

SPECCYALISTA VILÁG

A SPECCYALISTA BARÁTI KÖR LAPJA



SINCLAIR.HU

Játékújdonságok

A 90-es évek Pentagon sztorija

A Donkey Kong átszínezése



REGELINTERFACE

Hangdigitalizáló ZX81-re

Mit rejt a Spectrum prototípus ROM?

2023. 1. szám

TARTALOM

BEKÖSZÖNTŐ

LOAD ""

JÁTÉKÚJDONSÁGOK

SPECTRUMOLÓGIA

A 90-ES ÉVEK PENTAGON SZTORIJA

LOAD ""

DONKEY KONG ÁTSZÍNEZÉSE

PROGRAMOZÁSTECHNIKA

HANGDIGITALIZÁLÓ ZX81-RE

HARDVER SIMOGATÓ

REGELINTERFACE

PROGRAMOZÁSTECHNIKA

MIT REJT A SPECTRUM PROTOTÍPUS ROM?

4

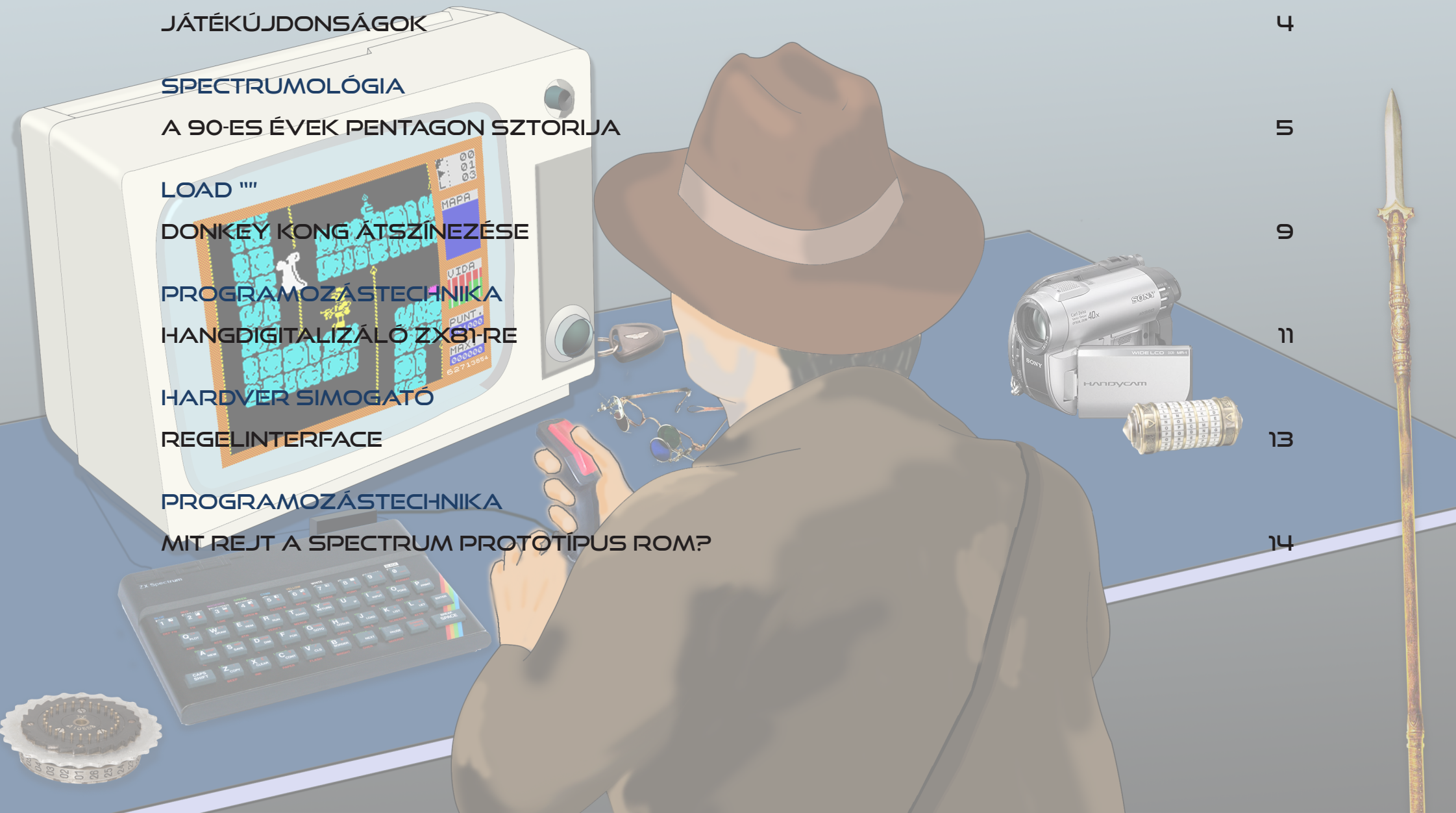
5

9

11

13

14



BEKÖSZÖNTŐ

Kedves olvasó!

Idén végre ismét jelentkezik a Speccyalista Világ! Különösen nagy örömmel tölt el bennünket, hogy ilyen hosszú kihagyás után ismét sikerült összehoznunk egy tartalmas számot.

Most is mint mindig, igyekeztünk az egykori Spectrum Világ szellemiségét megőrizni és minden olvasónknak adni valami számára érdekes olvasmányt.

Mindig azon munkálkodunk, hogy megtartsuk a lap tematikus jellegét, magyarul megpróbáljuk egy-két téma köré felfűzni a tartalmat, emellett persze más anyagok is helyet kaphatnak.

Ezt a számot ismét a régészet köré igyekeztük építeni, mert mindig feltárulhatnak érdekes momentumok Spectrumunk történelméből, így 40 év elmúltával is.

Igyekszünk áttérni a gyakoribb megjelenésre oly módon, hogy amint összegyűlik 16 oldal megtöltéséhez szükséges anyag, akkor azonnal megjelenik majd a lap. Az új, fekvő dizájn az optimális olvasást próbálja támogatni a monitoron, hisz manapság talán leginkább itt olvasgatjuk.

Örömmel jelenthetjük be, hogy már

látogathatjátok azt a külön domain-t (speccyalistavilag.hu), melyet a magazinnak regisztráltunk, ahol a Speccyalista Világ és FanZiX archívumok mellett mindenféle friss hírrel, érdekességgel igyekszünk majd szolgálni a jövőben.

A szerkesztőségünk címére ugyan most is kevés visszajelzés érkezett, de ennek ellenére mindig várjuk leveleteket, hogy olyan lapot hozzassunk létre, amely minden spectrumos ínyére van.

A Sinclair.hu portálon már rengeteg hasznos tudás halmozódott fel az elmúlt években, amelyekből az arra érdemes írásokat időről-időre leportalanított verzióban beválogatjuk majd, melyek így talán még több olvasóhoz juthatnak el.

Szeretnénk továbbra is betekintést engedni a berkeinken belül folyó projektekbe, legyen az akár hardver- vagy szoftverfejlesztés, archiválás, régészkedés.

A Spectrumos témák mellett már az előző számokban elkezdtünk foglalkozni a nagyöreggel, azaz a ZX81-gyel is, ezzel továbbra sem szeretnénk felhagyni, időről-időre helyet kap majd ökelme is.

Remélem a „lap” még mindig elnyeri a tetszéseket és lesznek, akik szívesen bekapcsolódnak a további munkába, akár csak egy-egy cikk erejéig. Ugyanakkor mindenképp várjuk véleményeket, ötleteiteket, melyekkel jobbá, érdekesebbé válhat ez a kis speccyalista kiadvány.

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

2023. június 30.

Kardos Balázs (Balee)

IMPRESSZUM

Főszerkesztő:

Kardos Balázs (Balee)

Lektorálás:

Pgyuri

Egri Imre (Zimi)

Szerkesztés, tördelés:

Kardos Balázs (Balee)

Grafika:

Molnár Péter (Mopi)

Szerzők: Kardos Balázs (Balee), Gondos Csaba (GoCoM), Nagy Dániel, Nickmann László (Niki), Pgyuri, Tarján Richárd (Ricsibu)

Rovatvezetők: Böszörményi Zoltán (Zboszor), Buzogány Csaba (Makranc), Egri Imre (Zimi), Mezei Róbert (M/ZX), Nagy Dániel, Tanács Imre (Kapitány), Kardos Balázs (Balee), Lakatos Péter (Latyi.ca), László József (FPGA Jocó), Pgyuri, Povázsay Zoltán (Povi), Taleyovics Dávid (G.o.D)

Szerkesztőség e-mail címe:

speccyalista.vilag.
szerkesztoseg@sinclair.hu

Kiadó: Speccyalista Baráti Kör

<https://sinclair.hu>

A magazin honlapja:

<https://speccyalistavilag.hu>

2023. június

LOAD ""

JÁTÉKÚJDONSÁGOK

MANIC JET SET WILLY



Szerző: **Norman Sword**
HW igény: **48K**
Stílus: **platform**
Vezérlés: **Billentyűzet**
Megjelenés: **2023., ingyenes**

Talán érdekes ötlet a Manic Miner és a Jet Set Willy összefésülése, valami ilyesmiről szólna ez az új kombó Norman Sword-tól.

Cél összegyűjteni a villogó kulcsokat, és pénzt felhalmozni. Különbőség a normál JSW-től és MM-től, hogy az ütközés pénzbe kerül Willy készpénzéből, ha minden elfogy a játéknak vége.

Irányítás a kezdő képernyőn:

<1> = lejátszási mód módosítása
<enter> = játék indítása

Irányítás játék közben:

<enter> = zene be-/kikapcsolása,
<H> = gyorsabb játék, rövidebb hangok,

<J> = lassabb játékmenet, hosszabb zenei hangok,
<ASDFG> = szünet, - a high score megjelenítése,
<QETUO> = balra,
<WRYIP> = jobbra,
<ZXCVBNM>+<szóköz> = ugrás.

DANTERRIFIK III.



Kiadó: **Toku Soft**
Szerző: **David Garcia**
HW igény: **48K**
Stílus: **platform**
Vezérlés: **Billentyűzet: Q, A, O, P, SPACE, Kempston, Sinclair joystick**
Megjelenés: **2021., ingyenes**

Egy újabb pokoli folytatása a sorozatnak, amely nem csak sok-sok kihívást jelentő pályát tartalmaz, hanem újabb halálos ellenségeket, borzasztó csapdákat, de ha elég okos vagy, megőrizheted mind a 99 életedet! Hát az eddig részeket figyelembe véve talán ez sem lesz elég. :)

Ebben a részben Danként játszol, aki mindenféle fizikai és szellemi kínzással szembesül a gonosz Dantemonium katedrálisban. A hely tele van mindenféle szörnyűséggel és aljas lényekkel, akik itt lakolnak múltbéli tetteikért

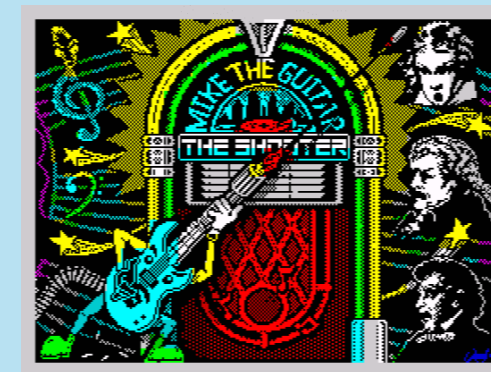
Ezek a német katolikus egyház fejének szavai:
"Az élet nem bocsátja meg a gyengeséget.
Dannek fizetnie kell minden büntetlen tettéért.
Dantemonium lesz az a hely, ahol megbűnhődhet."

