sir Clive szája íze szerint lett sikeres a ZX Spectrum...

Forrás:

<https://www.wired.com/2010/06/sinclair-zx80/>



"The games industry was being dragged along on the back of the Sinclair Spectrum, which was a thousand times more successful than Sinclair expected it to be," said Bruce Everiss, founder of Microdigital, one of the first computer stores in Europe. "He thought people would be cataloguing their stamp collections on the back of it. The fact that the Spectrum became 99 percent used for game playing took him by surprise."

azaz

"A játékipar nagy sikerrel lovagolta meg a Sinclair Spectrumot, amely így ezerszer sikeresebb lett, mint amire Sinclair eredetileg számított." - nyilatkozta Bruce Everiss, a Microdigital alapítója, amely egyike volt a legelső számítástechnikai üzleteknek Európában. "Úgy gondolta, az emberek a gép sikerének eredményeképp kezdik majd katalogizálni és rakosgatni a bélyeggyűjteményüket. A tény, hogy a Spectrumot 99%-ban játéokra használták, nagy meglepetés volt számára."

m/zx, Kapitány

PROGRAMOZÁSTECHNIKA - ASSEMBLY OVI

HOGYAN ÍRJUNK JÁTÉKOT ZX SPECTRUMRA - 6. RÉSZ

Táblázatok

A földönkívüli nem jár egyedül!

Tegyük fel az egyszerűség kedvéért, hogy egy Space Invaders játékot írunk, amelyben 11 oszlopos sorokban támadnak az idegenek, összesen 5 sorban. Nem lenne praktikus egyenként külön megírni mind az 55 támadóhoz tartozó kódot, ezért inkább táblázatot fogunk készíteni belőlük! A Sinclair BASIC-ben ehhez definiálhatunk három tömböt, mindegyiket 55 elemmel: egy tömböt a támadók x koordinátája számára, egyet az y koordinátáknak, valamint egy harmadikat a státusz bájtok tárolásához. Hasonló megoldással élhetünk, ha assemblerben kijelölünk három, egyenként 55 bájt hosszúságú területet a memóriában, majd egy támadó értékeinek eléréséhez csupán hozzá kell adnunk a támadó sorszámát a három táblázat kezdőcíméhez. Sajnálatos módon ez a megközelítés meglehetősen lassú és körülményes lenne.

Sokkal jobb módszer, ha az egy támadóhoz tartozó mindhárom értéket egy struktúrába csoportosítjuk, és ebből az 55 struktúrából készítünk egy táblázatot. Így beállítva **HL** értékét bármelyik támadó kezdőcímére, tudjuk, hogy **HL** ekkor a státuszbájtra mutat, **HL+1** az x koordináta értéke, míg **HL+2** az y koordinátáé. Az egyes támadókat megjelenítő kód valahogy így festene:

```
ld hl, aliens ; támadó adatszerkezete
ld b, 55 ; támadók száma
loop0:
call show ; megjelenítjük ezt a
támadót
djnz loop0 ; megismételjük minden
támadóval
ret
show:
ld a, (hl) ; kiolvassuk a támadó
státuszát
cp 255 ; ki van kapcsolva?
jr z, next ; ha igen, nem jelenítjük
meg
push hl ; támadó címe a veremb.
inc hl ; x koordinátára állás
ld d, (hl) ; koordináta kiolvasása
inc hl ; y-ra állás
ld e, (hl) ; koordináta kiolvasása
call disply ; megjelenítjük a támadót a
(d,e) pozícióban
pop hl ; visszaolvassuk a támadó
címét a veremből
next:
ld de, 3 ; támadó adatszerkezet
mérete
add hl, de ; következő támadóra ugrás
ret ; hl a következő támadóra
mutat
```

Az index regiszterek használata

A hátulütője ennek a rutinnak, hogy mindvégig rendkívül kell ügyelnünk arra, hogy **HL** hova is mutat éppen. Kézenfekvő ötlet lenne eltárolni **HL** értékét egy kétbájtos ideiglenes memóriacímen, mielőtt meghívnánk a **show** rutint, majd később visszaállítani azt, hozzáadni hármat az értékéhez a fő ciklus végén, majd ezután elvégezni a **djnz** utasítást. Amennyiben Nintendo GameBoy-ra készítenénk játékot a maga lebutított Z80 processzorával, valószínűleg ez jelentené a legjobb megoldást számunkra. Ellenben fejlettebb processzorral rendelkező gépeken – mint amilyen a Spectrum is, vagy a CPC464 -, használhatjuk az **IX** indexregisztereket, hogy némileg leegyszerűsítsük a kódot. Mivel az **IX** regiszterpár lehetővé teszi, hogy elmozdítsuk az indirekt címzésünket, nyugodtan ráállíthatjuk **IX**-et a támadóhoz tartozó adatstruktúra elejére, így is el tudjuk érni bármelyik elemet az adatszerkezetben anélkül, hogy **IX** értékét meg kellene változtatnunk újra. **IX** használatával így nézne ki a támadókat megjelenítő rutinunk:

```
ld ix, aliens ; támadó adatszerkezete
ld b, 55 ; támadók száma
loop0:
call show ; megjelenítjük ezt a
támadót
ld de, 3 ; támadó adatszerkezet
mérete
add ix, de ; következő támadóra ugrás
djnz loop0 ; megismételjük minden
támadóval
ret
show:
ld a, (ix) ; kiolvassuk a támadó
státuszát
cp 255 ; ki van kapcsolva?
ret z ; ha igen, nem jelenítjük
meg
ld d, (ix+1) ; x koord. olvasása
ld e, (ix+2) ; y koord. olvasása
jp disply ; megjelenítjük a támadót a
(d,e) pozícióban
```

IX-et tehát elegendő mindig csak az egyes támadók adatszerkezetének a legelejére állítanunk, és **IX** mindig az aktuális támadó státuszára fog mutatni, **IX+1** az x koordinátájára, és így tovább. Ez a módszer lehetővé teszi összetett, akár 128 bájt hosszúságú adatszerkezetek használatát minden egyes támadóhoz. Így annak a lehetőségét is kizárjuk, hogy belezavarodjunk, éppen melyik pontjára mutatnak az adatszerkezetnek a regisztereink egy adott pillanatban, ahogyan azt a **HL** regisztereket használó példában korábban láthattuk. Sajnálatos módon, az **IX** regiszterpár

használata némiképp lassabb a HL-nél, ezért érdemes kerülni a használatát az intenzívebb számítási feladatoknál, például grafikák esetében. (Az ilyen jellegű adatszerkezésre azért van szükség, mert egy adathalmaz egy példányát leíró adatszerkezet elemeinek a száma általában kevesebb, mint az adathalmaz példányainak a száma, így az IX, IY regiszterpárt és egy ofszetet használó 'ld' utasítás egyetlen művelettel eléri a kívánt elemét bármelyik példányban, így rövidebb kódot eredményez. Ennek feltétele, hogy az adatszerkezet 129 elemnél kevesebből álljon, ugyanis $(IX+n)$ és $(IY+n)$ esetében $-128 \leq n \leq 127$. Más regisztereket használva nincsen lehetőségünk eltolás megadására címzés esetén – a fordító megjegyzése).

Most alkalmazzuk ezt a módszert a százlábú játékunkban is! Először is el kell döntenünk, hogy hány szegmensre lesz szükségünk, és milyen adatot tároljunk az egyes szegmensekről! A mi játékunk esetében a szegmenseknek balra vagy jobbra kell tudniuk mozogni, amíg gombához nem érnek, majd lefelé kell mozogniuk, ezután vissza a másik irányba. Úgy látszik, szükségünk lesz egy jelzőre, ami azt jelöli, milyen irányba mozog az adott szegmens, valamint a szegmens x és y koordinátájának a tárolására is. A jelzőnk azt is tartalmazhatja, hogy az adott szegmens már megsemmisült, vagy még nem. Ez alapján hozunk létre egy hárombájtos adatszerkezetet:

```
centf defb 0 ; jelző, 0=balra, 1=jobbra,
          255=halott
centx defb 0 ; szegmens x koordinátája
centy defb 0 ; szegmens y koordinátája
```

Amennyiben 10 szegmens hosszúságúnak választjuk a százlábunk méretét, ennek megfelelően egy 30 bájtos hosszúságú táblázatnak kell helyet foglalnunk a memóriában. Minden szegmensen inicializálnunk kell az induláskor, majd törölhetjük, mozgathatjuk, újra megjeleníthetjük a játék folyamán.

A szegmensek kezdőértékeinek a beállítása nem túl bonyolult feladat, megteszi egy egyszerű ciklus is, amely a HL regiszterpár értékét növeli mielőtt beállítjuk az egyes bájtokat. Ez valahogy így nézne ki:

```
ld b, 10 ; inicializálandó szegmensek
          száma
ld hl, segmnt ; a szegmens tábla
segint:
ld (hl), 1 ; kezdés keppen jobbra mozog
inc hl
ld (hl), 0 ; fent indul
inc hl
ld (hl), b ; B regiszter mint y
          koordináta
inc hl
djnz segint ; megismételjük az összes
          szegmensre
```

Feldolgozni és megjeleníteni a szegmenseket egy kicsit bonyolultabb lesz, úgyhogy ehhez már az IX regisztereket hívjuk segítségül. Egy olyan egyszerű algoritmust kell elkészítenünk, ami egyetlen szegmensen mozgat balra vagy jobbra, amíg az el nem ér egy gombát, majd mozgatja lefelé, és végül irányt vált. Nevezzük ezt a rutint *prosegnek* („process segment”)! Ezután ciklust szervezünk, ami egyenként kiválasztja a szegmenseket, és meghívja a *proseg* eljárást. Amennyiben valóban helyes a mozgó algoritmusunk, már láthatunk is egy százlábút amint a gombok sorain kígyózik végig! A rutin alkalmazása a jelenlegi kódunkban nem fog túl sok nehézséget okozni – megvizsgáljuk a jelzőbájtot minden szegmens esetében (IX), hogy kiolvassuk a mozgás irányát. Ennek megfelelően növeljük vagy csökkentjük a vízszintes koordináta értékét (IX+2), majd leellenőrizzük az ezen a helyen található karaktercella attribútum értékét. Amennyiben azt találjuk, hogy az zöld-fekete: növeljük a függőleges koordinátát (IX+1), és átállítjuk a haladási irány jelzőt (IX).

