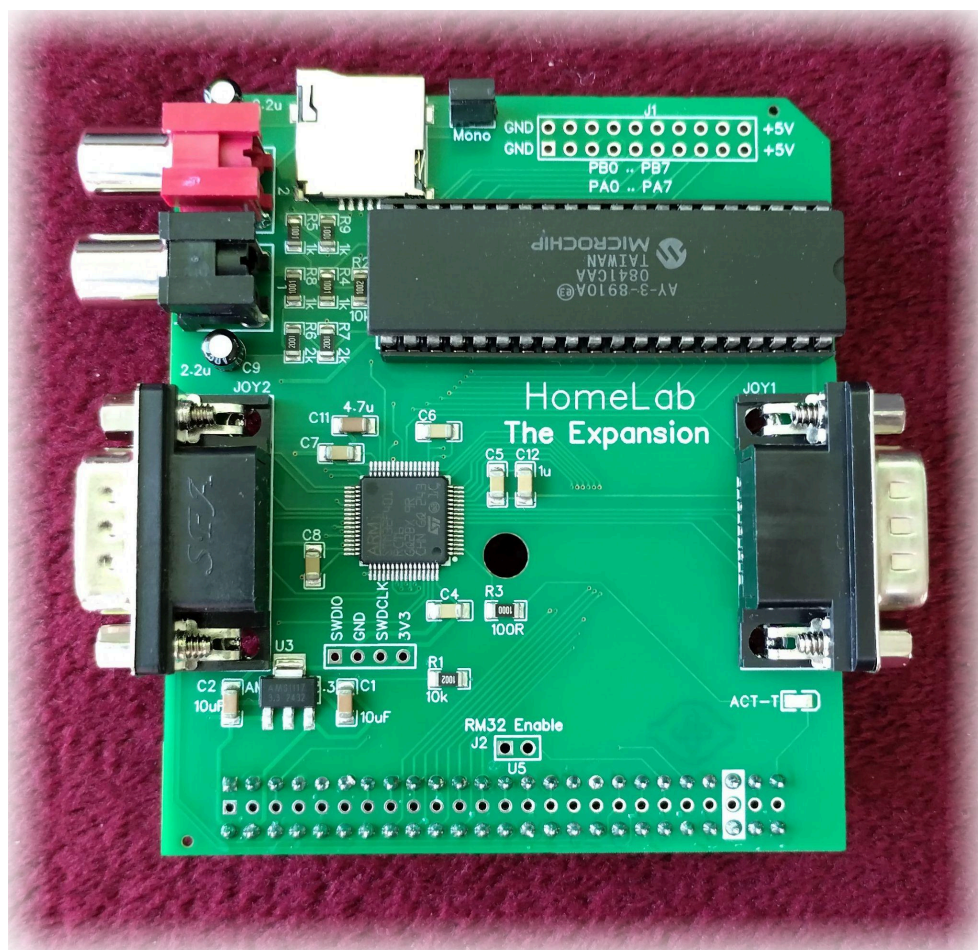


HomeLab 3 - The Cart

Felhasználóknak és programozóknak

2. változat



Tartalomjegyzék

Tartalomjegyzék.....	2
The Cart - mit tud és mit nem.....	4
Képességek.....	4
Korlátok.....	5
PROM.....	5
Rövid technikai kitekintő.....	6
Felhasználóknak.....	6
Belépési pontok a CART-ba.....	6
A menü.....	7
Parancsértelmező.....	8
Paraméterezés.....	8
Parancsok.....	8
Ments meg - avagy SAVE.....	9
Menten megszakadok!.....	10
Önteszt.....	10
Programozóknak.....	11
CART API.....	11
Hívható API szubrutinok.....	11
LOAD HTTP FILE.....	11
SAVE HTTP FILE.....	11
Alacsony szintű I/O Cart parancsok.....	12
A parancslista.....	13
\$00, MOUNT.....	13
\$01, OPEN DIR.....	13
\$02, READ DIR ENTRY.....	13
\$03, CLOSE DIR.....	14
\$04, OPEN HTP FILE.....	14
\$05, READ HTP BLOCK.....	14
\$06, CLOSE FILE.....	15
\$07, GET CURRENT PATH.....	15
\$08, SET CURRENT PATH.....	15
\$09, SET AND START TIMER.....	15
\$0A, STOP TIMER.....	16
\$0B, ACK TIMER.....	16
\$0C, OPEN BINARY FILE FOR READ.....	16
\$0D, READ BINARY BLOCK.....	17
\$0E, SEEK BINARY FILE.....	17
\$0F, OPEN BINARY FILE FOR WRITE.....	17
\$10, WRITE BLOCK TO BINARY FILE.....	17
\$11, DELETE FILE OR DIRECTORY.....	17
\$12, RENAME FILE OR DIRECTORY.....	18

Joystick kezelés.....	18
AY-3-8910 kezelés.....	18
Függelék.....	19
Példa: egy rövid bináris file beolvasása, assembly-ben.....	19
Példa: egy rövid BASIC program SD kártya listázására.....	20
Példa: SAVE_HTP_FILE hívása assembly-ben (RAM-ban!).....	20
Köszil.....	22

The Cart - mit tud és mit nem

Képességek

A HomeLab3 -hoz készült bővítőkártya a magyar HomeLab 3 és az RM32 mikroszámítógépek tulajdonosainak az életét megkönnyítendő készült el. A cart alapja egy STM32F401 típusú mikrokontroller, az alább felsorolt funkciók mindegyikét ez látja el.