Vajon Dan képes lesz visszaszerezni a személyazonosságát, és megtudja, mi történt vele, miért van bezárva a Dantemoniumba?

Rád vár a feladat, hogy fény derülhessen ezen rejtélyekre.

Jonathan Cauldwell A.G.D 4.8-as verziójával készült.

MIKE THE GUITAR



Szerző: **Sebastian Braunert**
Uwe Geiken
Karl McNeil
HW igény: **48K**
Stílus: **platform**
Vezérlés: **Billentyűzet**
Megjelenés: **2023., ingyenes**

Mike visszatért! A híres rockgitár, egyetlen küldetése, hogy hozza el a világnak a Rock'n Rollt!!!

Beethoven, Chopin és Mozart próbálják megállítani a mi kis barátunkat... Gitárhősünknek meg kell küzdenie a klasszikus hangszeres rohamcsapatokkal, mielőtt szembe szállhat a mesterekkel...

"Szeretem a rock and rollt. Szóval dobj még egy érmét a zenegépbe, bébi..."

Kardos Balázs (Balee)

SPECTRUMOLÓGIA

A 90-ES ÉVEK PENTAGON SZTORIJA

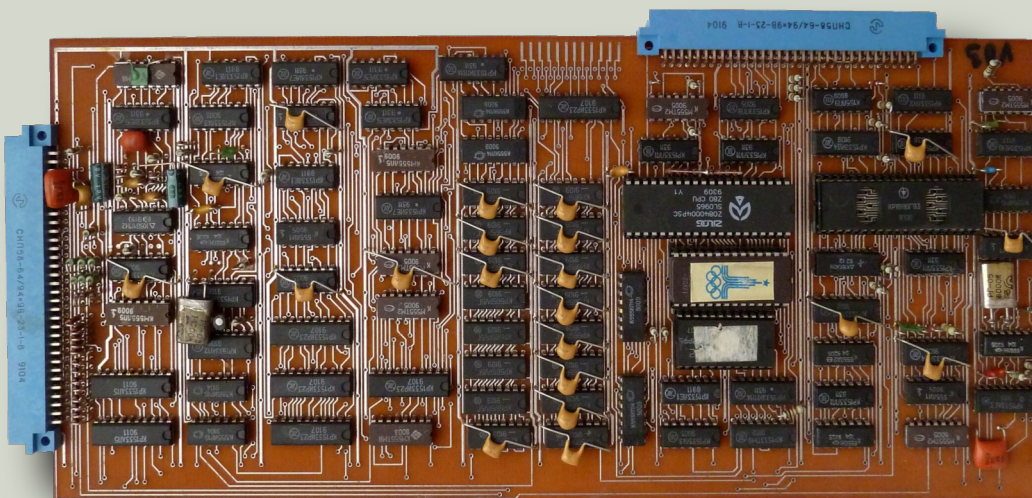
A Pentagonok a volt Szovjetunió utódállamaiban a legszélesebb körben elterjedt számítógépek voltak a 90-es években, melyek fénykorukat a többi ZX Spectrum klónnal együtt 1991 és 1996 között élték.

Elterjedtségüket a szabad hozzáférésű dokumentumoknak és annak köszönhetően, hogy akár házilag is könnyen megépíthetők voltak. Bővítésük is viszonylag egyszerű volt, így a 8 bites mikrók világában extrémnek számító kiegészítők is megjelentek hozzájuk. Többek között processzor tuningok, memória-bővítések, hangkártyák, merevlemez és valós idejű óra illesztések, új képernyőmódok jelentek meg, melyek elsősorban a crackerek, hackerek, fejlesztők vagy swapperek gépeibe kerültek.

A legnépszerűbb konfiguráció a Pentagon 128K lett, Beta-128 lemezegységgel, AY vagy YM hangchippel, esetleg Covox variánsokkal és Kempston egérrel.

A memória és a képernyőrutinok időzítése eltért az eredeti ZX Spectrumétól. Miután a legelterjedtebb gépeknek számítottak, ez a időzítés vált standarddá a demóknál és játékoknál is a volt Szovjetunióban. Tehát az erre érzékeny programok nem futottak tökéletesen más, a Spectrum időzítésével kompatibilis

klónokon és magától értetődően az eredeti Spectrum gépeken sem. Ezen a jobb programok setup alkalmazásával segítettek. Később megjelentek az automatikus Pentagon detektálással rendelkező szoftverek illetve azok, melyek az időzítésre érzéketlen kódolással érték el a kompatibilitást.



Mivel kapcsolási rajzként forogtak közkézen, az egyes Pentagon gépek megjelenése nagymértékben különbözhetett. Volt közöttük hagyományos 8 bites mikró idéző, illetve fekvő vagy toronyházba is szerelték a lapokat.

Processzoraik először az importált Z80A, B és H illetve klónjak voltak, később a hazai gyártású, de NDK-s Z80 klón, az U880 CPU variánsai, a

T34BM1, KP1858BM1/BM3 és KM1858BM1/BM3 lettek. A BM1 jelű processzorok NMOS, a BM3 CMOS alapúak. Ez utóbbi számít a Spectrumoknál a legritkébbnek, lévén a gyártás ideje (1995-96) a klasszikus Spectrum korszak végére esett az ex-Szovjetunió területén. A KP és KM a plasztik illetve kerámia

tokozásra utal.

A különböző verziók és moddingok kavalkádjában lehetetlen nyomonkövetni az egyes verziók számozását és pontos megjelenési dátumaikat. A biztos és bizonytalan pontok kombinálásából azonban relatíve pontos képet kaphatunk ezen gépek evolúciójáról.

A kiindulási pont a Pentagon 48K klón Beta-128 lemez interfésszel, később ezt bővítették folyamatosan az akkoriban már szintén nem a csúcstechnikát képviselő, viszont relatíve olcsón és egyszerű technológiával előállítható modulokkal. Ilyenek a 128K bővítés, az AY/YM-chip, Kempston joystick és ZX Lprint III printer interfészek, illetve a Beta-128 kisebb-nagyobb javításai és bővítései.

A történet ismertetése és dokumentálása közben a nevezéktan némileg önkényes egységesítése is elkerülhetetlen. A saját elnevezéseket félkövér és dőlt betűtípussal, a dokumentációkban és forrásokban szereplő gépneveket dőlt betűvel szedtük.

A legelső gép, a Pentagon 48K 1989-ben látta meg a napvilágot Moszkvában. Konstruktőre **Vladimir Drozdov** ismert rádióamatőr.

Az alaplap szemmel láthatóan két részből áll. Baloldalt maga a ZX Spectrum klón, jobb oldalról kapcsolódik hozzá a Beta-128 interfész. Ha megfordítjuk a lapot, a számítógép felőli rész szinte szabályos ötszöget formál, egyesek szerint innét ered a gép neve is. Más források szerint a gép alapját adó „ősklón” az orosz Honvédelmi Minisztérium főhadiszállásán vagy

valamelyik kutatóintézetben fejlesztették ki (az „orosz Pentagonban”) és Drozdov ennek alapján alkotta meg a gépet. Még ugyanebben az évben megjelent hozzá a 128K bővítés is.

A konstrukció alapját egyesek szerint a *Moskva 128K* (*Moskva 1989, Moskva New*), míg mások szerint a *Leningrád 1 48K* (*Leningrad 1987, Leningrad Old*) gép adja. A hasonlóság különösen az előbbivel megdöbbentő, az alaplap ugyanaz a két részre osztott forma. Ugyancsak közös pont, hogy létezett Pentagon 48K-nak Beta-128 nélküli verziója és a Moskva 128K-ból is készült ilyen változat. Ennél még szembeötlőbb az egyezés a *Zvezda 01 48K* (1991-1993) klónnal... A *Zvezda-001* (a *Zvezda-01* prototípusa) alaplapjának még a színe is pont ugyanolyan árnyalatú, mint a Pentagon 48K-nak és egyben a cég által előzőleg gyártott Agat-9 számítógépnek. Feltételezhető, hogy a megmaradt NYÁK készletek felhasználásáról van szó.

Logikusan végigkövetve a forrásokat és dátumokat, a gyártás valószínűleg a Zvezda gyárában folyt (melynek neve 1991-ig Zagorskij Elektromehaničeskij Zavod – ZEMZ volt) Sergiev Posad városában (korábban a Zagorsk nevet képviselte a település).

Először tehát Drozdov megkonstruálta a gépet a Leningrád 1 48K alapján, később a ZEMZ állami megrendelés híján maradt (Agat-9) elektrotechnikai részlege belefogott a gyártás

előkészítésébe. A klón megkapta a Pentagon 48K márkanévet 1989 nyarán, majd ősszel piacra is dobták. Mindeközben a tervező megalkotta 128K bővítést. 1990 januárjában kapott egy bugfixet Z01.90. néven. A módosítás feltehetőleg az eredeti Spectrummal való jobb kompatibilitást célozza a **Pentagon 48K Z1990. 01.** esetében, akárcsak a későbbi 128K-s nagytesóknál.

Végül a *Pentagon 48K* kifutásakor 1991-ben a gyár piacra dobta a klónt *Zvezda-01* néven, ezúttal már sötétebb NYÁK-on, apróbb fejlesztésekkel és saját házban.

Ezalatt a *Pentagon 48K* és a 128K bővítés, valamint a ZX Lprint III interfész egységesítéséből létrejött a *Pentagon 128K* szintén 1989-ben. Ennek lehetett továbbfejlesztése a Moskva 128K egy Kempston joystick interfésszel és jobb ZX Spectrum kompatibilitással. Mindenesetre a két gép származási helye, évszáma egyezik, valamint ez a kettő az, amelyikről a legkevesebb információval rendelkezünk és specifikációjukban is legközelebb esnek egymáshoz. Tekinthetőek ugyanazon gép két variánsának is. Valószínűleg sosem kerültek nagy számban legyártásra, fotódokumentáció nem is maradt fent róluk, csak kapcsolási rajz. Viszont az ATM és MikroART, illetve Solon és GrandRomMax vegyesvállalatok biztosan szereztek korai Pentagonokat és továbbfejlesztették az alaptípust.

Az **ATM** rövidítés kezdetben a moszkvai **Associaciya Tvorcheskoy Molodezhi**, azaz az **Association of Creative Youth** (Kreatív Fiatalok Egyesülete) nevet takarta, majd az **Association for Technics and Microelectronics** névre váltottak megtartva a már bevezetett rövidítést. A **MikroArt** szintén hasonló, a technika és tudomány iránt elkötelezett fővárosi fiatalok csapatából állt, akik 1992 áprilisától már önálló céget alapítottak a társulásból a gépgyártás és technikai publikáció területén és azóta is működnek.

A szintén moszkvai **Solon**hoz csatlakozott a fryazinoi **GrandRoMax (GRM)**. A Solon cég egy igen univerzális számítástechnikai vállalkozás volt: programok terjesztése, szakirodalom publikálása, hardverfejlesztések: ezek mind-mind a profiljába tartoztak. Jelenleg is működnek mint kiadóvállalat **SolonPress** néven. A partnervállalat hivatalos neve **GrandRomMax Independent Science-Manufacturing Laboratory of Computer Techniques**, tehát a cég magát függetlennek valló számítástechnikai kutató és gyártóintézetként aposztrofálta.

A különböző 128K-s Pentagonok elnevezéseinek tisztázása még a korabeli szakkönyvek, dokumentációk, folyóiratok és diskmag-ok átolvasásával is lehetetlen. Hiszen ugyanazt a gépet még a saját gyártója is másképpen hívta az egyes hirdetésekben, dokumentációkban, nem is beszélve a köznyelv, illetve a

kalózkidadások által teremtett zűrzavarról.