Rendben, valóban van még néhány eshetőség, amit meg kell vizsgálnunk: például mi a helyzet, ha elértük a játéktér valamelyik szélét? Ez az eset sem bonyolultabb: itt is pusztán a szegmensek koordinátaértékeit kell megvizsgálnunk és ha szükséges, irányt kell változtatnunk vagy a képernyő tetejére kell ugrani. Továbbá a szegmenseket el kell tüntetnünk az eredeti helyükről, mielőtt átmozgatnánk őket és újra megjelenítenénk, ám az ehhez a művelethez szükséges lépések már nem jelentenek újdonságot.

A frissített kódunk így néz ki:

```
; Példa 6.1
; Fekete képernyőt szeretnénk

ld a, 71 ; fehér tinta (7) fekete
          papíron (0),
          ; világosan (64)
ld (23693), a ; képernyő színek beállítása
xor a ; akkumulátor nullázásának
          gyors módja
call 8859 ; maradandó keretszín

; Grafika beállítása

ld hl, blocks ; felhasználói grafikus
          elemek címe
ld (23675), hl ; az UDG ide mutasson

; Rendben, kezdődjön a játék!

call 3503 ; ROM rutin - képernyő
          törlése, 2-es csatorna
          nyitása
```



```

; Koordináták inicializálása
ld hl, 21+15*256
; kezdeti koordináták a HL-be
ld (plx), hl; játékos koordinátái
call basexy ; x és y pozíciójának
; beállítása
call splayr ; játékos törzsének
; megjelenítése

; Ez a főhurok
mloop equ $

; Játékos törlése
call basexy ; x és y pozíciójának
; beállítása
call wspace ; üres hely a játékos
; pozíciójába

; Törölve van a játékos, átmozgathatjuk az új
; pozícióba mielőtt újra megjelenítjük.
ld bc, 63486; billentyűk 1-5/joystick port
; 2
in a, (c) ; kiolvassuk a megnyomott
; gombokat
rra ; legkülső bit = 1-es gomb
push af ; megjegyezzük
call nc, mpl ; ha megnyomva, mozgás balra
pop af ; akku helyreállítása
rra ; következő bit (2-es
; helyiérték) = 2-es gomb
push af ; megjegyezzük
call nc, mpr ; ha megnyomva, mozgás jobbra
pop af ; akku helyreállítása
rra ; következő bit (4-es
; helyiérték) = 3-es gomb
push af ; megjegyezzük
call nc, mpd ; ha megnyomva, mozgás lefelé
pop af ; akku helyreállítása
rra ; következő bit (8-es
; helyiérték) = 4-es gomb
call nc, mpu ; ha megnyomva, mozgás felfelé

; Az átmozgatás megtörtént, újra
; megjeleníthetjük a játékost.
call basexy ; x és y koordináta beállítása
call splayr ; játékos megjelenítése

; Folytatjuk a százlábúval
ld ix, segmnt; szegmens adattábla címe
ld b, 10 ; szegmensek száma a
; táblázatban
censeg:
push bc
ld a, (ix) ; a szegmens be van kapcsolva?
inc a ; 255=kikapcsolt, 0-ra
; inkrementálódik
call nz, proseg; ha aktív, feldolgozzuk a
; szegmenst
pop bc
ld de, 3 ; 3 bájtt szegmensenként
add ix, de ; ix regiszter a következő
; szegmensre
djnz censeg ; megismételjük minden
; szegmensre
halt ; várakozás

; Visszaugrás a főhurok elejére
jp mloop

; Játékos balra mozgatása
mpl:
ld hl, ply ; Emlékezzünk, y a vízszintes
; koordináta!
ld a, (hl) ; Mi a mostani érték?
and a ; Nulla?
ret z ; Ha igen, nem tudunk tovább
; balra menni!

; leellenőrizzük nincs-e gomba az útban
ld bc, (plx); aktuális koord
dec b ; 1 négyzettel balra nézünk
call atadd ; annak a pozíciónak a címe
cp 68 ; a gombok világosak (64) +
; zöldek (4)
ret z ; ha gombát találunk, nem
; mehetünk arra
dec (hl) ; különben y = y-1
ret

; Játékos jobbra mozgatása
mpr:
ld hl, ply ; Emlékezzünk, y a vízszintes
; koordináta!
ld a, (hl) ; Mi a mostani érték?
cp 31 ; A jobb szélén vagyunk (31)?
ret z ; Ha igen, nem tudunk tovább
; jobbra menni!

; leellenőrizzük nincs-e gomba az útban
ld bc, (plx); aktuális koordináta
inc b ; 1 négyzettel jobbra nézünk
call atadd ; annak a pozíciónak a címe
cp 68 ; a gombok világosak (64) +
; zöldek (4)
ret z ; ha gombát találunk, nem
; mehetünk arra
inc (hl) ; különben y = y+1
ret

; Játékos felfelé mozgatása
mpu:
ld hl, plx ; Emlékezzünk, x a függőleges
; koordináta!
ld a, (hl) ; Mi a mostani érték?
cp 4 ; a pálya tetején vagyunk (4)?
ret z ; Ha igen, nem tudunk tovább
; felfele menni!

; leellenőrizzük nincs-e gomba az útban
ld bc, (plx); aktuális koordináta
dec c ; 1 négyzettel fentebb nézünk
call atadd ; annak a pozíciónak a címe
cp 68 ; a gombok világosak (64) +
; zöldek (4)
ret z ; ha gombát találunk, nem
; mehetünk arra
dec (hl) ; különben x = x-1
ret

```

```

; Játékos lefelé mozgatója
mpd:
  ld hl, plx ; Emlékezzünk, x a függőleges
              koordináta!
  ld a, (hl) ; Mi a mostani érték?
  cp 21      ; A képernyő alján vagyunk
              (21)?
  ret z      ; Ha igen, nem tudunk tovább
              lefele menni!

; leellenőrizzük nincs-e gomba az útban
  ld bc, (plx); aktuális koordináta
  inc c      ; 1 négyzettel lentebb nézünk
  call atadd ; annak a pozíciónak a címe
  cp 68      ; a gombok világosak (64) +
              zöldek (4)
  ret z      ; ha gombát találunk, nem
              mehetünk arra

  inc (hl)   ; különben x = x+1
  ret

; A játékos törzsének, x és y koordináta
; értékének beállítása, ez a rutin kerül
; meghívásra a törzs törlése és megjelenítése
; előtt.
basexy:
  ld a, 22   ; AT pozicionáló kód
  rst 16
  ld a, (plx) ; játékos függőleges
              koordináta
  rst 16     ; beállítjuk
  ld a, (ply) ; játékos vízszintes
              koordináta
  rst 16     ; ezt is beállítjuk
  ret

; Megjelenítjük a játékost a jelenlegi
; PRINT pozícióban.
splayr:
  ld a, 69   ; cián tinta (5) fekete
              papíron (0),
              ; világosan (64)
  ld (23695), a ; beállítjuk az ideiglenes
              színeket
  ld a, 144  ; 'A' UDG ASCII kódja
  rst 16     ; játékos kirajzolása
  ret

wspace:
  ld a, 71   ; fehér tinta (7) fekete
              papíron (0),
              ; világos (64)
  ld (23695), a ; beállítjuk az ideiglenes
              színeket
  ld a, 32   ; SZÖKÖZ karakter
  rst 16     ; üres hely megjelenítése
  ret

segxy:
  ld a, 22   ; pozíció karakter ASCII kódja
  rst 16     ; pozíció kiírása
  ld a, (ix+1); szegmens x koordinátája
  rst 16     ; koordináta kód kiírása
  ld a, (ix+2); szegmens y koordinátája
  rst 16     ; koordináta kód kiírása
  Ret

proseg:
  ld a, (ix) ; Megvizsgáljuk a szegmens ki
              van-e kapcsolva
  inc a      ; az ütközésészlelő rutinnal
  ret z      ; ha ki volt kapcsolva, ez a
              szegmens már halott
  call segxy ; beállítjuk a szegmens
              koordinátákat
  call wspace ; üres hely kiírása
  call segmov ; szegmens mozgatója
  ld a, (ix) ; Ismét megvizsgáljuk, hogy ki
              van-e kapcsolva
  inc a      ; az érték növelésével
  ret z      ; ha ki van, halott szegmens
  call segxy ; szegmens koordinátái
  ld a, 2     ; 2-es attrib. kód = piros
              szegmens
  ld (23695), a ; ideiglenes attribútumérték
  ld a, 146   ; UDG 'C' a szegmens
              megjelenítéséhez
  rst 16
  ret

segmov:
  ld a, (ix+1); x koordináta
  ld c, a      ; GP x terület
  ld a, (ix+2); y koordináta
  ld b, a      ; GP y terület
  ld a, (ix)   ; státusz jelző
  and a; balra tart a szegmens?
  jr z, segml  ; balra - ugrás a megfelelő
              kódrészhez

; különben a szegmens jobbra tart!
segmr:
  ld a, (ix+2); y koordináta
  cp 31        ; már elértük a képernyő jobb
              szélét?
  jr z, segmd ; igen - szegmens lefelé
              mozgatója
  inc a        ; jobbra nézünk
  ld b, a      ; GP y koordináta
  call atadd   ; attribútumcím megkeresése
  cp 68        ; a gombok világosak (64) +
              zöldek (4)
  jr z, segmd ; ha gomba van jobbra, inkább
              lefelé mozgunk
  inc (ix+2)   ; nincs akadály, mozoghatunk
              jobbra
  ret

; ha a szegmens balra tart!
segml:
  ld a, (ix+2); y koordináta
  and a        ; már elértük a képernyő bal
              szélét?
  jr z, segmd ; igen - szegmens lefelé
              mozgatója
  dec a        ; balra nézünk
  ld b, a      ; GP y koordináta
  call atadd   ; attribútumcím megkeresése
              (dispx,dispy)
  cp 68        ; a gombok világosak (64) +
              zöldek (4)
  jr z, segmd ; ha gomba van balra, inkább
              lefelé mozgunk
  dec (ix+2)   ; nincs akadály, mozoghatunk
              balra
  Ret

```

```

; ha a szegmens lefelé mozog!

segmd:
  ld  a, (ix) ; szegmens iránya
  xor  1      ; visszafordítjuk
  ld  (ix), a ; új irány tárolása
  ld  a, (ix+1); y koordináta
  cp  21      ; már elértük a képernyő alsó
              ; szélét?
  jr  z, segmt ; igen - szegmens a képernyő
              ; tetejére

; Ezen a ponton mindenképpen lefelé mozgunk,
; függetlenül attól, hogy van-e ott gomba,
; ami blokkolhatja a szegmens útját. Bármi áll
; a szegmens útjában az törlésre kerül.

  inc  (ix+1) ; nem értük el a kép alját,
              ; mozgás lefelé
  ret

; szegmens mozgatása a tetőre

segmt:
  xor  a      ; egynértékű ld a,0-val, de 1
              ; bájtot spórolunk
  ld  (ix+1), a; új x koordináta = a képernyő
              ; teteje
  ret

; Egyszerű pszeudo-véletlenszám generátor.
; Egy mutatót léptet a ROM területen
; (a seed-ben tárolva), visszaadva
; a megcímzett bájttartalmát.