Áttekinthetőbb megoldás nem lévén, érdemes a ZX Spectrum 128K, 128K+2 és 128K+3 elnevezések analógiája alapján generációkra osztani a 128K-s Pentagonokat kiegészítve a fejlesztő nevével. Úgy tűnik a gyártók is valami ilyesemben gondolkodtak, még a nevek is hasonlóak – habár korántsem alkalmazták ezt következetesen. Az évszámoknál a gyártó által használt első ismert hirdetést, illetve a műszaki dokumentáció évszámát vettem figyelembe. Ahol a gyártás és forgalmazás éve valószínűleg a következő évre eshetett, azt is jeleztem.

Tehát, a már említett első generációs *Pentagon 128K* a 128K-s modell (1989) integrált Beta-128 lemez és ZX Lprint III printer interfészekkel. Fejlesztője ismeretlen, csupán az A. K. monogram utal a készítőre.

A második generáció, a *Pentagon 128K 2+* ennek továbbfejlesztése a Beta-128 bugfixeivel, módosításaival és a sorozatgyártásra optimalizálással. Itt már eltűnt az alaplap hátoldalának ötszög formája (ATM-hirdetés: 1990).

A harmadik generáció, a *Pentagon 128K 3+* integrált AY/YM és joystick interfészt kapott (ATM: gyártás kezdete nem ismert, de 1994-ben már biztosan gyártották; Solon-dokumentáció, GRM alaplap: 1993). A lényegi hardverfejlesztések itt le is

záródtak a PC-klónok előretörésével, illetve az ex-szovjet gazdaság elmélyülő recessziójával.

Az utolsó, a negyedik generáció a *Pentagon 128K 4+*, ami már csak méretükben csökkentett alaplapon jelent (Solon-hirdetés: 1994, GrandRomMax dokumentáció: 1995).

Mindezek fényében a részletes evolúció feltehetőleg a következőképpen fest.

A legszélesebb körben elterjedt klón az ATM és a MikroArt gépe lett. Az ATM második generációs gépét tehát 1990 novemberében kezdték hirdetni, a köztudatban megragadt ***Pentagon 128K 2+ ATM***, *Pentagon 128K++ ATM*, *ATM-128K*, *Pentagon 128K/91* és *ATM 1991* elnevezéseket figyelembe véve a forgalmazás 1991-ben kezdődhetett.

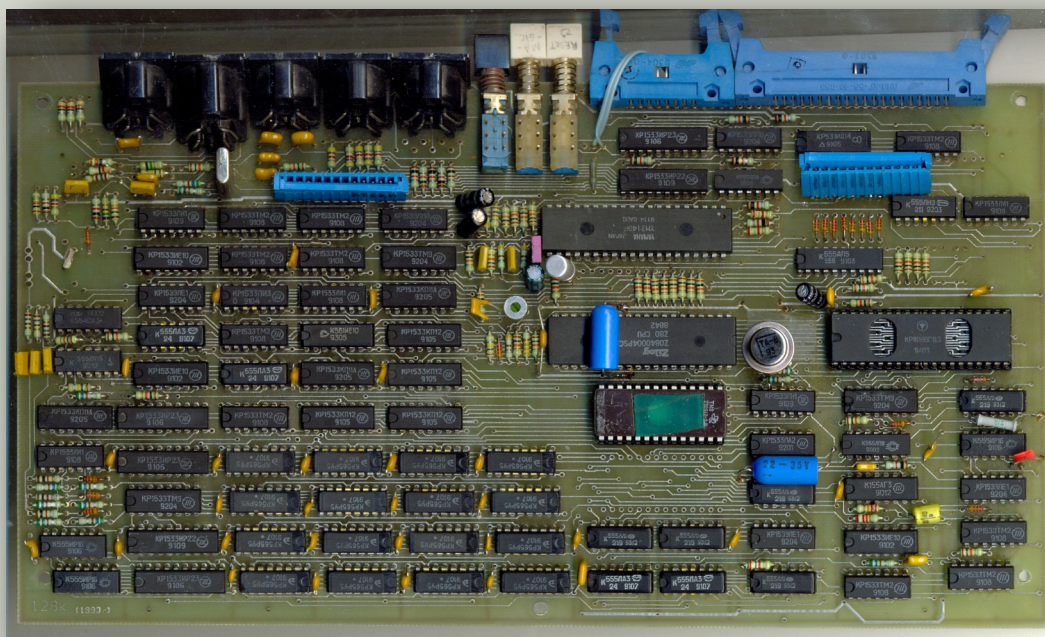
A *Pentagon 48K* és *128K* alaplapon vették kiindulási pontnak, emellett megalkották saját AY-YM adapterkártyájukat is. Ezt a kártyát a Z80 processzor foglalatába kell dugni, és tartalmazza a Z80-at, valamint az AY-chipet (a vezérléshez szükséges összes jel a Z80 aljazon van). A NYÁK-on jól kivehető egy vízszintes üres sor, mely a következő szériánál kap szerepet.

Az ATM harmadik generáció hivatalos elnevezése is *Pentagon 128K 2+ ATM*, de ismert még *Pentagon 128K 2.1*-ként is, holott neve logikusan ***Pentagon 128K 3+ ATM*** lenne. Itt az előző alaplapon vízszintes üres részére

integrálták az AY-chipet és a Kempston vezérlőt. A hangchip a nem szabványos 3,5 MHz-es frekvencián üzemel. A kazettás magnetofon használatát ez a modell már nem támogatta. A ZX Lprint III interfész csatlakozója is hiányzik, a vezérlő ROM foglalatja azonban még

mutatnak: *Pentagon 128K 2+ Solon*, *Pentagon 128K ++ Solon*, *Pentagon 1993*, *Pentagon-128 03*, *Pentagon-128K 1993g* neveken egyaránt ismert.

Az internetes forrásokon felül leginkább a Solon saját felhasználói



megtalálható az alaplapon.

Az ATM második generációs alaplapon két moddingja ismert, a ***Pentagon 128K 2+ ATM Z1992*** és ***Pentagon 128K 2+ ATM Z1994***. Mint a *Pentagon 48K* esetében, ezek is a jobb kompatibilitást szolgálhatták.

A Solon 1993-ban rögtön a harmadik generációval lépett a piacra. A ***Pentagon 128K 3+ Solon*** elnevezései még az előzőeknél is nagyobb káoszt

gépkönyvére támaszkodhatunk, ha mélyebbre akarunk ásni a témában, mely nemes egyszerűséggel a ZX Spectrum 128K címet viseli és 1994-ben jelent meg.

Első alaplajuk a PS.03 feliratot hordozza magán és *Pentagon-128 03*-ként szerepel a Solon partnerének, a *Mikronix*-nek a könyvbéli hirdetésben. Az alaplapon dokumentációján a *Pentagon-128K 1993g* cím található. A lap mérete egyébként

megegyezik a *ATM-128K* dimenzióival, a Solon is egy általuk módosított *Pentagon 128K 2+* gépként hivatkozik rá. A képernyő-időzírtési problémákat kijavították, de pont ez lett a gép egyik vesztesége, hiszen addigra már az ATM-féle időzírtés vált standarddá.

Érdekes módon a gépkönyv külön kiemeli, hogy nem szándékoznak *128K RAM*-nál nagyobbakat használni, hiszen nagyon kevés program igényel ennél több memóriát. A gépekbe mégis 1 darab *256K*-s *RU7* RAM-ot tettek a szokásos 2 darab *64K*-s *RU5* chip helyett, melyből *128K*-t lát alapsan a gép. Bár állítólag a *RU7* chippek stabilabb működésűek a *RU5*-nél, lehet, hogy ez az oka a furcsa ellentmondásnak. Innen egyébként már csak egy kis modding kell a *256K* RAM teljes kihasználásához.

Harmadik generációként itt is integrált lett a sztereó AY áramkör és a joy interfész a Kempston mellett már a Sinclair szabványt is támogatja. A TV mellett monitort is köthetünk rá. Az orosz szabványnak megfelelő DIN csatlakozók már az alaplapon kerültek. Így a gépházak hátuljához egy előre perforált alumínium csatlakozólapot kell felszerelni. A *Beta-128* interfész variánsa, a *Beta Turbo* itt két meghajtót tud kezelni. A firmware a Profi klónból származik (1990 október), amely kétszeres pozícionálási, olvasási és formázási sebességet biztosít, az írás maradt standard sebességű. Az áramkör tápellátásán is javítottak. Ez a *Beta-*

Turbo verzió gyárilag hibás. Ha egy turbós gépen írt lemezre nem turbós gépen írunk, a lemez tartalma megsemmisül. A végleges moddingok a Spectrofon diskmag 12. (1995. március: Vladimir Larkov) és 14. (1995. augusztus: Kirill Gromov) számában jelentek meg. Lényegük, hogy az írási sebességet az írási műveletek előtt kell normál sebességre állítani és nem pedig közben.

A fentebb említett könyvben hirdetésként már szerepel egy mini-toronyház Spectrum 128 felirattal. A valóságban a negyedik generációs PS-04 alaplap egy ehhez igen hasonló házat kapott, mely azonban Solon ZX Spectrum feliratot viseli magán. Nem elírás, az egyik alaplapnál pont, a másiknál kötőjel köti össze a típus és generáció megjelölését! Az előzőek alapján elnevezhetjük a **Pentagon 128K 4+ Solon**-t Pentagon-128 04-nek, vagy a fenti kiadvány beharangozója alapján Pentagon-128K 1994g-nek is. A lap formája drasztikusan megváltozott, viszont a Beta-Turbo körüli dróthalmaz az előző bugfixet sejteti.

Amiről nincsen fotó, az a **Pentagon 128K Solon**. Ez nem tartalmazza az AY/joystick részt, viszont a jobb oldalon levő táblára felforraszt-hatóak ezek. Maga a cég is kínált ilyen mini kiegészítő áramköröket. A fenti gépkönyv egyszerűen Pentagon 128K-nak nevezi. Mindezek után nyugodtan elkeresztelhetjük **Pentagon 128K + Solon**-nak, hiszen egyrészt alapban kínálja a helyet a

bővítésekhez a sima Pentagon 128K-hoz képes, másrészt az időzítések is kijavítva.