```

```

random:
  ld  hl, (seed) ; Mutató
  ld  a, h
  and  31        ; Az első 8 KB-on belül
              ; tartjuk
  ld  h, a
  ld  a, (hl) ; „Véletlen” szám a mutatott
              ; helyről
  inc  hl      ; Mutató léptetése
  ld  (seed), hl
  ret

seed:
  defw 0

```

```

; Kiszámolja a (dispx, dispy) koordinátánál
; lévő karakter attribútum értékének címét.

```

```

atadd:
  ld  a, c      ; függőleges koordináta
  rrca      ; szorzás 32-vel
  rrca      ; Jobb léptetni maradékkal
              ; 3-szor gyorsabb, mint
  rrca      ; balra léptetni 5-ször
  ld  e, a
  and  3
  add  a, 88    ; 88x256=attribútumok
              ; kezdőcíme
  ld  d, a
  ld  a, e
  and  224
  ld  e, a
  ld  a, b      ; vízszintes pozíció
  add  a, e
  ld  e, a      ; de=attribútum címe
  ld  a, (de) ; visszaadjuk az attribútum
              ; értékét az akku-ban
  ret

```

```

plx  defb 0 ; játékos x koordinátája
ply  defb 0 ; játékos y koordinátája

```

```

; UDG grafika

```

```

blocks:
; játékos törzse
  defb 16,16,56,56,124,124,254,254
; gombák
  defb 24,126,255,255,60,60,60,60
; szegmens
  defb 24,126,126,255,255,126,126,24

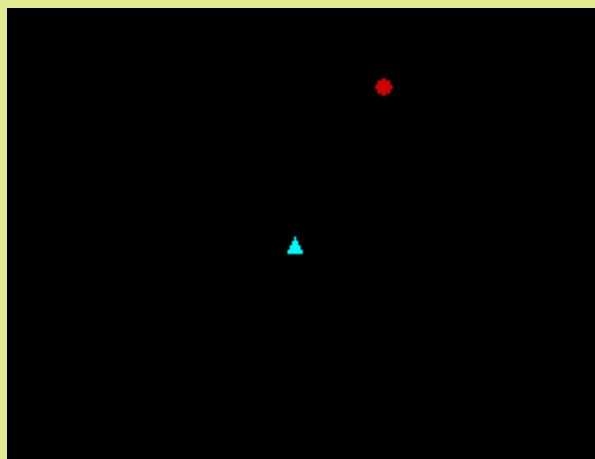
; Szegmens tábla
; Formátum: 3 bájttal szegmensenként, 10 szegmens
; bájttal: 255=szegmens kikapcsolva,
; 0=balra, 1=jobbra
; bájttal 2 = x (függőleges) koordináta
; bájttal 3 = y (vízszintes) koordináta

```

```

segmnt:
  defb 0,0,0 ; szegmens 1
  defb 0,0,0 ; szegmens 2
  defb 0,0,0 ; szegmens 3
  defb 0,0,0 ; szegmens 4
  defb 0,0,0 ; szegmens 5
  defb 0,0,0 ; szegmens 6
  defb 0,0,0 ; szegmens 7
  defb 0,0,0 ; szegmens 8
  defb 0,0,0 ; szegmens 9
  defb 0,0,0 ; szegmens 10

```

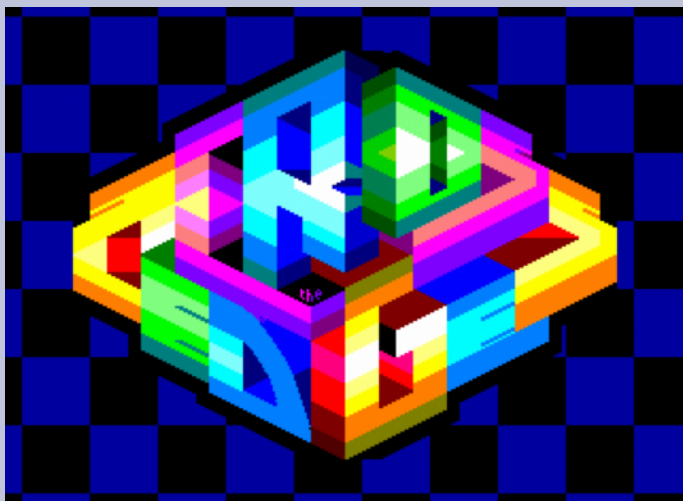


Tanács Imre (Kapitány)

Folytatjuk ...

SCENE ESSZENCIA

ACROSS THE EDGE



Alkotó: deMarche

Kategória: Demó

Party: Chaos Constructions 2016

Helyezés: 1

Vélemény

Ez a Speccy demó visszahozza a '90-es évek hangulatát. Commodore gépeken, főleg Amigán láthattunk ilyen effekteket annak idején. Persze ott sok demóeffekt azért alakult ki, mert hardveből, kevés számítási igénnyel meg lehetett oldani. Spectrumon nem ez a helyzet, itt azért trükközni kell (számítási teljesítmény még kevesebb van). A C=64-es demókban terjedt el a 2000-es évek végétől, hogy azokat az effekteket, amiket 10-15 évvel korábban Amigán kitaláltak, elkészíteték. Figyelembe véve, hogy az A500 7-szer gyorsabb procival bír, nem beszélve a többi chipről, ez szép teljesítmény. No persze azért a grafikai felbontás jóval kisebb a kisgépen.

Ezt a hangulatot hozza el az Across the Edge Speccy-re. Egészen pontosan Pentagonra, kihasználva a gigascreen adta lehetőségeket. Szüksége van a TR-DOS-ra is, a demó 2,5 Mbájtos, 4 lemezen terpszkedik. Cserébe egészen hosszú, több, mint 5 perces.

Ötletesen vették át az old-school effekteket, megspékelték pofás grafikákkal, hozzáillő, szintén a régi demoscene-t idéző zenével. Az effektek válogatása, sorrendje jó lett, de az átvezetésekkel nem foglalkoztak sokat.

A szexepiljét az adja, hogy végig, 5 percen keresztül folyamatosan használja a bordert, nem lehet észrevenni, hogy hol ér véget a screen és hol kezdődik a border. Sok-sok pepecselés lehetett vele. Kivetítőn nagyot üt egy ilyen demó. Örök darab, még úgy is, hogy nem pure Spectrumon fut. A gigascreen leszámítva a többi kivitelezhető lenne egy 128-ason.

Érdekes effektek:

- Gigascreen logó, borderen mozgó négyzetrács
- Borderre kihúzott 3D négyzetrács, plusz pár pattogó golyó
- Border, border, border... ja, és gigascreen

Értékelés:

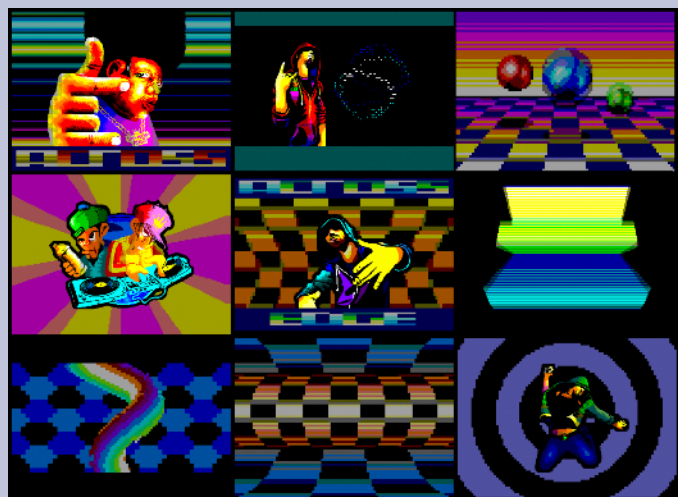
Code (by tmk, bfox):	5
Gfx (by tmk, rainman):	5
Mzx (by n1k-o):	5
Dezign (by zeebr):	4

Ajánló: deMarche stuffok, amit még érdemes megnézni: Mescaline Synesthesia, Paralactika, Scroller, Make Demos Not 1k Intros

Emulátor: Unreal Speccy. Benne van a csomagban, bekonfigurálva.

Videó: <https://youtu.be/b-kkz12foaQ>

G.o.D. / AbaddoN



HARDVER SIMOGATÓ

KLAVIA-TÚRA: BILLENTYŰZET SAGA

Kedvenc Sinclair gépeinknek az egyik hendikeppje bizonyos a billentyűzet, ezért aztán szép számmal készültek a különféle megoldások. Ez a cikksorozat ezen a porondon próbál egy kis útmutatást nyújtani.

SAGA 1 EMPEROR

Az eddig a Sinclair.hu-n bemutatott billentyűzetekkel ellentétben, melyek felhorgadt gyűjtőszendvélyem okán mostanában hozzám került darabok, a Saga 1 Emperor-t hosszú éveken át használtam az első, az igazi Spectrumommal. Az érzelmi kötődés ellenére igyekszem objektív tesztnek alávetni, ez alkalommal hosszú távú tesztelés tapasztalataival megspékelve.