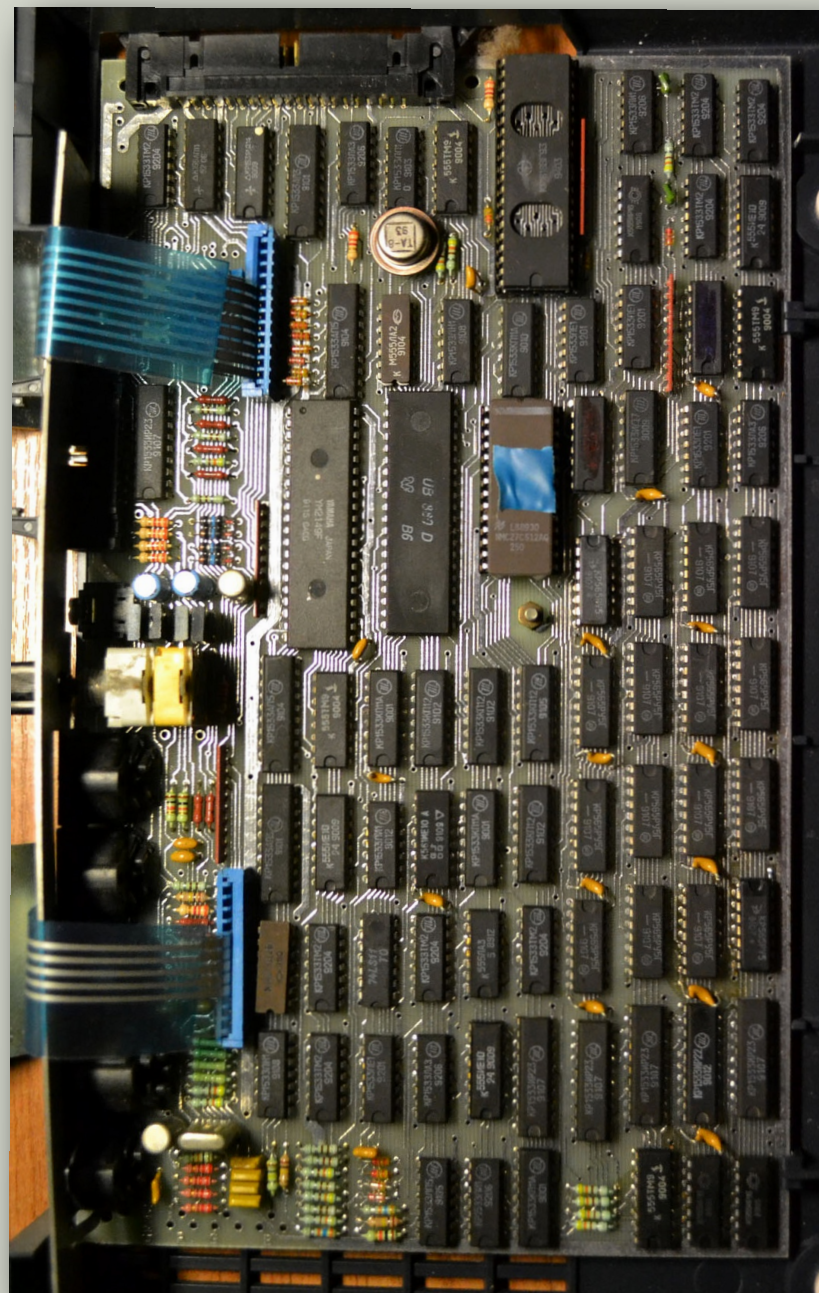
A GRM partnercég szintén 1993-ban kapcsolódott be a gyártásba és még egyszer áttervezték az alaplapokat a harmadik és negyedik generáció esetében. Maradtak az alaplapra integrált DIN csatlakozók és a fém takarólemezek, a lap mérete természetesen csökkent.

Első gépük *GrandRoMax 1* néven indította pályafutását. A lapon a 128K (1993g) felirat olvasható, ez tehát a **Pentagon 128K 3+ GRM**. GRM(1) vagy GRM-1 néven egyaránt ismert lett.

A *GrandRomMax Grandboard 2+* és a *GrandRomMax Grandtower 2+* a **Pentagon 128K 4+ GRM** két változata. Ezek egy új, még keskenyebb alaplapra épültek, mely a GRM2+ nevet viselte. Az alaplap két publikus verzióban létezik: v5: 1995 és v8: 1996. Az első gép a hagyományos 8 bites jellegű kiépítésben, a második mini-toronyként került forgalomba.

A GRM gépek sajnos örökölték a Beta-Turbo hibát, melyet az előbb említett bugfixekkel próbáltak orvosolni.

Tarján Richárd (Ricsibu)



LOAD ""

DONKEY KONG ÁTSZÍNEZÉSE

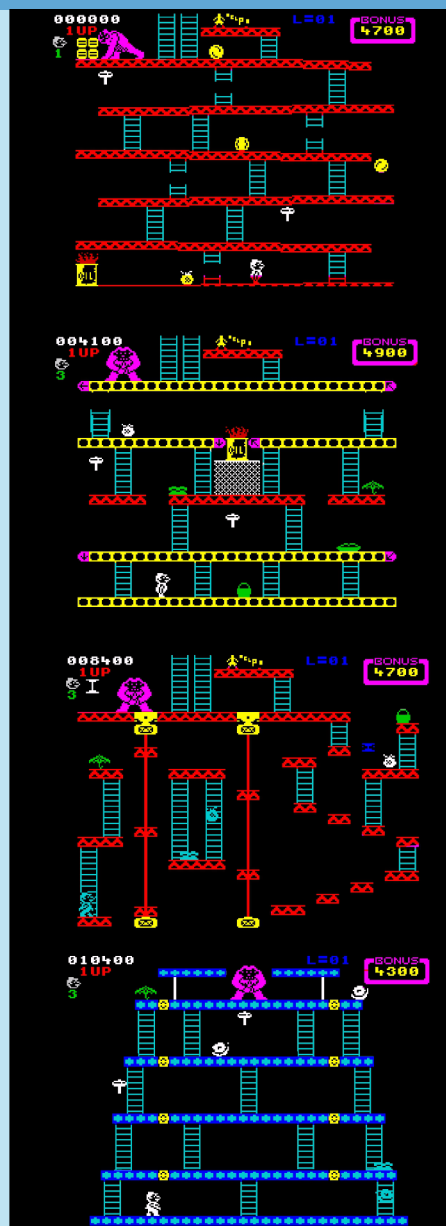
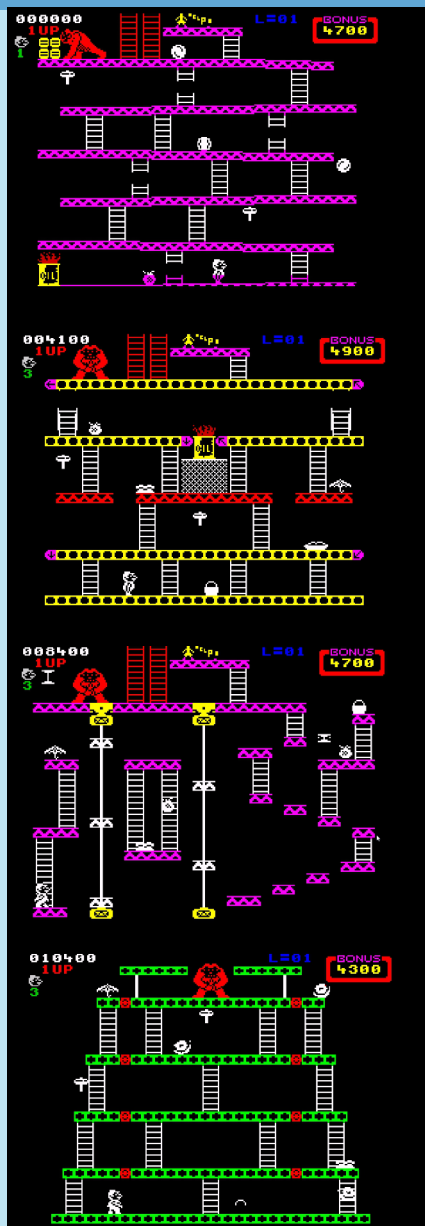
A 80-as évek elején a Nintendo kiadta minden idők egyik leghíresebb játékát, ez volt a Donkey Kong.

Mariónak el kellett kerülnie az akadályokat számos platformon keresztül, néha kalapáccsal a kezében, hogy megmentse a bajba jutott lányt egy óriás hordót hajigáló gorillától, Donkey Kong-tól. Számos különféle platformra portolták, így Spectrum átírat is született.

Rev Stuart Campbell a World of Spectrum (WOS) fórumain tette fel a kérdést, hogy lehet-e módosítani a színeket az Ocean-féle Donkey Kong verzióban, mivel mindig idegesítette, hogy nyilvánvaló ok nélkül minden szinten - legfőképp az utolsó pályán – mások a színek, miközben a Speccy palettájával nagyon hasonlót lehetne megvalósítani, mint az eredeti arcade játék megjelenése.

Nos, Pgyuri állhatatos munkájának köszönhetően mostantól elérhető - Rev Stuart Campbell tervei alapján - az „újrászínezett” verzió.

A következő képeken először az eredeti spectrum verzió, majd az általa javasolt “eredeti” Donkey Kong színezésű terv látható.



Lássuk a kódot...

Sokszor történtek furcsa megoldások egy-egy arcade játék ZX Spectrumra átírásakor. Így történt ennél a játéknál is, sok helyen a színek könnyedén egyezők lehetnek volna az eredeti játékkal, de mégis - néha nem a legjobban - megváltoztatásra kerültek.

A színek cseréje egy játékban nem a legegyszerűbb, sőt sokszor lehetetlen is a kód átírása nélkül, de itt most volt esély az eredetihez hasonló állapotra.

Néhány színkorlát magából a program belső szervezéséből, működéséből adódóan megmaradt, mint például minden pályán ugyanaz az alapszín (fekete-fehér), ami bár átállítható, de akkor sem változtatható termenként és minden tárgy, mozgó objektum ennek a színét veszi fel, tehát a tárgyak sem kaphatnak egyedi színt.

Mindössze 195 bájtnyi cserélgetés, jó pár türelem óra és elkészült a mű.

Néhány szín keresgéléséhez komolyabb kutakodás, visszafejtés kellett, de ez az egyediség jellemző a gépi kódban íródott programokra.

Bonus grafika

\$D615 >> 18 byte \$43 értékre lila keret (pirosról)

Kong grafika

\$D6BB >> 12 byte \$43 értékre lila Kong (pirosról)

Hercegnő helye

\$8C80 >> 1 byte \$42 értékre piros padló a hercegnő helyén, ha Kong elragadja (liláról)

Bevezető

\$D42D >> Minden 4. byte \$45 értékre \$FF-ig világoskék létra (fehérről)

\$D325 >> Minden 6. byte \$42 értékre \$FF-ig piros padló (liláról)

\$D52A >> Minden 4. byte \$42 értékre \$FF-ig piros padló a létránál

\$CB61 >> 1 byte \$03 értékre lila felmászó Kong (pirosról)

1. szint

\$D349 >> Minden 4. byte \$45 értékre \$FF-ig világoskék létra

\$D222 >> Minden 6. byte \$42 értékre \$FF-ig piros padló

\$D442 >> Minden 4. byte \$42 értékre \$FF-ig piros padló a létránál

2. szint

\$D386 >> Minden 4. byte \$45 értékre \$FF-ig világoskék létra

\$D282 >> Minden 6. \$43 értékű byte cseréje \$42 értékre \$FF-ig piros padló (sárga marad)

\$D48B >> Minden 4. \$43 értékű byte cseréje \$42 értékre \$FF-ig piros padló a létránál (sárga marad)

3. szint

\$D3C3 >> Minden 4. byte \$45 értékre \$FF-ig világoskék létra

\$D2B0 >> Minden 6. byte \$42 értékre \$FF-ig piros padló

\$D4C4 >> Minden 4. byte \$42 értékre \$FF-ig piros padló a létránál

4. szint

\$D3F4 >> Minden 4. byte \$45 értékre \$FF-ig világoskék létra

\$D306 >> Minden 6. byte \$42 értékre \$FF-ig sötétkék padló (zöldről)

\$D4F1 >> Minden 4. byte \$41 értékre \$FF-ig sötétkék padló a létránál

\$D6AE >> 1 byte \$46 értékre sárga csapdák a padlóban (pirosról)

Finálé

\$98C4 >> 1 byte \$01 értékre sötétkék leeső padló

\$9928 >> 1 byte \$43 értékre lila zuhanó Kong

\$9971 >> 1 byte \$41 értékre sötétkék padló a szerelmesek alatt

Ha valakinek nem lenne türelme begépelni (kém. én 3-szor csináltam meg, hogy ellenőrizsem magam), akkor letöltheti a módosított programot.