Külső

A hófehér ház esztétikailag telitalálat. Így éreztem, amikor a nyolcvanas években megláttuk az elsőt egy müncheni bolt kirakatában és így érzem most is. A hatvanhét billentyű kiosztásában az ötletszerűség versenyez az átgondoltsággal. Példának okáért szerintem senki sem tud ésszerű magyarázatot adni arra, hogy mire jó két „ENTER” egymás tőzomszédságában, ugyanakkor méltatnom kell azonban a Speccy játékokban kvázi szabványnak számító irányítógombok, a „Fel-le-balra-jobbra Q-A-O-P” elhelyezkedésére hajazó, kurzormozgató billentyű duplikátumokat. A játékok kedvelőinek nem kell magyarázat, hogy ezek miért is jók így és mire. Az sem véletlen, hogy a „Symbol Shift”-ből négyet találunk, lévén gyakran szükségünk lesz rájuk. Ugyanis az „=”, vagy a „#” gombok nem egyetlen gombnyomásra produkálják ezeket a szimbólumokat, hanem használnunk kell hozzájuk a megfelelő shifteket. Összefoglalva, a Saga 1 a kiosztást tekintve az otthoni felhasználók zömét képező játékosoknak és hobbyprogramozóknak készült. Így húsz év távolából,

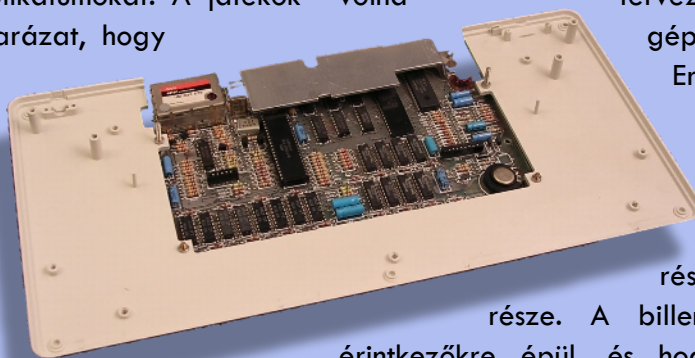
PC-s kiosztásra átszokva is kellemes, könnyen meg lehet tanulni, már ami az alapvető gombokat illeti. A billentyűzet felső fertályán lévő mélyedés szándékolatlan tolltartónak készült, de mindenkit lebeszélnek a rendeltetésszerű használatról, mert nincs is randább, mint az akaratlanul, de összefirkált szépséges ház. (A lefotózott darabról is jó időbe telt eltüntetni a golyóstoll nyomokat.) Egy nagy hiányossága van, mégpedig az, hogy a gombokra matricákkal kerültek a feliratok. A



tapasztalat mondatja velem, hogy ezek huzamosabb használat után lekopnak, elcsúsznak, a szélük felkunkorodik és összeszedik a koszt. Ezek a problémák megfigyelhetők lefotózott darabon is. Úgy tűnik ez törvényszerűen bekövetkezik előbb-utóbb a matricázott billentyűzeteknél, hiszen a Dk'Tronics is ezzel a problémával küzd. A Saga 1 – számos vetélytársához hasonlóan – megtartja az eredeti Spectrum ház alsó platniját és nyugdíjazza a teljes felső részt. Egy csavarhúzó és némi kezűgyesség segítségével öt perc alatt tényleg össze lehet rakni.

Belbecs

Ha tüzetesebben szemügyre vesszük, akkor nagyon úgy tűnik, mintha eredetileg olyan állapot akartak volna tervezni, amit aztán több géphez is felhasználhatnak. Erre utalnak a Saga 1 alsó szendvicsdarabján látható, gumilábak rögzítésére szánt bemélyedések, illetve a hátsó résznek, a buszcsati feletti része. A billentyűzet gumimembrános érintkezőkre épül, és hogy tartós, azt mi sem bizonyítja jobban, minthogy még az én, bizonyosan alapos nyúzásnak kitett, húsz éves darabom is tökéletesen működik. Javítani szerintem esélytelen a billentyűpanel kialakítása okán, de talán nem is lesz rá szükség. Az alaplaphoz „nemszeretem”



műanyagba ágyazott szalagkábelek futnak, de egyrészt ezek jó hosszúak, másrészt nem valószínű, hogy az ember túl sokat szedné szét ezt a részt, így remélhetőleg nem lesz gond a mechanikai ellenállóságukkal. Ami a bővíthetőséget illeti, nincs szégyenkezni valója az Emperornak. Szemre nincs túl nagy hely, de a gyakorlat mást mutat. Kis „fiányákok”, kapcsolók, csatlakozók simán elférnek benne. Ma már nem lenne szívem belekontárkodni, de az eredeti példány-nyomba simán belefért a videó kimenet – a házon belül van is egy kis segédkeret erre a célra, ha szemből nézzük a bal fertályon –, a 80 kilobájtos bővítés, az NMI gomb, a RESET, egy Power LED, de egy módosított hűtőborda is – egy a billentyűzet teljes hosszában elnyúló alumínium idom, rajta az alaplapról áthelyezett feszültségstabilizátorral.

(Látom nem fogom megúszni és le kell fotóznom az öreg csontot is.) Lehet, hogy nem fér bele az Interface 1, mint mondjuk a Transformba, de őszintén szólva ez engem sohasem zavart.

Jó vétel?

Az eredeti Speccy billentyűzethez képest minden csak jobb lehet – kivéve az ISS Super-Keyboard-ot – úgyhogy már csak ezért sincs igazán „rossz vétel” Spectrum billentyűzet. A valódi használati értéket tekintve is úgy gondolom mindenképpen jó vétel, de csak akkor, hogyha a billentyűkön lévő matricák nem kopottak, nem potyognak le, és a gombsapkák nem igényelnek mosásos tisztítást, lévén ezeket még a mai mindenre nyomtató printerek korában is igencsak körülményes pótolni. Ha valakinek van lelke átalakítani, és beleépíteni ezt-azt, akkor is jó bolt. Nem a világ legjobb Speccy billentyűzete – az a Lo Profile – de esztétikus és jól használható, könnyen megtanulható és megszerethető kiosztással.

Summázás

Le kell szögezmem, hogy elfogult vagyok, hiszen hosszú éveken keresztül ilyen billentyűzetet

használtam az első Speccymhez. De talán mégis objektívnek tekinthető az a vélemény, hogy a Saga 1 Emperor nevében van némi egészséges túlzás (Emperor = Császár), de mindenképpen az egyik legigényesebb Spectrum billentyűzet.

Samu József (Sam. Joe)

SAGA 2+

Már régóta szerettem volna egy Saga billentyűzetet beszerezni, de viszonylag ritkán bukkan fel ilyen az e-bayen és akkor is drágán.

Nemrég viszont végre sikerült szert tennem egyből kettőre is: egy Saga 2+ típusúra és egy Saga 1 Emperor-ra, így régi vágyam várt valóra és ha már így esett méltatom is.

Külsín

A ház és a gombok fehér színűek. A 71db billentyűn a Speccy parancsai nem lettek mind feltüntetve, viszont mellékeltek egy fóliát, amin ezek szerepelnek, és ami a billentyűzet részéhez illeszthető. –

felső
Nekem ez szimpatikusabb megoldás, mintha matrica lenne felragasztva, mert az a használat közben lejön, a széle elkoszolódik, pótolni pedig nehéz.

Már első ránézésre is feltűnik, hogy a felső sorban nincsenek numerikus gombok, helyette, többek között, a fájlműveletekhez szükséges parancsok vannak kivezetve. Pl.: RUN, LOAD, SAVE, VERIFY, CAT, INK, PAPER, ERASE. A billentyűzet bal oldalán helyezkedik el a DELETE (hasonlóan a Spectrum Plus-hoz), két oldalt a kurzor gombok, jobb oldalon pedig egy numerikus billentyűzet és az írásjelek.

Belbecs

A Saga 2+ három fő részből áll, az alsó és a felső keretből, illetve magából a billentyűzetből. Ez utóbbinak az aljára van ragasztva a gombok lenyomását érzékelő fólia. A Speccy beszereléséhez el kell távolítani a 48K-ás gép felső részét, az alsó részét pedig négy csavarral rögzíteni kell a Saga aljára. Ezek után csatlakoztatni kell a billentyűzet kábelét – baloldalon egy kis panelt közbe iktatva –



a Speccy alaplapjához. Az át, illetve összerakás annyira egyszerű, hogy elég hozzá egy csavarhúzó és kb. 5-10 perc szabadidő.

Mint ahogyan a többi Saga házban, ebben is van hely az esetleges bővítő paneleknek, illetve látszik belülről, hogy egy másik, pl. video kimenet beépítését is előkészítették.

A működés elve, hasonlóan a ZX Spectrum Plus-hoz az, hogy a műanyag gombok alján van egy szilikon sapka, ami egy többrétegű fóliát nyom meg. A gombok és a fólia között van egy, kb. 1mm vastag szivacszerű réteg, amiből a gombok alatt 12 mm átmérőjű körök vannak kivágva. 22 olyan gomb alatt, amely dedikált funkcióval bír (azaz a Speccy-n egyébként csak több gomb egyidejű lenyomásával érnék el, pl. DELETE, INK, stb.) egy piros, Millennium Falconhoz hasonló alakú műanyag alkatrész van, ami stabilabbá teszi a fólia lenyomását.

Mivel nálam ezek a speciális gombok sajnos csak részben működtek, és mert egy alapos tisztítás egyébként is ráfért a billentyűzetre, szétszedtem

őkelmét. A szétszedés közben viszont sajnos a fólia és a gombok elváltak egymástól, mert a köztük lévő szivacsréteg az idő múlásával elporladt, ezért a szivacsréteget valamilyen módon pótolnom kell majd. Egyelőre még nem adtam fel a reményt, hogy meg tudom javítani a szerkezetet, a javításról mindenképp cikkezem majd egy másik rovat hasábjain.

Jó vétel?

Véleményem szerint a Saga-k a legszebb billentyűzetek, amik valaha a Speccy-khez készültek és még ma is a legjobb vételnek bizonyulnak ebben a kategóriában.

Náray József (Njosef)

SAGA 3 ELITE

A Sinclair ZX Spectrumhoz készült professzionális billentyűzetek történetében a SAGA Systems az a cég, akinek a legszebb termékekért járó képzeletbeli első díjat kellene adni, és talán az a cég, aki a legtöbb különböző modellt kínálta. Az első a sorban a Saga 1 Emperor - mely szerepelt már itt -, míg a rá meglehetősen hasonlító Saga 3 Elite a csúcsmo- dell és sorban az utolsó feltéve, ha nem számoljuk a Lo

Profile-t, melynek mindkét generációját forgalmazták a saját nevük alatt is. Zavarban vagyok a piacra dobás dátumát illetően. A World of Spectrum 1987-re datálja az Elite-et, de ez a dátum a Sinclair Usernek a Saga 2+-ról szóló, és a 3-mast már ismertként emlegető, 1986 februárjában megjelent cikke alapján pontatlannak tűnik, pont úgy mint a gombok száma, mely a valóságban 88.