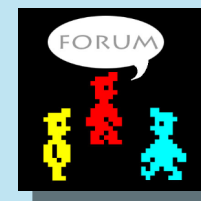
Érdekes, hogy az első pálya a legnehezebb, ha azon sikerül átvergődni, az utána következők elég könnyűek. Volt szerencsém a játéktermi géphez is 5 Ft-os alapon, azért az... igazán ütős. :)

Pgyuri, a Speccyalista Kerekasztalon megjelent bejegyzései alapján, szerkesztette: Bali

Források:

Pgyuri, a Speccyalista Kerekasztalon megjelent bejegyzése

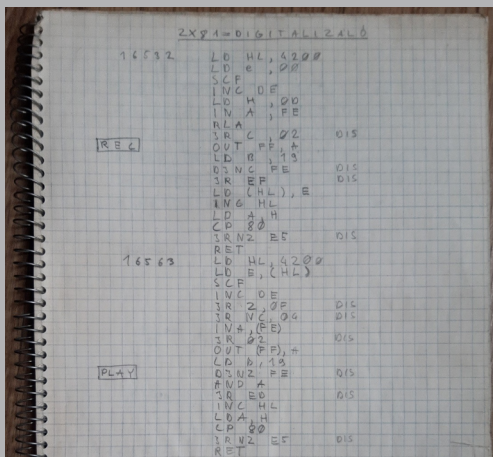
[Indie Retro News - Donkey Kong - Arcade classic now with a ZX Spectrum colour fix!](#)



PROGRAMOZÁSTECHNIKA

HANGDIGITALIZÁLÓ ZX81-RE

A most közzétett programom eredeti forráskódját 1986-ban írtam. Nemrégiben találtam rá az egyik régi spirálfüzetben. Begépeltem, de nem működött, mert sok hibája van. Sajnos csak ezt az assembly forrást találtam meg, hex kód, vagy BASIC loader sehol. Talán valamelyik kazettámon, vagy másik füzetemben megvan. Mindegy, nekiálltam és kijavítottam. A szegényes, egy bites monó tudása ellenére is csodaszám volt akkoriban.

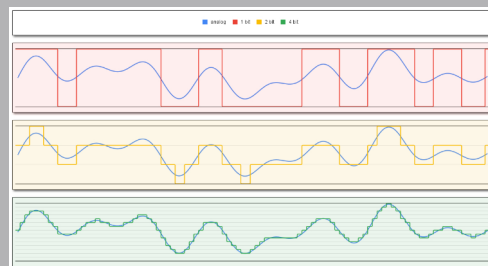


Pár év múlva terveztem és építettem egy 2 bites hangdigitalizálót a C64-hez. Annak már szebb lett a hangja, a kétszeres bitmélység, azaz a részletesebb mintavételezés miatt. Több demóhoz is használtam. Még egy sampler alapú DAW-ot is írtam, ami hasonló volt, mint az Amiga sampe módszere, csak 1 csatornán. Ezután építettem egy 4 bites készüléket is, de

az már csak tesztelve lett, demóhoz/játékhoz semmi sem készült vele. Pedig, a 4 bit már sokkal részletesebb mintavételezést eredményezett, s ezzel jóval tisztább, szebb lett az eredmény. De "a sors keze" úgy hozta, hogy lett egy Amigám, és akkor már inkább ahhoz építettem egy 8 bites digitalizálót.

A digitalizálásról

"Digitális jelfeldolgozásnak (angolul Digital Signal Processing, DSP) vagy digitalizálásnak nevezzük azt a folyamatot, amikor egy fizikai mennyiséget valamilyen módon számítógéppel feldolgozhatóvá tesszünk." Wikipedia



Ez azt jelenti, hogy azt a hangot, amit már analóg elektronikus jellé alakítottunk, még át kell alakítani olyan formára, hogy a számítógép tudjon vele mit tenni. Manapság ezt a hangkártyák végzik, de akkoriban még nem léteztek, illetve, csak a hang szintetizálására alkalmas chippek voltak a legtöbb mikroszámítógépben.

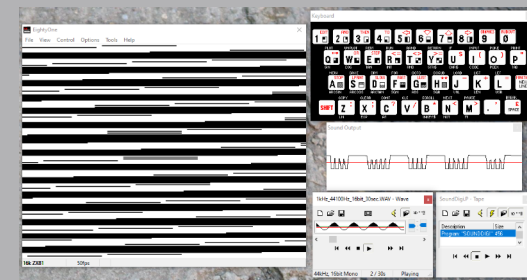
- Az alábbi ábrákon a kék vonal szimbolizálja az analóg hangot.
- Az 1 bites (piros), azaz LO/Hi állapot nagyon durva eredményt ad.
- A 2 bitesnél (sárga) már 4 jelszintet tudunk megkülönböztetni. Látható, hogy némileg finomodott az eredmény, kezd hasonlítani az eredetihez.
- Tovább duplázva a bitek számát, a 4 bit (zöld) 16 jelszintet produkál, ami egész jól leköveti az analóg jelet.

Ez utóbbi megoldást alkalmaztuk a C64-en és más hasonló gépen, ami a maga korában nagyon jónak számított. Ezt már csak az Amiga múlta felül a 8 bitjével (256 jelszint), ami szinte a kék vonal lenne. Ahogy egyre jobb lett a technika, nagyobbak lettek az igények, s bevezették a 16 bitet, amit a mai napig használunk a hétköznapi eszközeinkben, például MP3 lejátszók, mobiltelefonok, TV-k, és a számítógépek hangkártyái is. Ennél nagyobb bitszámot (24/32) csak a stúdiókban használnak.

Amit még fontos tudni, az a mintavételezési frekvencia (azaz, minta/másodperc vagy bit/másodperc), amit általában kilohertz (kHz) adnak meg. Itt is annál szebb a felvétel, minél nagyobb az érték.

Megvalósítás a ZX81-en

Az általános CD-nél 44.1 kHz a mintavételi sebesség, 16-bittel. Ami azt jelenti, hogy másodpercenként 88200 bájtt kerül feldolgozásra. Ez nemcsak a RAM kapacitást haladja meg, hanem a legtöbb korabeli mikroszámítógép CPU sebességét is. Nincs mit tenni, kompromisszumot kell kötni. Ha azt választjuk, hogy maximum 3 kHz-es hangmagasságot akarunk rögzíteni (ez nagyon sok hanganyagnál elég), akkor 6 kHz-es mintavétel kell. Ez 1 bittel 750 bájtt másodpercenként, ami kb. 19 másodperc hanganyagot jelent a 16 kB-os RAM 14 kB-nyi szabad helyén.



Az egy bites mód választásának okáról alább írok.

Mivel a bitszám adott, így amin tudunk változtatni, az a mintavételi frekvencia, aminek a beállítása meghatározza a rögzített hang hosszát, s ezzel fordítottan arányosan változtatva a hang "szépségét". A felvett hang jellege határozza meg az

ajánlott minimális mintavételi frekvenciát.

A program tesztelésénél készített képen jól látni, hogy az 1 kHz-es jel esetén, a nulla állapotban 3-4 mintát vesz. Ez azt jelenti, hogy ezzel a beállítással a maximálisan hibátlanul feldolgozható frekvencia pont a fent említett 3-4 kHz. Nincs más dolgunk, mint egy 3 kHz-es LOW-PASS filtert alkalmazni a digitalizálandó zenére, hogy a felharmonikusok ne zavarjanak be a folyamatba. Ez a beállítás tökéletesen megfelel az emberi beszédhez is.

A további méréseknél látni, hogy a 992 Hz-es jel felvétel után 1010 Hz lesz a lejátszás során. Ez valójában 0,018%-os gyorsulás, ami szinte semmi, de legyünk precízek, ez az eltérés könnyen javítható. Az eltérés abból adódik, hogy a programban a felvételi és visszajátszási utasítások nem ugyanazok. Ha az utasítások M ciklusidejét összeszámoljuk, pár NOP utasítás megoldja.

A program és működése

A ZX81 olyan olcsó kis gépnek készült, hogy semmilyen hangeszköz, még egy beeper sem került bele. Ráadásul még könnyen programozható USER port, LPT port, vagy bármilyen párhuzamos I/O port sem került bele. Van ugyan bonyolultabb egy digitalizálót tenni, mert még egy I/O modullal kell illeszteni a buszra. 17 évesen még nagy feladat lett volna ez akkoriban a számomra, így maradtak hát a



gyárilag ráépített, a programok tárolására és betöltésére szánt magnó-csatlakozók. Ezzel meg is kaptuk a választ arra, hogy miért csak 1 bites lett a digitalizáló programom.

- A program lelke a magnó portot olvasó IN A,(\$FE), és az azt író OUT (\$FF) utasítás.
- A [R/P]LOOP2 ciklus egyenként dolgozza fel az egymás utáni 8 bitet, ami kiadja az 1 bájtot.
- A [R/P]LOOP1 ciklus a beállított HL regiszter szerint indulva a záró memóriacímig megismétli a fenti ciklust.
- A mintavételi sebességet az LD B,\$16 / DJNZ \$FE mikrociklus végzi. A B regiszterbe töltött érték határozza meg a mintavételi sebességet. Ez nem konkrét érték, csak arányszám.

```
//----- Recorder -----
REC:  LD HL,$4400

// START OF LOOP BLOCK
RLOOP1: LD E,$00 // M2 \
        LD D,$08 // M2 / M4

RLOOP2: IN A,($FE) // M3 \
        RLC E // M2 \
        BIT 7,A // M2 \
        JR Z,RSOFF // M2/3 * / M9/10

        SET 0,E // M2 \
        OUT ($FF),A // M3 \
        JR RNEXT // M3 / M8

RSOFF:  NOP // M1 \
        NOP // M1 \
        NOP // M1 \
        NOP // M1 \
        NOP // M1 \
        NOP // M1 \
        NOP // M1 / M7

RNEXT:  LD B,$16 // M2 SPEED \
        DJNZ $FE // M2/3 * \
        DEC D // M1 \
        JR NZ,RLOOP2 // M2/3 * / M7/9

        LD (HL),E // M2 | M2
        // END OF LOOP BLOCK
        // TOTAL M37/40

INC HL
LD A,H
CP $7F
JR NZ,RLOOP1
LD A,L
CP $A0
JR NZ,RLOOP1
RET

//----- Player -----
PLAY:  LD HL,$4400
        LD A,$10

// START OF LOOP BLOCK
PLOOP1: LD E,(HL) // M2 \
        LD D,$08 // M2 / M4

PLOOP2: NOP // M1 \
        NOP // M1 \
        NOP // M1 \
        NOP // M1 \
        NOP // M1 \
        RLC E // M2 \
        JR C,PSOFF // M2/3 * / M9/10

        NOP // M1 \
        NOP // M1 \
        OUT ($FF),A // M3 \
        JR PNEXT // M3 / M8

PSOFF:  NOP // M1 \
        IN A,($FE) // M3 \
        NOP // M1 \
        NOP // M1 \
        NOP // M1 / M7

PNEXT:  LD B,$16 // M2 SPEED \
        DJNZ $FE // M2/3 * \
        DEC D // M1 \
        JR NZ,PLOOP2 // M2/3 * / M7/9

        LD E,$00 // M2 | M2
        // END OF LOOP BLOCK
        // TOTAL M37/40

INC HL
LD A,H
CP $7F
JR NZ,PLOOP1
LD A,L
CP $A0
JR NZ,PLOOP1
RET
```

- A NOP-ok és a programot nem befolyásoló utasítások, csak a már említett pontos szinkronizálás miatt vannak.

Sajnos, a ZX81 a magnó-kimenetet a video-kimenettel közösen használja (ULA chip), így normál kép nem látható a program futása alatt, csak a hangváltozással arányos csíkozását nézhetjük.