Külcsín

Legyen a SAGA 3 Elite születésének dátuma akár 86, vagy 87, a PC dominanciája és billentyűkiosztásának kvázi szabványá válása még a távoli jövő. Azzal együtt nagyon valószínű, hogy a tervezője láthatott már XT gombsort, mert a bal szélén helyet foglaló tíz gomb erősen emlékeztet annak funkciógombjaira. Az Elite-en megtalálható minden földi jó; numerikus klaviatúra, kiemelt kurzorvezérlő, nagy méretű "Enter" és még az amúgy "Symbol Shift"-tel előcsalogatható jelek - \$, " és a

többi – is egyetlen gombnyomásra jelennek meg a képernyőn. Még ergonómiáról is beszélhetünk, mert a legalsó gombsor alatti kis em majdnem elég a tenyerünket széles ahhoz, hogy letámasszuk rajta. (Tuningötlet: érdemes kiegészíteni egy a kereskedelemben kapható szivacsos, vagy zselés csuklótámasszal, mert az Elite nagyjából olyan széles, mint egy PC-s gombsor.) A gombsapkákra végre szitázva vannak a feliratok és nem matricákon. Viszont a gyártó elég érdekes módon csak a főbb funkciókat tüntette fel, míg az összes többi egy vékony műanyag lapra nyomtatott számárvezetőre került fel. Nos, itt álljunk meg egy pillanatra! Az én példányom eredeti tulajdonosa ezt a lapot kétoldalú ragasztóval a gombok fölé erősítette – pont passzol oda méretben, talán nem véletlenül –, és a típusjelzés matricáját is áthelyezte a jobb alsó sarokba. Mikor lebontottam a tisztításkor, akkor jöttem rá, hogy miért. Két, viszonylag kicsi, négyzet alakú területen valami megmaradt a ház gyönyörűsége, hófehér műanyag fedelét. Valószínűsítem, hogy az Elite-hez is mellékelhettek eredetileg két tartó fülecskét a számárvezetőhöz – mint a később bemutatott Saga 2+-hoz – amik aztán az idővel letörhettek, vagy elveszhetek. Így az én példányomon elfedi a Saga



termékekre oly jellemző felső tolltartó mélyedést a matrica. Szerencsére egy spanyol oldalon sikerült pár jó képet találnom arról, hogy hogyan is kéne kinéznie makulátlan állapotában egy ilyen szépségnek. Ismét érdekesség, hogy az én példányom alsó lemeze – a Saga 1-gyel ellentétben itt nem marad meg a Speccy eredeti házának alsó darabja – barna színű, míg a hispániai kolléga példányán ez is fehér. A Saga 3 Elite megjele-nésében tán még az eleddig számomra etalonként funkcionáló Lo Profile-t is maga mögé utasítja. De mi van belül?

Belbecs

Mindenekelőtt

meglehetősen sok

üres hely! Bőséggel elférnek

ide mindenféle kiegészítések,

“fianyákok”, “reset” kapcsolók és videókivezetések

- mint azt a mellékelt ábra mutatja, az én darabomat valamely előző tulajdonos el is látta az utóbbi kettővel. A gombsor ugyanazt a gumimembrános technológiát használja, mint a Saga 1 és szemlátomást ugyanaz a cég is gyártotta. Itt torpanjunk meg megint egy pillanatra! Állítom, hogy nincs új a nap alatt! Mi sem bizonyítja ékeesebben, hogy már a nyolcvanas években is léteztek ODM-ek és OEM-ek, mint a ház alsó darabján látható matrica. (Original Design Manufacturer, akik elkészítették az adott terméket egy másik cégnek, hogy az rányalhassa a logóját, bedobozolja és a kiskereskedők felé értékesítse és Original Equipment Manufacturer akik pont azt tették amit épp az előbb említettem.) Jelen esetben az OEM a Saga volt, míg az ODM egy Preh nevű cég, mely bizony mind a mai napig létezik! Valószínűleg csak a tervezőmérnöke tudná megmondani, hogy a billentyűből az alaplaphoz futó két “nyuszifülből” az egyik szándékosan lett úgy kialakítva, ahogy vagy egyszerűen gyártási bakiról van szó. Ugyanis a bal oldali leffentyűn pont az ellenkező oldalra kerültek az érintkezők, mint a Speccy alaplaphoz ültetett foglalóban. Ezt áthidalandó egy kis konverter ékelődik be a kettő közé. Ezen ugyan minden érnél kialakítottak ellenállások beiktatásához forrasztási helyeket, de nem használták ki őket. Lehetséges, hogy ez is olyan elővigyázatosság a Spectrum alaplaphoz változatokkal történő kompatibilitás megőrzésére, mint amelyet már láttunk a Transform esetében? Én a

bakira szavazok, mert a kis átalakítóval a “nyuszifül” túl hosszú lett, valahol vissza kell hajtani, hogy csatlakoztatni lehessen. Itt átkoznám egy kicsit az előző tulaját, aki persze a legostobább módon, az RF modulátor felett hajtotta meg a törékeny szalagot, ami persze a meleg hatására és az idő múlásával némileg megtört. Korrekcióra nincs mód, mert másként hajlított helyzetben már nem vezet, de hogy meddig tart majd ki így, abba bele sem merek gondolni. Ugyanez a gépromboló használt mindenféle szedett-vedett csavarokat a gép összeállításához,

ezzel megrepszette a

billentyű NYÁK-ját

tartó műanyag csonko-

kat. Persze ezt sem

lehet már korigálni

rendes méretű

csavarokkal, mert ami

elrepedt, az ugye

elrepedt. Ismét a rosszul

álasztott csavarok miatt

csálén a gépben, tehát az

megv áll az alaplaphoz RF kivezetés sem a helyén van, amit ez az elvárásolt fakezű lepkevadász úgy gyógyított meg, hogy belevágott egy metrálóalagút méretű lukat a gép hátuljába az RCA csatlakozó köré. Mindezt csak azért tartottam fontosnak megjegyezni, hogy láttassam, hogy az E-Bay-es vásárláshoz jókora adag mázli is kell. A Saga 3 Elite belsejét pont olyan jóra értékelem, mint a Saga 1-es kisöccsénél.

Jó vétel?

Minden szempontból jó vétel. Szépséges megjelenés, jó billentyűkiosztás, bizonyítottan megbízható technológia belül és rengeteg hely a beépítendő kiegészítéseknek. Vajh' ha Sir Clive a Spectrum+ helyett - ami nem több, mint egy látványos, de éppúgy a sokszor elátkozott fóliamembránra épülő konstrukciójú billentyűzet - valami ilyennel jön elő, akkor talán a Sinclair Research története is máshogy alakulhat.

Summázás

Még régebben az AMS Lo Profile billentyűzetét kiáltottam ki az "aftermarket" Spectrum profi billentyűk királyának. (Milyen érdekes, hogy a Saga is árusította ezt a saját logójával!) Nos, akkor a Saga 3 Elite az ilyen eszközök királynője kell, hogy legyen!

Samu József (Sam. Joe)

HARDVER ÖTLETEK - ZX SPECTRUM

RÉSZLETEK FPGA JOCÓ TITKOS NAPLÓJÁBÓL - 1. RÉSZ

... avagy a CPLD Spectrum építésének rejtelmői...

2016.09.06. 16:29

Az FPGA-val szerzett tapasztalatok alapján, és a speccyalisták igényei, szempontjai miatt felmerült bennem egy CPLD-vel megvalósított ZX Spectrum ötlete. A CPLD, mint programozható áramkör az FPGA kistersvéérének mondható. Jóval kevesebbet tud

és jóval egyszerűbb a felépítése is.

Valójában csak egy jó "kövér" GAL vagy PAL áramkör, amiben az

ULA (annak digitális része) megvalósítható. Elavult technológiának számít és viszonylag

nehéz beszerezni, mert már nem is gyártják.

Miért érdemes akkor mégis ezzel a technológiával

foglalkozni? Igazából

nem is érdemes, de érdekes kihívás. Talán egyetlen

érv szól mellette, miszerint a CPLD még közel áll a

klasszikus 5V-os TTL technológiához, ami eredeti ZX Spectrum megalkotásakor elérhető volt. Ennek

köszönhetően az áramkör, noha 3,3V-ról működik, mégis 5V toleráns bemenetekkel rendelkezik, tehát

egy klasszikus Z80-as processzorhoz könnyedén illeszthető.

Az erőforrások igencsak szűkösek, de az ULA, mint

logikai áramkör mégis elfér benne. Ez sok is és kevés is. Sok, hiszen a Spectrum az maga az ULA, annak

képkalkoló képessége az, amiből egy egyszerű Z80-as alaprendszer (CPU, ROM, RAM) ZX Spectrummá

válí. Ugyanakkor kevés is, mert a CPLD videó kimenete csupán digitális RGB lehet (hasonlóan az

eredeti ZX Spectrum 1289-hoz). A digitális RGB (pontosabban RGBI az intenzitás bittel együtt) elavult,

nem találni ma olyan monitort, amely fogadni képes, legfeljebb a klasszikus CRT modellek (CGA,

Commodore 1084), amelyek a gyűjtők jövoltából

még működnek. Szerencsére az RGBI jelből ellenállásokkal elég egyszerű előállítani analóg RGB

és szinkron jeleket, amit már a legtöbb TV-n megtalálható Euro-Scart bemenet tud fogadni. Ez sem

lesz örökéletű sajnos, hiszen ez a fajta bemenet is elavult. Opció még az RGB átalakítása PAL composite

videójellé, amely szintén elavult, de egyelőre a televíziógyártók még használják. Az RGB-CVBS átala-

kításra létezik céláramkör (AD725ARZ), bár ez sem új technológia, de még elérhető.

A háttértár (SD kártya) és az esetleges egyéb mai modern

eszközök (USB billentyűzet, USB game controller) illesztésére PIC mikrokontrollert választottam. A

mikrokontroller C nyelven programozható, bőséges példaprogram és szoftver-

könyvtár áll rendelkezésre. Külön erénye, hogy USB Host módban képes működni, tehát a rendszerhez

ezzel könnyen tudunk USB eszközöket illeszteni majd. Ugyanakkor a mikrokontrollernek rendszervezélő

feladatokat is szánok, adott esetben a Z80-at inaktíválva képes lesz átvenni annak szerepét, így tudja a teljes memória és I/O tartományt olvasni, írni.

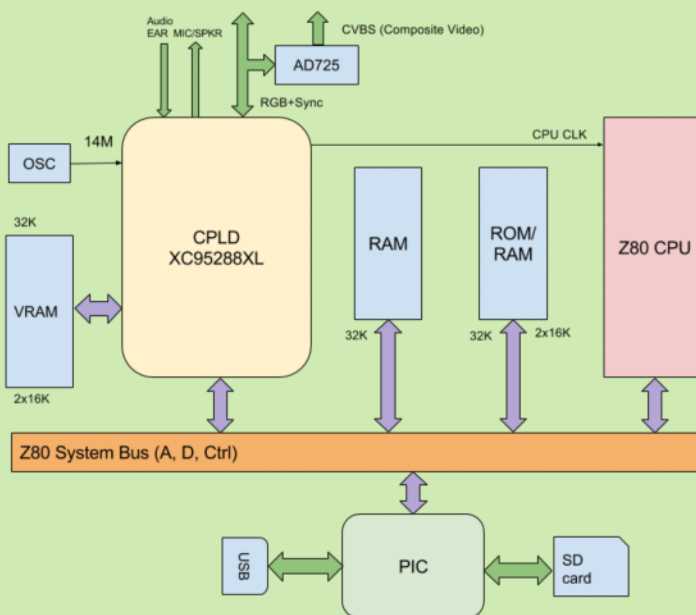
Miért jó ez? Ezen a módon például egyszerű lesz a programok betöltése (SD kártyáról közvetlenül a memóriába), vagy bizonyos extra - az eredeti

Spectrumban nem létező - funkciók megvalósítása. Az alaprendszer ROM nélkül terveztem, a spectrum ROM területére is statikus RAM-ot helyeztem. Ezért a

mikrokontrollernek az induláskor ezt a területet a spectrum ROM-al fel kell töltenie, hogy a Z80 el tudjon indulni.

A rendszer további felépítése, azaz a Z80 CPU és megfelelő mennyiségű statikus RAM kialakítása nem

okozhat gondot, számos működő példát találunk erre. A CPLD-ben megmaradt erőforrásokat



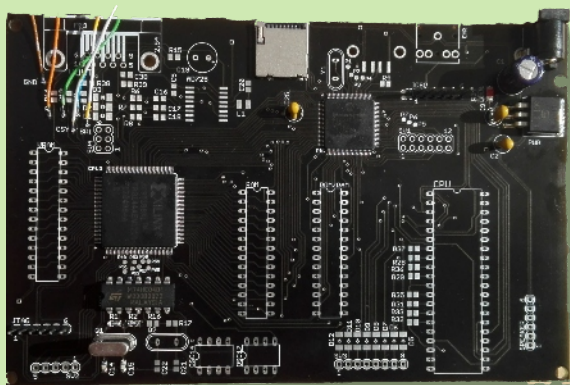
felhasználhatjuk a rendszer bizonyos kapuzásaira (címdekódolás), így valójában több plusz alkatrészre nem lesz szükség. Sajnos a CPLD nem tartalmaz belső oszcillátort, ezért azt hagyományos 74-es TTL vagy CMOS áramkörrel és quartz kristállyal kell kialakítani.

2016.09.06. 16:38

Egy kedves hölgy azelőtt is vállalt már SMD beforrasztást, így most is rábíztam ezt a feladatot (a CPLD-t, a PIC-et és az SD kártyát).

Elsőként a tápfeszültséghez szükséges alkatrészeket forrasztottam be. A Z80 és a memóriák 5V-os tápot igényelnek, de a CPLD és a PIC csak 3.3V-ot. Természetesen mindkét áramkör 5V toleráns bemenetekkel rendelkezik, így az illesztés nem okoz gondot. Amikor ezek az áramkörök a kimeneteken magas szintet adnak, az csak 3.3V körül lesz, de a TTL szabvány szerint az 5V-os áramkörök ezt már logikai 1-nek fogják érzékelni.

A legalapvetőbb tesztekhez a video kimenet elengedhetetlen. A tervezéskor az RGB és szinkron jeleket digitális formában forrpontokra kiveztem, így ezekkel és egy digitális monitorral (Commodore 1084S) valós képjelet kaphatunk.

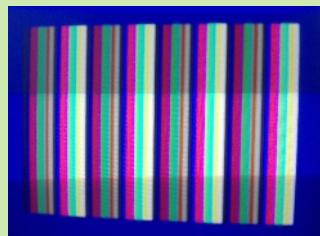
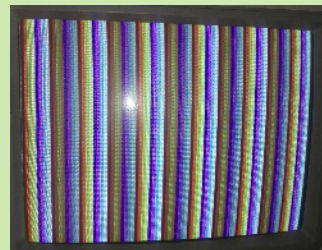


A CPLD számára szükséges 14MHz-es jelet egy 74HC hagyományos inverter állítja elő. A tesztekhez erre mindenképp szükség volt, órajel nélkül az "ULA" nem működik. Beforrasztásra került még a két programozó port (2x6 tűscesor), és egy vörös színű led, így a két alapáramkör, a CPLD és a PIC már tesztelhető lesz.

2016.09.06. 17:00

Az első egyszerű képkalkoló teszt eredménye. Jelzi, hogy az órajel, valamint a CPLD az elképzelt szerint működik. A 7MHz-es pixel órajel osztva 448

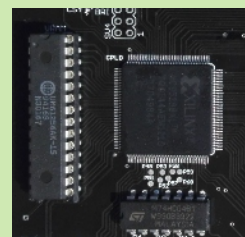
oszlop és 312 rasztorsor előáll, itt a horizontális számláló alapján 4 pixel széles csíkokat rajzol az áramkör. Észrevehető, hogy ez a színskála még nem a ZX Spectrum színeit tartalmazza, mert van benne barna és szürke szín is. Ez még a digitális RGB jel, és a monitor a CGA szabvány szerint jeleníti meg a 4 bites színeket.



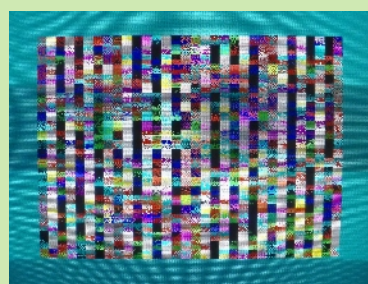
A bitek átvariálása után a színek már a Spectrum szerinti sorban vannak, de még mindig látjuk a barna és szürke színeket. Ezek az analóg RGB kimeneten már nem így jelennek majd meg. Próbaképpen a "Border" színét is állítottam (valójában csak bepozicionáltam a majdani képterületet és adtam egy konstans maszk színt a külső résznek).

2016.09.17. 07:36

Következő lépés volt a video ram beforrasztása. Az eredeti elképzelés szerint (a spectrum +3-hoz hasonlóan) a video ramot a CPU az "ULA" áramkörön keresztül - belső multiplexerekkel kapcsolva - éri majd el. (A beforrasztott video RAM foglalatban, a CPLD mellett bal oldalon)



A video RAM beforrasztása után következett az első ULA szerű kép megjelenítése.



Természetesen a VHDL programot is be kellett tölteni az CPLD-be, ez már majdnem teljes ULA megvalósítás, hasonló ahhoz a kódhoz, amit korábban használtam. A stabil állókép bizonyítja, hogy az időzítések helyesek, a video RAM olvasása megfelelően működik. Ez egy nagy lépés volt előre!

2016.09.17. 07:39

Következő lépés memória elérése a PIC felől. Az elképzelés szerint a végleges rendszerben majd a PIC letiltja a Z80-at (a BUSRQ jel segítségével), így a

RAM-ok tartalma szabadon módosítható. Igen ám, csak hogy a video RAM területet csak a CPLD-n keresztül lehet elérni, amelynek időzítése a 14MHz-es alapjelhez van szinkronizálva. A Z80 szintén ennek fázisban lévő hányadosával működik (3.5MHz), tehát a contended memória elérés ott nem lesz gond. A PIC viszont egy független oszillátortól kapja a 4MHz-es órajelet, ez pedig soha nem lesz szinkronban a 14MHz-es clock domain-el.

Ezért a video RAM-ot a PIC csak úgy tudja elérni, ha a CPLD éppen nem akarja olvasni azt. Szerencsére a video terület 312 raszersorából az ULA (CPLD) csak 192-ben olvas, tehát a fennmaradó 120 sorban az elérés megengedett. A CPLD egy dedikált lábon jelzi a PIC-nek, hogy a video RAM mikor szabad, így az író rutin ezt figyelve tudja elérni a video RAM-ot. (És vár, ha a video terület éppen olvasás alatt áll. Ez lassítja az elérést, de üzemszerűen a PIC-nek egyébként sem kell gyakran ehhez a területhez fordulnia.)

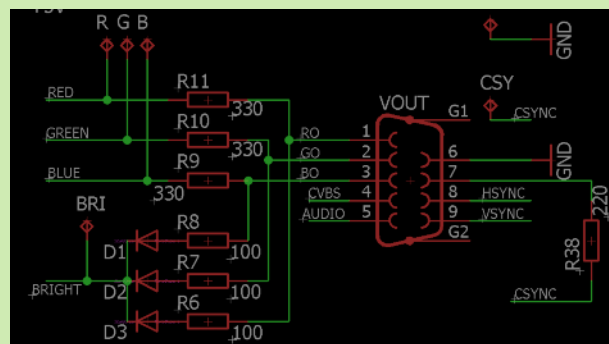
Természetesen ez a probléma nem áll fenn a felső 32K és a Spectrum ROM számára fenntartott RAM területeken, ott nem szükséges a PIC-nek szinkronizálnia az írást a CPLD-hez. Nyilvánvalóan a Z80-at ezekben az esetekben is le kell tilani.

2016.09.23. 12:06

A digitális RGBI átalakítása analóg RGB jellé, illetve kompozit videójellé a következő feladat. Az analóg RGB kialakítása a megfelelő kapcsolással megoldható, természetesen méretezni kell az ellenállásokat figyelembe véve, hogy a CPLD 3.3V-os magas szintet ad, az RGB komponensek pedig max 1V-on lehetnek a legmagasabb intenzitásnál. További fontos adat, hogy a monitorok bemenetei 75 ohmos ellenállást tartalmaznak a GND potenciál felé. Amikor a Bright (intenzitás) jel magas, a diódák lezárnak, tehát a velük sorba kötött ellenállásokon nem esik feszültség. Amikor a Bright jel alacsony, olyankor az ellenállások valójában párhuzamos kapcsolásba kerülnek a termináló 75 ohmos ellenállással.

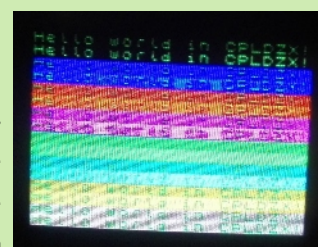
A kompozit videójel előállítását az AD725ARZ típusjelű céláramkör végzi. Ennek működtetéséhez szükség volt egy 17.7344 MHz-es második

oszillátorra, amely a színsegédvidő jel frekvenciájának négyszerese. Az áramkör fogadja a színkomponenseket (a megfelelő kapacitív illesztésen át), valamint a szinkron jeleket, és ezekből előállítja a PAL kompozit videójelet (a fényesség és szín jeleket is, bár ezeket ebben a megoldásban nem használom).