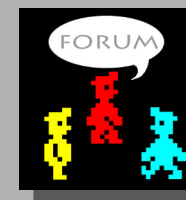
Zárszó és letöltés

Természetesen, ma már ez a program csak érdekességgént, vagy oktatási példaként funkcionálhat. Bár, a

programnak akkoriban sem volt sok haszna, mert a játékhöz használhatatlan, a zenéléshez meg... hát igen. :)

Ha szeretnél játszani vele, itt megtalálod. Minimum 16 kB-os RAM modul szükséges hozzá.

Nickmann László (Niki)



HARDVER SIMOGATÓ

REGELINTERFACE

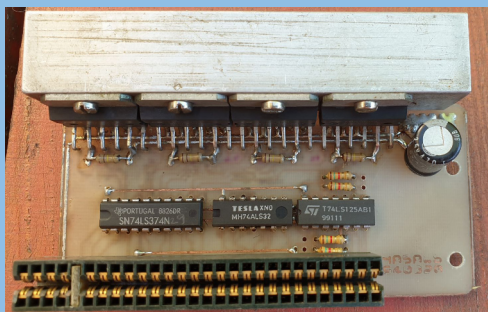
Réges régen, amikor még pezsgett a Spectrum élet, Pista barátunk beszerzett egy "Regelinterface" nevű illesztőt a németországi Hertrich-Computer cégtől.

Az interface számtalan érdekes funkcióval rendelkezett, 8 csatornás analóg és digitális bemenettel, nagyáramú kimenetekkel motorvezérléshez.

Sok, nagyrészt BASIC példaprogramot is küldtek kazettán a hardverhez, amiknek a segítségével saját ötleteket vezérléseket lehetett megvalósítani.

Pista plottert épített függönykarnis-profilból, és műanyag tálcából kivágott, melegen meghajlított alkatrészekből.

Két potméteres csuklós kart épített az analóg bemenetekhez, amivel vektorgrafikus rajzot tudott bevinni a Spectrumba, a házat teljesen digitalizálta 3D-ben!

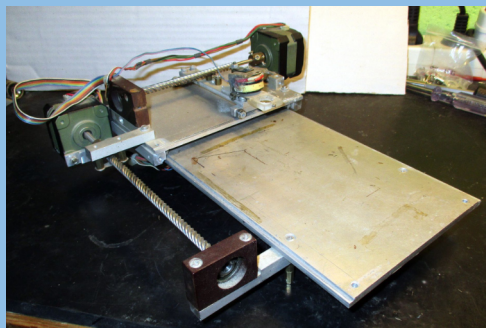


motorvezérlő

Gáborral rögtön megtetszett az ötlet, és nekifogtunk saját ötletekkel és módosításokkal saját illesztőt tervezni.

A motormeghajtó részt különválasztottuk az analóg részekről, előbbiből plotter vezérlő született, utóbbiból tárolós oszcilloszkóp.

A plottert mi már komolyabb mechanikai gépmunka segítségével készítettük, én egy mini változatot 20x10 cm rajzfelülettel, Gábor egy nagyot A4 méretben.



Ekkor került "feláldozásra" a 3"-os floppy meghajtó is, a léptetőmotor és tollmozgató mechanika beszerzés után.

Spectrumhoz szintén Pistán keresztül kaptunk egy NYÁK rajzoló programot ami eredetileg nyomtatóra dolgozott, négy képernyős nagyságú szerkesztővel.

Adódott az ötlet, hogy a tervet rögtön a fóliás lemezre rajzoljuk csőtollal a

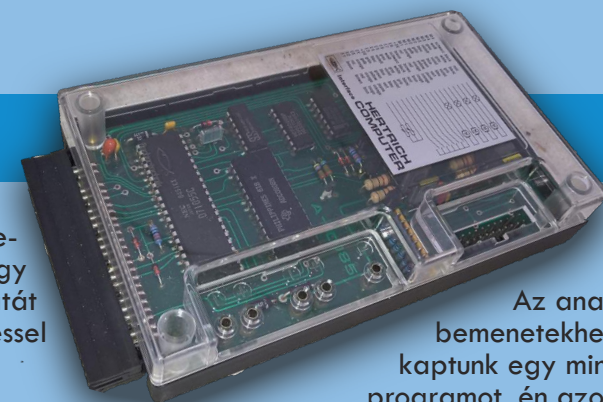
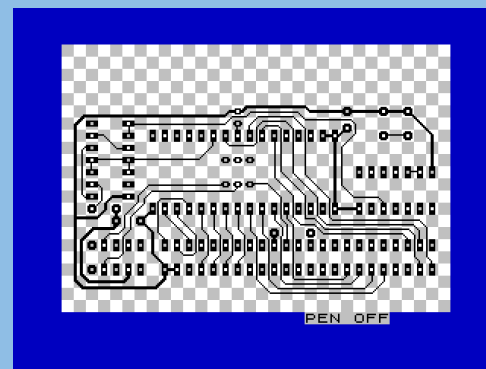
plotter segítségével. Ehhez én írtam egy kiegészítő programot, ami a négy képernyőnyi rész rajzolatát minél kevesebb toll felemeléssel végigköveti a plotteren.

Számtalan áramköri panel született ezután ilyen módon, a mechanika és a program 0,3 mm felbontást tett lehetővé.

Az eredeti illesztőhöz kapott egyik mintaprogram vektoros karektereket tudott BASIC-ben kiegészített parancsokkal rajzolni.

Gábor számtalan készülék előlap feliratozást készített így a plotter segítségével.

A Pataky spectrum klubban Bodor Zoliék (a SIOS alkatói) több vektorgrafikus fontot is adaptáltak a plotterhez, és az érettségi tablójukon a névtáblákat azzal írták meg.



Az analóg bemenetekhez is kaptunk egy mintaprogramot, én azon is módosítottam, mert mi gyorsabb A/D átalakítót használtunk, ehhez pedig kellett az eredeti programon is módosítani.

Az oszcilloszkóp lassú lefutású és max. hangfrekvenciás jelek feldolgozására volt alkalmas.

Konkrét felhasználása volt, amikor egy tenisz edző ismerőssel a különböző márkájú teniszütők csuklóra gyakorolt visszahatását vizsgáltuk, a nyélre rögzített rezgés átalakítóval. Az eredményeket a pályán elmentettük és később elemeztük.

Az analóg illesztő nemrég került ismét elő a doboz mélyéről egy másik projekt kapcsán, de erről majd legközelebb.

Gondos Csaba (Csaba)



PROGRAMOZÁSTECHNIKA

MIT REJT A SPECTRUM PROTOTÍPUS ROM?

Mint az közismert, a ZX Spectrum ROM-ja (firmware) nem teljesen készült el, amikor a gép legyártásra került. Az egyik hobbi projektem egy olyan saját ROM írása, amiben a kód úgy működik, ahogyan eredetileg tervezték és mindenben megfelel a Manualnak (ellenben a ROM-mal, ami néhány apróságban eltér a manuáltól), amit Steven Vickers írt, a ROM főszerzője (ezt tekintjük az igazi specifikációnak). Emiatt a projekt miatt izgatottan vizsgáltam meg a ZX Spectrum prototípus ROM-ját, amit John Grant, a Nine Tiles vezérigazgatója nagylelkűen a British Centre for Computing History-nak adományozott. John Grant a ZX Spectrum ROM-ján is dolgozott és ő az egyedüli szerzője a ZX80 ROM-nak, amiből a ZX Spectrum is örökölt részeket.

Amint a ROM dump elérhetővé vált a Centre honlapján, ahogy másoknak is, az első dolgom nekem is az volt, hogy elindítottam egy Spectrum emulátort a ROM-mal. A saját ZX82 projektemhez már korábban írtam teszteket minden olyan hibára amit ismerek a kifejezés kiértékelő alrendszerben. Némileg csalódottan vettem észre, hogy ezek a hibák, kivétel nélkül még mindig jelen vannak. Aztán végigpróbáltam minden ismert hibát a megjelenítő driverben. Hasonlóan, egy hiba sincs javítva. Lássuk az NMI (meleg reset) bugot! Szintén. Amikor írtam pár sor

BASIC kódot és RUN-nal futtattam, M RAMTOP no good error hibát kaptam. GO TO utasítással futtatva működött és mivel a ZX Spectrumban a RUN valójában egy CLEAR és egy GOTO, kipróbáltam a CLEAR-t is. Ugyanazt a hibaüzenetet kaptam. Felfedeztem egy új hibát, ami a végleges ROM-ban nincs benne.

A hexdump három érdekes változásra világít rá:

- a MOVE token kapott egy hash jelet: MOVE # (mint pl. az OPEN # vagy a CLOSE #), amit a prototípus billentyűzetére nem vezettek át
- egy új hiba üzenet került hozzáadásra: S Device unformatted
- a karakterkészlet előtti üres hely sokkal kisebb lett

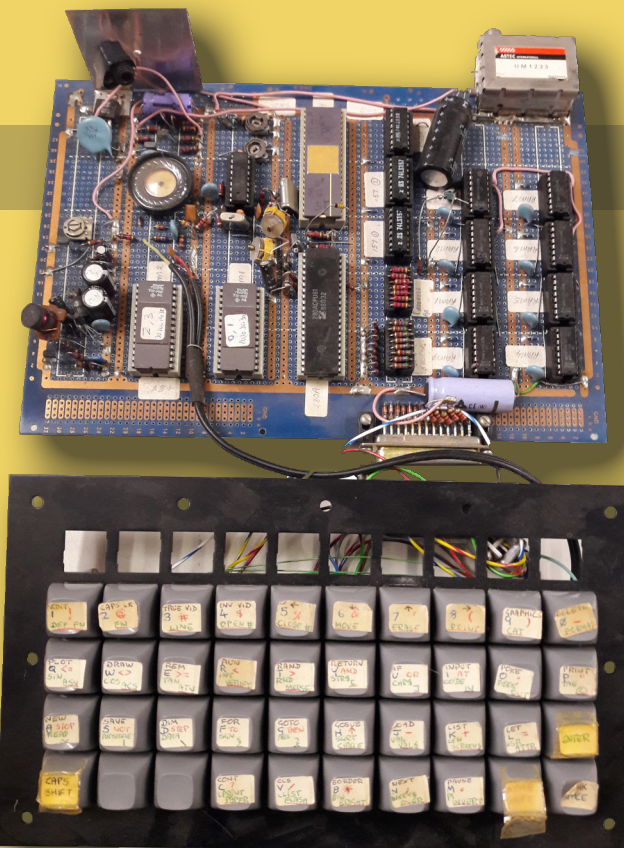
Ezen a ponton világossá vált, hogy a Nine Tiles az új featureök fejlesztésével volt elfoglalva a hibajavítás helyett (valószínűleg nem is tudtak akkor a hibákról). Remélve hogy többet is megtudok az új featureökről, egy disassemblerrel felfegyverkezve belevágtam. Ez a ROM, ahogyan a hexdumpból is látjuk, egy ponton fejlettebb a végleges ROM-nál, ez a terület pedig az I/O alrendszer. A fejlettebb nem feltétlen jelenti, hogy használhatóbb

is: ez a ROM nyilvánvalóan fejlesztés alatt állt, néhány dolog, ami működött a végleges ROM-ban, itt nem működik és néhány része minőségben nem üti meg egy végleges ROM szintjét (a tényleges végleges ROM egy PoC volt egy kiállítási demóhoz).