Sajnos itt nem minden úgy alakult, ahogy gondoltam. A Bright jel ki- és bekapcsolása nem mutatott érdemi változást a színekben. Valószínű, hogy az ellenállásokkal sorbakötött diódák nem nyitnak ki, mert túl alacsony nyitófeszültséget kapnak. Végző soron ez logikus. Egy normál dióda nyitófeszültsége 0.7V, ami nagyjából megfelel a teljes szintelítettséghez szükséges feszültségnek. A csökkentett fényerőhöz viszont (amikor a Bright bit 0), elegendő lenne a teljes feszültség 70-75%-a, ami csak 0.5V körüli értékre jön ki. Ez viszont ahhoz kevés, hogy a dióda kinyisson, így gyakorlatilag a Bright bit változása a kimeneti szintet nem befolyásolja. Megoldás lehet schottky dióda alkalmazása, ami már 0.3V körüli nyitófeszültségnél is kinyit, vagy a diódák elhagyása. Ez a második variáció azonban elég összetett eredő ellenállás értékeket adna, mert így az alapszínek egymást is befolyásolnák. Ezt több CPLD kimenet felhasználásával lehet megoldani, vagy egy plusz TTL/CMOS meghajtóval, ami elválasztja az alapszíneket egymástól (pl. 74HC125A) Néhány kísérletet kell még ezzel csinálnom, hogy megfelelő megoldás szülessen. A jó hír viszont, hogy maga az RGB/CVBS konverter (AD725) tökéletesen működik, a kimeneti videójel minősége elfogadható.

UPDATE: beszereztem a schottky diódákat, ezekkel a Bright bit működni látszik. A feszültségszintekhez tartozó ellenállásosztó értéket még ki kell pontosan



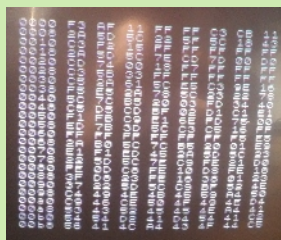
számolnom, de lényegében a bit a fényerőt csökkenti, tehát működik. A mellékelt kép a kompozit videó kimenetről készült, a fénykép mindősége nem túl jó (a valóságban azért ennél jóval szebb) A karakterek kiírását itt még PIC végzi, a Z80 egyelőre nincs az áramkörben.

2016.09.30. 07:32

Tegnap beforrasztottam az adatbusz felhúzó ellenállásokat (8x10K), a billentyűzethez tartozó diódákat és felhúzó ellenállásokat, valamint a DIP foglalatokat a felső 32K RAM, a ROM és a Z80 CPU számára. A processzor egyelőre nem indul. Jó lenne tudni, hogy miért. A ROM feltöltését ellenőriztem, jónak tűnik. Több hibalehetőség is van sajnos:

- Az órajel nem biztos, hogy megfelelő (anno az eredeti spectrumban is egy tranzisztor hajtotta meg, nem pedig azt ULA közvetlenül). Lehet, hogy az LVTTL szint kevés az 5V-on működő processzornak.
- A PIC úgy engedi el a buszt, hogy input-ba kapcsolja az összes adat-, cím- és vezérlő vezetékét. Ez lehet, hogy nem jó így, mert a bemenetek más terhelést jelentenek, mint ha nagyimpedanciás állapotba kerülnének ezek a jelek. Meg kell oldani a nagyimpedanciás állapotot is.
- A Z80 RESET nem stabil. Itt is az lehet a gond, hogy a PIC kapcsolja, és lehet, hogy a kapcsoló lábát már kinyírtam a kísérletek alatt. Kell egy közvetlen RESET kapcsoló.
- Ellenőrizni kell a CPLD-ben az ULA implementációt, illetve kísérletképpen egyszerűsíteni kell. Jó lenne eljutni egy olyan állapotba, hogy a Z80 legalább a keretszint tudja módosítani, és egyelőre a contended memória elérést kikapcsolni.
- Több elérő Z80-al is tesztelni kell. Van legalább 3 féle processzorom különböző gyártóktól. Bár ezek alapvetően kompatibilisek, de elképzelhető, hogy az áramköri viszonyok függvényében eltérően működnek (pl egyik nem annyira érzékeny az órajel szintjére, mint a másik).

A ROM tartalom visszaellenőrzése írás után. Ezt még mindig a PIC csinálja, nem a Z80. Ránézésre jónak tűnik.



(A PIC írja tele a RAM-ot a ROM tartalmával, aztán visszaolvassa.)

2016.10.17. 07:47

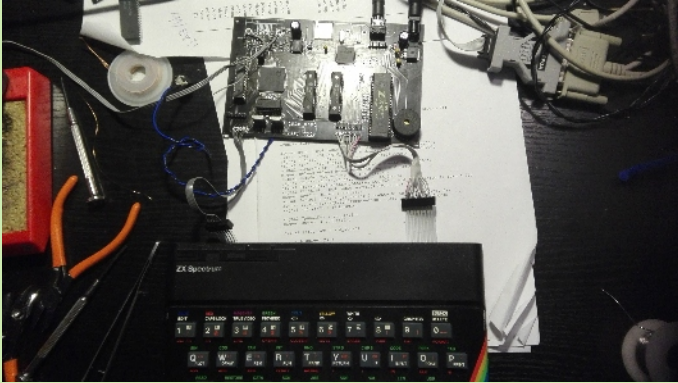
Elindult a Spectrum ROM. Magam is nehezen hiszem, és vannak még problémák, de a bejelentkező felirat megjelent a képernyőn a szokott módon. Rég örültem ennyire ennek az egyszerű fehér képernyőnek. Több hibát is feltártam, amelyek együttesen azért eléggé megnehezítették a dolgomat, de most újra optimista vagyok, hiszen a koncepció szemmel láthatóan jó, és végre kézzel fogható (jól látható) eredménye van a munkámnak. Tehát milyen tervezési és megvalósítási hibákat követtem el?

- A PIC és a Z80 között megosztott busz elég érzékeny pont. Azt hittem, hogyha a PIC lábait bemenetre kapcsolom, az elegendő lesz, és akkor a Z80 már tud működni. Elméletben ez így van, de a gyakorlat azt mutatja, hogy nem igaz. A PIC lábait nagyimpedanciás állapotra kell hozni (és hát az input is ugyanilyen lenne elvben, de mégse). Ilyen üzemmóddja a PIC-nek nincs, de szerencsére tud open-drain kimenetként működni, ami ugyanaz, mint a TTL rendszerekben az open-collector. A PIC lábait open-drain kimenetbe kapcsolom, és 1-re állítom, így azok lezárnak, tehát a külvilág felé nagyimpedanciás állapotot (Z) mutatnak.
- A spectrum speakert két párhuzamosan kötött cmos (74HC04) kapuval akartam meghajtani. A kimenetek is össze vannak kötve és a bemenetek is, elméletben tehát nem lehet zárlat, de a gyakorlat megint mást mutat. Ha a két kapu eltérő késleltetéssel rendelkezik, akár csak ns felbontásban is, lehetséges a zárlat. Valószínűleg meg is történt, mert nem működik a dolog. Most ki kell forrasztanom az IC-t, és szétválasztva a két kaput újat tenni a panelre. Ráadásul ezt az IC-t beforrasztottam, nincs foglalat. Hálátlan feladat. De korrigálom.
- A RESET logikán változtatni kell. Eredetileg a CPLD és a Z80 RESET jeleit összekötöttem, hogy mindkettőt alaphelyzetbe lehessen hozni. Ez nem jó megoldás, mert a CPLD másképp viselkedik, ha épp RESET ciklusban van, mint egyébként, így szét kell választani a két RESET ciklust, és a PIC-el először a CPLD-t, majd utána, amikor a CPLD már működik a Z80-t kell inicializálni.

- Valószínűleg áramköri, vagy forrasztási gondom van, mert a speakert meghajtó láb 1-es állapotánál a rendszer nem működik. Egyelőre ezt a lábat elváltam, így most az indulás stabil.

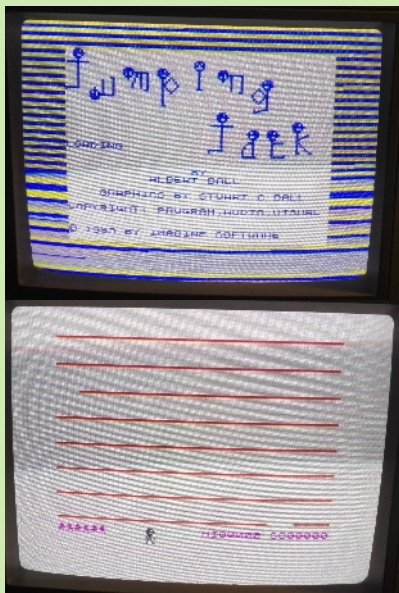
2016.11.03. 13:27

Az illesztett a billentyűzet, végső soron ez így már egy működő Spectrum:



Az EAR bemenet feszültségszintje még nem a legtokéletesebb, de ez nem is meglepő, hiszen ez egy analóg jel, amit egy digitális bemenet fogad. Valószínűleg az eredeti ULA chip-ben volt valamiféle erősítés, hogy a pár száz millivoltos analóg jelet a digitális szintre húzta, nekem viszont ilyen még nincs, csak "bízom" benne, hogy az analóg szint elég magas. A PC-m kimenetéről majdnem maximális hangerővel kapok olyan jelet, amit az EAR bemenet már érzékel. Ez így nyilván nem jó, tehát valamit még alakítani kell rajta, de végső soron a betöltés - ha a jelszint megfelelő - stabilan működik.

Betöltés az EAR bemeneten keresztül:



Elindult az első játék!

László József (FPGAJoco)
Folytatjuk...

HARDVER ÖTLET ATARI JOY TESZTER



Szeretnénk bemutatni nektek egy igen praktikus kis szerkezetet, mellyel Atari típusú botkormányainkat tesztelhetjük gép és teszt programok nélkül. Az eszköz **Gaál Csaba** fejlesztése. Tetszetős, kompakt kivitel, profi kivitelezés. A hátulján helyet kapott egy CR2032-es gombelem, így bárhol használhatjuk. Még a ledék fényerejét is beállíthatjuk egy potméter segítségével.

A múltkoriban igencsak nagy hasznát vettem, amikor egy doboznyi botkormánnyal kellett végeznem.

Szerintem ez senkinek sem hiányozhat a kelléktárából. Amennyiben valakinek felkeltette az érdeklődését írjon a szerkesztőség címére vagy nekem.