Az első meglepetés a ROM legelső byteja. A végleges ROM-ban ez egy DI utasítás, ami letiltja a maszkolható megszakításokat. Jó oka van annak, hogy az indító rutin ezzel az utasítással kezdődik: ugyan a maszkolható megszakításokat letiltja a Z80 egy RESET után, a gépet úgy is lehet resetelni, hogy a 0-s címre ugrunk a programból. Ebben az esetben az engedélyezett megszakítások problémát okozhatnak. Igaz, ez nem egy kritikus probléma. Kövessük az indító procedurát most, hogy már ismerjük a különbségeket:

```
; top address
0000  START:
        ld de, 0ffffh
; initialize UDG's and RS232
0003  ld  b, 001h
0005  jp   RESET
1545  RESET:
        di
; 0x5C80 (PR_CC) in this build
```

```
1546  ld    iy, PTR_IY
154a  ld    a, 007h
154c  out   (0feh), a
154e  ld    a, 03fh
1550  ld    i, a
1552  ld    h, d
1553  ld    l, e
1554  RAM_FILL:
        ld    (hl), 002h
1556  dec   hl
1557  cp    h
1558  jr    nz, RAM_FILL
155a  RAM_READ:
155a  and   a
155b  sbc   hl, de
155d  add   hl, de
155e  inc   hl
```



```

155f jr nc, RAM_DONE
1561 dec (hl)
1562 jr z, RAM_DONE
1564 dec (hl)
1565 jr z, RAM_READ
1567 RAM_DONE:
dec hl
1568 push hl
1569 djnz RAM_SET
156b ld de, l3eafh
156e ld bc, 00a8h
1571 ex de,hl
1572 lddr
1574 ld h, d
1575 ld l, e
1576 ex (sp), hl
1577 inc de
1578 ld ix, 0000ch
157c ld bc, 00040h
157f exx
1580 RAM_SET:
exx
1581 ld (P_RAMT), hl
1584 ld (RASP), bc
1588 ld (UDG), de
158c ld (BAUD), ix
1590 ld hl, FONT - 0100h
1593 ld (CHARS), hl
1596 pop hl
1597 ld (RAMTOP), hl
159a ld (hl), 03eh
159c dec hl
159d ld sp, hl
159e dec hl
159f dec hl
15a0 ld (ERR_SP), hl
15a3 im 1
15a5 ei
15a6 ld hl, CHINFO
15a9 ld (CHANS), hl

```

Hasonlít a végleges ROM-hoz, de néhány dolgot áthelyeztek, a 24-T-

state bemelegedési késleltetését eltávolították és az UDG terület mellett az RS232 sebességét is az eredetire állították vissza. Egy nagyon fontos változás ami miatt ez a ROM nem működik sok meglévő szoftverrel és hardverrel amit a végleges ROM-hoz írtak az az, hogy az IY regiszter PR_CC-re (5C80) mutat az ERR_NR helyett, megváltoztatva ezzel az összes rendszerváltozó offszetjét. A változás logikus: elméleti maximumra növeli az elérhető rendszerváltozók számát az IY offszeten keresztül. Az előző értékkel néhány változó nem elérhető (erről később beszélek).

Ez az a rész ami tartalmazza a CLEAR és RUN hibát, ami korábban említettem. A 1568 és 1576-on lévő utasítások feltételezik, hogy a stack pointer már inicializálva van, de valójában ez csak később történik meg, 159d-nél. A következő történik: indításkor az UDG inicializáció felülírja a pointert, ami a fizikai RAM tetejére mutat a vermen és így a P_RAMT rendszerváltozó hibásan kerül inicializálásra, ami eltöri a CLEAR rutint.

Mivel ebben a ROM-ban több rendszerváltozó van a CHINFO címen, a csatorna információs terület, ami a rendszerváltozók után jön, megváltozott. Az egyik közülük, amit én BAUD-nak hívo a ZX Interface 1 és a ZX Spectrum 128k hasonló funkcionalitású változója miatt, az RSR232 időzítés számlálójának kezdeti értékét tartalmazza.

Az inicializáló procedura további

része nagyrészt ugyanaz, mint a végleges ROM-ban. Nem kerülheti el a figyelmünket azonban az, hogy a kezdeti csatorna információ hosszabb lett. Nézzük meg közelebbről:

```

192b CHINFO_START:
192b K_CH:
defw PRINT_OUT
192d defw KEY_INPUT
192f defb "K"
1930 S_CH:
defw PRINT_OUT
1932 defw REPORT_J
1934 defb "S"
1935 R_CH:
defw ADD_CHAR
1937 defw REPORT_J
1939 defb "R"
193a P_CH:
defw PRINT_OUT
193c defw REPORT_J
193e defb "P"
193f T_CH:
defw SERIAL_OUT
1941 defw SERIAL_IN
1943 defb "T"

```

Egy új csatorna típus, a "T", jelent meg az RS232 miatt. A stream tábla alapján nincs default stream hozzárendelve, de ez teljesen jó így. A rutinok vagy működnek vagy nem és mivel a prototípus hardvernek nincs soros interface-e, ezért nem lehet tesztelni. Az FD számú portot a 128k-ban teljesen másra használták, így nem létezik konkrét hardver ami ezekkel a rutinokkal működne. A ZX Interface 1, amihez vélhetően ezeket a rutinokat fejlesztették, végül az F7-es portot használta. Nagyon valószínű, hogy a rutinok korai fejlesztési

fázisban vannak, mivel jelen formájukban, még ha működnének is, nem tudnának meghajtani egy soros nyomtatót az LLIST detokenizálásának hiányában. A személyes véleményem az, hogy ellenben a ZX Interface 1 és a ZX Spectrum 128k működésével, a detokenizer rendes helye az LLIST utasítás maga, kellene, hogy legyen, nem pedig a output service rutin. Talán a Nine Tiles is ezt az utat választotta volna, azonban sajnos Ian Loganek nem volt más lehetősége a ZX Interface 1-ben, mivel az LLIST utasítás a végleges ROM-ban (csakúgy mint ebben a ROM-ban) csak a LIST #3 rövidítése.

A másik jelentős különbség a végleges ROM-mal szemben a rutin, ami a SAVE, LOAD, VERIFY és MERGE utasításokat implementálja, nézzük meg:

```

0607 SAVE_ETC:
0607 pop af
0608 ld a, (T_ADDR)
; reflects difference in syntax table
060b sub 0ech
060d ld (T_ADDR), a
0610 rst 18h
; Streams can be used!
0611 cp "#"
0613 jr z, USE_STREAM
; file name
0615 call CLASS_0A
0618 and a
0619 jr USE_NAME
061b USE_STREAM:
061b rst 20h
; stream
061c call CLASS_06
061f scf

```

0620	USE_NAME:	065c	ld bc, 000Ah	1a72	add hl, de	1aa6	defb "P"
0620	call SYNTAX_Z	065f	SA_NAME:	1a73	jr c, CLOSE_1	1aa7	defb CLOSE_P - \$
0623	jr z, SA_DATA		push ix	1a75	ld bc, STRMS_END	1aa8	defb "N"
<i>; CF'=1 - stream, CF'=0 - file name</i>				1a78	add hl, bc	1aa9	defb CLOSE_N - \$
0625	ex af, af'	0661	pop hl	1a79	ld c, (hl)	1aaa	defb "T"
0626	ld bc, 00011h	0662	inc hl	1a7a	inc hl	1aab	defb CLOSE_T - \$
0629	ld a, (T_ADDR)	0663	ex de, hl	1a7b	ld b, (hl)	1aac	defb 00h
062c	and a	0666	jr SA_DATA	1a7c	CLOSE_1:	1aad	CLOSE_N:
062d	jr z, SA_SPACE	0668	USE_STRM_DO:		ex de, hl		call CLOSE_N_DO
062f	ld c, 022h	0668	call SET_STREAM	1a7d	ld (hl), c	1ab0	CLOSE_T:
0631	SA_SPACE:	066b	SA_DATA:	1a7e	inc hl	1ab0	CLOSE_P:
0631	rst 30h		rst 18h	1a7f	ld (hl), b	1ab0	CLOSE_S:
0632	push de			1a80	ret	1ab0	CLOSE_K:
0633	pop ix			1a81	CLOSE_2:		pop hl
0635	ld b, 00bh				push hl	1ab1	ret
0637	ld a, 020h			1a82	ld hl, (CHANS)	1ab2	STR_DATA:
0639	SA_BLANK:			1a85	add hl, bc	1ab2	call FIND_INT1
0639	ld (de), a			1a86	inc hl	1ab5	cp 010h
063a	inc de			1a87	inc hl	1ab7	ret c
063b	djnz SA_BLANK			1a88	inc hl	1ab8	REPORT_O2:
<i>; restore stream/filename flag</i>				1a89	ld c, (hl)	1ab8	rst 8
063d	ex af, af'			1a8a	ex de, hl		<i>; O Invalid stream</i>
063e	sbc a, a			1a8b	ld hl, CLOSESTRM	1ab9	defb 17h
063f	dec a			1a8e	call INDEXER	1aba	STR_DATA2:
<i>; \$FF - filename \$FE - stream</i>				1a91	jr nc, CLOSE_3	1aba	call STR_DATA
0640	ld (ix+001h), a			1a93	ld c, (hl)	1abd	STR_DATA1:
0643	jr c, USE_STRM_DO			1a94	ld b, 000h	1abd	add a, a
0645	call sub_2fe6h			1a96	add hl, bc	1abe	add a, 016h
0648	ld hl, 0fff6h			1a97	jp (hl)	1ac0	ld l, a
064b	dec bc			1a98	CLOSE_3:	1ac1	ld h, 05ch
064c	add hl, bc				ld hl, 0005h	1ac3	ld c, (hl)
064d	inc bc			1a9b	add hl, de	1ac4	inc hl
064e	jr nc, SA_NAME			1a9c	ld a, (hl)	1ac5	ld b, (hl)
0650	ld a, (T_ADDR)			1a9d	inc hl	1ac6	dec hl
0653	and a			1a9e	ld h, (hl)	1ac7	ret
0654	jr nz, SA_NULL			1a9f	ld l, a	1ac8	OPEN:
0656	rst 8			1aa0	jp (hl)		rst 28h
<i>; F Invalid file name</i>				1aa1	jp (hl)	1ac9	defb 01h, 38h
0657	defb 0eh			1aa2	CLOSESTRM:	1acb	call STR_DATA2
0658	SA_NULL:			1aa2	defb "K"	1ace	ld a, b
	ld a, b			1aa3	defb CLOSE_K - \$	1acf	or c
0659	or c			1aa4	defb "S"	1ad0	jr z, OPEN_1
065a	jr z, SA_DATA			1aa5	defb CLOSE_S - \$	1ad2	ex de, hl

065c	ld bc, 000Ah	1a72	add hl, de	1aa6	defb "P"
065f	SA_NAME:	1a73	jr c, CLOSE_1	1aa7	defb CLOSE_P - \$
	push ix	1a75	ld bc, STRMS_END	1aa8	defb "N"
0661	pop hl	1a78	add hl, bc	1aa9	defb CLOSE_N - \$
0662	inc hl	1a79	ld c, (hl)	1aaa	defb "T"
0663	ex de, hl	1a7a	inc hl	1aab	defb CLOSE_T - \$
0664	ldir	1a7b	ld b, (hl)	1aac	defb 00h
0666	jr SA_DATA	1a7c	CLOSE_1:	1aad	CLOSE_N:
0668	USE_STRM_DO:		ex de, hl		call CLOSE_N_DO
0668	call SET_STREAM	1a7d	ld (hl), c	1ab0	CLOSE_T:
066b	SA_DATA:	1a7e	inc hl	1ab0	CLOSE_P:
	rst 18h	1a7f	ld (hl), b	1ab0	CLOSE_S:
		1a80	ret	1ab0	CLOSE_K:
		1a81	CLOSE_2:		pop hl
			push hl	1ab1	ret
		1a82	ld hl, (CHANS)	1ab2	STR_DATA:
		1a85	add hl, bc	1ab2	call FIND_INT1
		1a86	inc hl	1ab5	cp 010h
		1a87	inc hl	1ab7	ret c
		1a88	inc hl	1ab8	REPORT_O2:
		1a89	ld c, (hl)	1ab8	rst 8
		1a8a	ex de, hl		<i>; O Invalid stream</i>
		1a8b	ld hl, CLOSESTRM	1ab9	defb 17h
		1a8e	call INDEXER	1aba	STR_DATA2:
		1a91	jr nc, CLOSE_3	1aba	call STR_DATA
		1a93	ld c, (hl)	1abd	STR_DATA1:
		1a94	ld b, 000h	1abd	add a, a
		1a96	add hl, bc	1abe	add a, 016h
		1a97	jp (hl)	1ac0	ld l, a
		1a98	CLOSE_3:	1ac1	ld h, 05ch
			ld hl, 0005h	1ac3	ld c, (hl)
		1a9b	add hl, de	1ac4	inc hl
		1a9c	ld a, (hl)	1ac5	ld b, (hl)
		1a9d	inc hl	1ac6	dec hl
		1a9e	ld h, (hl)	1ac7	ret
		1a9f	ld l, a	1ac8	OPEN:
		1aa0	jp (hl)		rst 28h
		1aa1	jp (hl)	1ac9	defb 01h, 38h
		1aa2	CLOSESTRM:	1acb	call STR_DATA2
		1aa2	defb "K"	1ace	ld a, b
		1aa3	defb CLOSE_K - \$	1acf	or c
		1aa4	defb "S"	1ad0	jr z, OPEN_1
		1aa5	defb CLOSE_S - \$	1ad2	ex de, hl

1ad3	ld	hl, (CHANS)	1b0d	pop	bc	1b40	pop	bc	1b87	dec	bc
1ad6	add	hl, bc	1b10	ld	hl, (OPEN_X)	1b41	ld	c, a	1b88	bit	7, b
1ad7	inc	hl	1b11	ld	a, h	1b42	MOVE_LOOP:		1b8a	jp	nz, REPORT_B1
1ad8	inc	hl	1b12	or	l	1b42	ld	a, c	1b8d	inc	bc
1ad9	inc	hl	1b14	jr	z, REPORT_F	1b43	push	bc	1b8e	ld	(BAUD), bc
1ada	ld	a, (hl)	1b15	jp	(hl)	1b44	call	CHAN_OPEN	1b92	ret	
1adb	ex	de, hl	1b15	OPENSTRM:		1b47	pop	bc	1b93	VAL_REST:	
1adc	cp	"K"	1b16	defb	"K"	1b48	MOVE_READ:		1b93	inc	de
1ade	jr	z, OPEN_1	1b16	defb	OPEN_K - \$	1b48	call	INPUT_AD	1b94	dec	bc
1ae0	cp	"S"	1b17	defb	"S"	1b4b	jr	c, MOVE_WRITE	1b95	call	STK_STO
1ae2	jr	z, OPEN_1	1b18	defb	OPEN_S - \$	1b4d	jr	z, MOVE_READ	1b98	ld	b, 01dh
1ae4	cp	"P"	1b19	defb	"P"	1b4f	ret		1b9a	rst	28h
1ae6	jr	nz, REPORT_O2	1b1a	defb	OPEN_P - \$	1b50	MOVE_WRITE:		; VAL		
1ae8	OPEN_1:		1b1b	defb	"N"	1b50	push	af	1b9b	defb	1dh, 38h
	call	OPEN_2	1b1c	defb	OPEN_N - \$	1b51	ld	a, b	1b9d	ret	
1aeb	ld	(hl), e	1b1d	defb	"T"	1b52	push	bc	1b9e	CAT_ETC:	
1aec	inc	hl	1b1e	defb	OPEN_T - \$	1b53	call	CHAN_OPEN		jp	REPORT_O2
1aed	ld	(hl), d	1b1f	defb	0	1b56	pop	bc			
1aee	ret		1b20	OPEN_K:		1b57	pop	af			
1aef	OPEN_2:			ld	e, 001h	1b58	rst	10h			
	push	hl	1b22	jr	OPEN_END	1b59	jr	MOVE_LOOP			
1af0	call	STK_FETCH	1b24	OPEN_S:		1b5b	FORMAT:				
1af3	ld	a, b		ld	e, 006h		call	STK_FETCH			
1af4	or	c	1b26	jr	OPEN_END	1b5e	ld	a, (de)			
1af5	jr	nz, OPEN_3	1b28	OPEN_P:		1b5f	and	0dfh			
1af7	REPORT_F:			ld	e, 010h	1b61	cp	"N"			
1af7	rst	8	1b2a	OPEN_END:		1b63	jr	z, FORMAT_N			
; F Invalid file name			1b2a	dec	bc	1b65	cp	"T"			
1af8	defb	0eh	1b2b	ld	a, b	1b67	jr	z, FORMAT_T			
1af9	OPEN_3:		1b2c	or	c	1b69	jr	REPORT_F			
	push	bc	1b2d	jr	nz, REPORT_F	1b6b	FORMAT_N:				
1afa	ld	a, (de)	1b2f	ld	d, a	1b6b	call	VAL_REST			
1afb	and	0dfh	1b30	pop	hl	1b6e	call	FIND_INT1			
1afd	ld	c, a	1b31	ret		1b71	ld	(STATION), a			
1afe	ld	hl, OPENSTRM	1b32	OPEN_N:		1b74	ret				
1b01	call	INDEXER		jp	OPEN_N_DO	1b75	FORMAT_T:				
1b04	jr	nc, OPEN_4	1b35	OPEN_T:		1b75	call	VAL_REST			
1b06	ld	c, (hl)		ld	e, 015h	1b78	rst	28h			
1b07	ld	b, 000h	1b37	jr	OPEN_END	1b79	defb	34h, 00h, 40h, 42h,			
1b09	add	hl, bc	1b39	MOVE:		03h, 76h, 01h, 05h, 38h					
1b0a	pop	bc		call	STR_DATA	1b82	call	FIND_INT2			
1b0b	jp	(hl)	1b3c	push	af	1b85	dec	bc			
1b0c	OPEN_4:		1b3d	call	STR_DATA	1b86	dec	bc			

Na, ez már sokkal kiforrottabb kód, mint a közel használhatatlan stubok, amik a végleges ROM-ban vannak, de még így sem teljes.

A CLOSE még mindig elszáll nemlétező streameknél, de ebben az esetben a tervezett működés teljesen világos, ellentétben a végleges ROM-ban lévő verzióval. Az általam CLOSE_3-nak elnevezett rutin bináris patchelés nyomait mutatja, ezen a részén biztosan aktívan dolgoztak. Ellenben a végleges ROM-mal, a csatorna táblázat az ismert lezárási procedurákkal rendesen terminálva van. Ismeretlen csatornák esetén a lezáró procedurák címeit a CLOSE_3 közvetlen a csatorna leíróból olvassa ki, ami nyilvánvalóan a megfelelő absztrakció. Ezzel elkerülhető lett volna, hogy a ZX Interface 1 és társai trappeljék ezt a részét a ROM-nak. Az egyetlen probléma a nem hozzárendelt csatornák detektálása

és kezelése.

Az STR_DATA-n és a hozzátartozó rutinokon, amik segítenek megtalálni a stream leíró, kicsit finomítottak, ami egy érezhető előrelépés a végleges ROM-ban találhatóval szemben.

Az OPEN-t befejezték. Ezt a verziót kellett volna kiadni a végül kiadott helyett. Ha a gép nem találja a csatorna betűjét az ismert nyitó procedurák táblájában akkor egy rendszerváltozót olvas ki (OPEN_X-nek neveztem el a kódban) és ezt használja, ha be van állítva. Egyébként F invalid file name hibát dob. Sajnos a pót-ROM-ok, mint pl. az én ZX82-m nem tudják ezt a megoldást használni, mert inkompatibilis lenne sok létező szoftverrel és hardverrel.

Az "N" (LAN) és "T" (RS232) csatornáknak vannak megnyitó és bezáró procedurái. Az "N" megnyitó rutinja egy távoli station számot vár az N betű után az OPEN # string paraméterében; ezzel szemben a ZX Interface 1-ben ez külön paraméter.

A MOVE utasítás, ami a végleges ROM-ban stubolva van, két, vesszővel elválasztott string paramétert vár. Ezt egy logikusabb utasítássá alakították, ami két numerikus paramétert vár, vesszővel és hash-sel elválasztva: MOVE #from, #to és streamek közt mozgat adatot EOF-ig. Nagyon hasonló a ZX Interface 1 MOVE #from TO #to formátumához, leszámítva az elválasztó tokenet. Még a BREAK kezelésének hiánya is ugyanaz.

A FORMAT, ami egy string paramétert vár, szintén stubolva van a végleges ROM-ban. Itt viszont az RS232 baudját és a LAN interface station számát tudja állítani. Mindkét esetben a stringen belül, ami a csatorna azonosító szám után jön. A ZX Interface 1-ben és a ZX Spectrum 128k-ban ezek extra numerikus paraméterei a FORMAT-nak. A VM bájt kódban lévő számítás a baud értékét egy időzítő számlálóra alakítja.

Valószínűleg nem túl pontos és nincs benne ellenőrzés de néhány értéknél működhet. Hardver hiányában nehéz tesztelni. Az OPEN-nel szemben a FORMAT nem teszi lehetővé az engedélyezett csatornák kiterjesztését, úgyhogy ez nincs teljesen készen.

A szintaxis tábla megfelelő bejegyzései módosítva lettek, ezért ezek az utasítások működnek:

1f12	P_FORMAT:
1f12	defb 0ah, 00h
1f14	defw FORMAT
1f16	P_MOVE:
	defb 06h, "#", 06h, 00h
1f1b	defw MOVE
1f1d	P_ERASE:
	defb 0ah, 00h;
1f1f	defw CAT_ETC
1f21	P_CAT:
	defb 00
1f22	defw CAT_ETC

A "T" és "N" csatornák plusz service rutinjai nem annyira érdekesek és valószínűleg nem is működnek. Soha

nem voltak igazi hardveren tesztelve, mivel soha nem létezett ilyen hardver.

A ROM többi része megegyezik a végleges ROM-mal, kivéve az áthelyezett címeket és a rendszerváltozók offszetjeit.

Mindent összevetve mondhatjuk, hogy a firmware fejlesztés még hetekkel le volt maradva a megjelenéstől. Ha befejezték volna az elkezdett dolgokat a ZX Spectrum egy sokkal jobb és időtállóbb számítógép lett volna. Az, hogy ettől sikeresebb vagy hosszabb életű lett volna-e vagy ellenkezőleg, a több, mint egy hónapos késés miatt hamarabb feledésbe merült volna, az már sosem derül ki.

Ez a szoftver archeológia lehetetlen lett volna a következő személyek nélkül:

- John Grant, aki a Centre for Computing History-nak adományozott egy prototípust
- [Centre for Computing History](#), akik publikálták a ROM tartalmát
- John Grant és Steven Vickers, akik történelmi és technikai kontextust nyújtottak privátban
- Ian Logan és Frank O'Hara, akik úttörő munkát végeztek a végleges ROM visszafejlesztésében
- Andrew Owen, aki egyéb történelmi és technikai információkkal látott el
- a szüleim: Burda Olga és Nagy András, akik egy ZX Spectrum+-t vettek nekem 1986-ban
- a feleségem: Nagy Valentina, aki támogatott a munkám során

Nagy Dániel

Az eredeti cikk angolul jelent meg ['Inside the Spectrum Prototype ROM'](#) címmel, melyet Nagy Dániel engedélyével fordított magyarra Barabás Péter (z0d).