Kardos Balázs (Balee)

HARDVER ÖTLETEK - ZX SPECTRUM

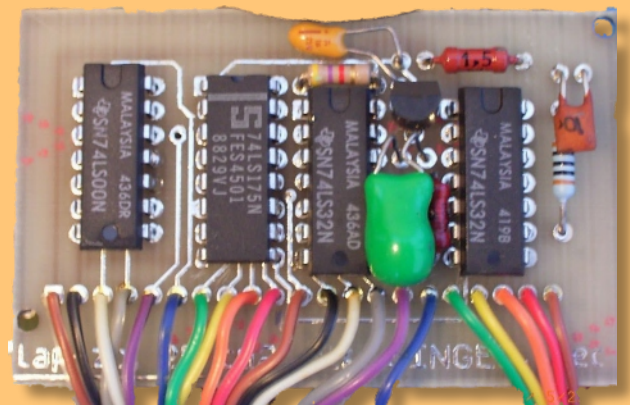
SZUPER 80K

A 80K-ás bővítő mellett anno a "Szellőző Művekben" létezett egy Szuper 80K névre keresztelt megoldás is. Ez az unokaöcsém, **Stock Péter** szellemi terméke. Elég sok panel készült belőle akkoriban. A szerkezet sokkal több volt, mint a hagyományos 80K-s lapozók. Az alapfunkció megmaradt a szokott porton, így amelyik program használta (Super 50K másoló) az lapozta sikeresen.

A ROM helyére lehetett a RAM-ot betenni, amivel ki lehetett próbálni különböző ROM verziókat, (Puzsar ROM, Demak ROM stb.) Nem kellett epromokat égetni, csak betölteni és átkapcsolni.

Lehetőség volt a ROM terület írására is természetesen az árnyék RAM-ba. A videó RAM mellé is be lehetett lapozni egyéb célokra.

A kapcsolási rajzát EAGLE-ben rajzoltam meg némi magyarázattal. A NYÁK terv csak papíron van meg, de nem különösebben nagy méret lehet ötletelni vele.



D2: ROM helyén 64K RAM "L" része (0=kikapcsolva; 1=bekapcsolva)

D3: ROM terület írás engedélyezés (0=nem engedélyezett; 1=engedélyezett, ekkor a megfelelő árnyék RAM-ba írhatunk, ROM kiválasztás esetén is.

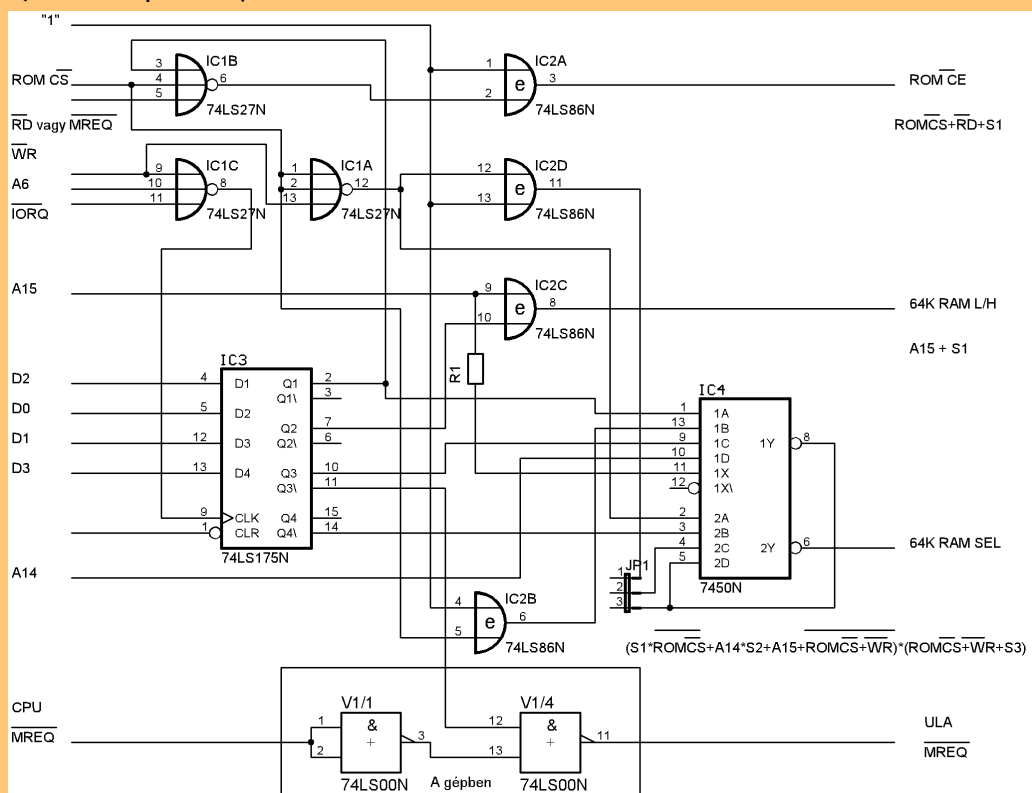
K1: Be=ROM területre íráskor (ha engedélyezett) az árnyék RAM-ba ír. Ki=ROM-ba írva ott nincs RAM. (nem lehet írni)

D0: 64KRAM L/H (0=normál; 1=fordított)

D1: Video RAM mellett 64K "L" része (0=kikapcsolva; 1=bekapcsolva)

Székely Gábor (HA5AWS)

Kép: Horváth László (Leslie.WSS)



REJTVÉNY

Beküldendő a megfejtett idézet, valamint az idézet tulajdonosa 2017. február 28-ig az spv.rejtvény@sinclair.hu email címre. A helyes beküldők között USB ZX Spectrum billentyűzet panelt, ZX diagnosztikai kártyát és játékkazettákat sorsolunk ki. A nyerteseket a következő számban tesszük közzé és email-ben értesítjük Őket. Összesen öt helyes megfejtő kerül díjazásra.

Vízszintes

1. Idézet első része
2. Mindent lesöpör - sárga egyik színárnyalata - Részben kiról!
3. Befed - Férfi keresztnév, az utazók védőszentjéről kapta, aki ezt viseli
4. Az orrához - Az egyik fővárosi metró típus gyártója - Horvát pénz
5. Ásóval és nagyharanggal együtt említik - Papíron rögzített
6. Gyermekgondozási segély, röv. - Végtag - Somogy megyei község - Svéd férfi keresztnév
7. Város Kelet-Flandriában - Ódon - Állatok részére elkerített terület
8. Zéta egynemű betűi - Csuk - Cipel - Női énekhang
9. Bemutató verzió - Lét - Arra a helyre
10. USA-szabvány - A túloldalra - Valaminek a lényege, veleje
11. Német névelő - Feldolgozásra kivágott fa - Orosz vadászrepülő típusa
12. Veszteség - Durva örlemény - Isme
13. Ezüstfehér mésztartalmú csillámpala - Vés - IAY
14. Város Moszkvától délnyugatra - Hangtalan - Tíz -18. hatványát, trilliomodot jelentő előtag
15. Mondat része - Buda német neve - 18. századi irodalomtörténész (Péter) - Amerícium vegyjele
16. Bács-Kiskun megyei község - Sportversenyen a dobogóra állók kapják - Olasz férfinév
17. Videoton számítógépek betűjele volt - A buddhizmus japán változata - Amerikai tábornok volt a polgárháborúban - Mirr-..., a kandúr
18. Csinos, szemrevaló népiesen - Hold angolul
19. Rövid fel- és leszállásra alkalmas repülőgépekre alkalmazott angol mozaikszó - Világbajnok argentin labdarúgó (Mario) - Bór és klór vegyjele
20. Lejátszó készülék - Kiment a divatból
21. Napfizikai műhold-család - Francia város, egy ásványvíz névadója - A bíróságok csúcsszerve

REJTVÉNY

Függőleges

- 1: Szellemdús - Az idézet harmadik része
- 2: Kínai súlymérték volt - Nobel-díjas író, költő, drámaíró - A nézőtér része - Korpusz
- 3: Jellemez - Papi tisztség a rómaiaknál - Lakozik
- 4: VOA - Csörte - Tehát, latinul - Kossuth-díjas matematikus volt (Tibor)
- 5: Megragad - Hosszú morzehang - Ellenérték - ... mars! - Igekötő - Poén angol eredetű szóval
- 6: Forma - Mező - Rövidítés diplomások neve előtt - Világító sugárnyaláb
- 7: Sebes - Kis patak - ... Maria Rilke (osztrák költő) - Orosz realista festő
- 8: Egyforma mássalhangzók - Az idézet második része - Juttat
- 9: Abban az időben - Spanyol cigánylány - Mennyiségegység, röv. - Fradiban játszó spanyol válogatott kézilabdázónő
- 10: Hőmérséklettel kapcsolatos - Róka egynemű betűi - Fogoly - Magasságot fokoz
- 11: Becézett Aranka - Súlyarány, röv. - Képző, -zet párja - Állóvíz - Angol prepozíció - Ipari, kereskedelmi központ Japánban
- 12: Cseh autómárka - József Attila verse - Olasz származású, francia nyelvű belga énekes, a '60-as évek sztárja volt (Salvatore) - VÚG
- 13: Becézett női név - Egyfajta agyagásvány - Oscar Wilde vígjátéka
- 14: Uralkodói szék - Örömittas - Csillag a Sas csillagképben - Renault egyik típusa
- 15: (a kanyarodó fő sor alatt) Victor Hugo regénye

A 2015/2-ES SZÁMUNKBAN MEGJELENT REJTVÉNYÜNK MEGFEJTÉSE ÉS NYERTESÜNK

A megadott határidőig helyes megfejtést beküldő olvasóink
közül sorsoltuk ki a szerencsés nyertest.

Aki, nem más mint:

MEZEI RÓBERT

Gratulálunk!

ZX Spectrum microdrive kazetta nyereményét postán juttatjuk el.

C	A	S	T	L	E	M	A	S	T	E	R
O	L	T		I	S	I	M		E	B	O
L	A		H	Á	Z		A	E	R	O	B
O	R	I	E	N	T	A	L	H	E	R	O
R	I	N	G		E	R	G	O		R	T
O	K	A		Á	R		Á	L	R		F
F		L	É	L		R	M		É	V	A
M	A		D	A	D	A		O	P	E	C
A	L	T	E	R	E	D	B	E	A	S	T
G	L	A	N	C		N	E	P		Z	O
I	A	R		O	M	A	R		A	T	R
C	H	R	I	S	H	I	N	S	L	E	Y

REJTVÉNY

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1						E	GY					S			R
2															
3															
4															
5	T														V
6															
7															
8															
9							L								
10															
11	G														
12															N
13															
14									M						
15															
16															
17															
18	I														U
19															
20															
21															
22	N			E						V					