Nem kell többet magnóval, telefonnal és kábelekkel bajlódni, a programokat (.http kiterjesztésű magnó állomány fájlokat) egy microSD kártyáról tudja beolvasni. Ez az alap funkció, emiatt készült a cart. A többi csak a ráadás.

Mik is ezek a *ráadások*?

- A cart nem csak olvasni tudja az microSD kártyát, de írni is tud rá. Pl. egy megírt BASIC programot is ki lehet menteni (akár felülírva az előző változatot).
- 2 joy csatlakozó portot kínál azaz akár 2 játékos módra is lehetőséget biztosít (ehhez persze megfelelő játék is kell).
- A kártya két RCA csatlakozó formájában sztereó hangkimenettel rendelkezik, amihez a hangot egy AY-3-8910 -es PSG chip állítja elő. Ez a chip rendelkezik 2db I/O porttal is melyek elérhetőek a kártyán (megfelelő csatlakozó beültetése után). A hang chip 1.5MHz -es órajelen üzemel. Ez az órajel független a HL3 -as órajelétől
- A két RCA kimenet sztereó hangot ad: az AY chip 1. csatornája 'bal' oldalt (fekete, alsó RCA csatlakozó), a 2. csatornája középen, míg a 3. csatorna a jobb oldali (piros, felső) RCA kimeneten szól. Lehetőség van monósításra is, ehhez egy jumpert kell zárni. A hangkeverés sajátosságai miatt a bal és a jobb csatorna hangja is hallatszik az ellenkező oldali csatornán, csak sokkal halkabban.
- Ezeken túl a cart periodikus megszakítás-kérélmeket tud küldeni a Z80 -as CPU-nak. A megszakítás kérelem periódusideje 16 bites előjel nélküli számon, 100 µs felbontásban állítható. A leghosszabb periódus kb 6.5 másodperces.
- Programozóknak egy ún. API-t biztosít, aminek segítségével (új) HL3 -as programok számára elérhetővé válnak a file-műveletek, HTP file-ok írása/olvasása és a megszakítás kezelés.
- A cart úgy lett kialakítva, hogy fizikailag kapcsolható Nagy Attila RM32 -es alaplapjához. Ez egy modernizált HomeLab3 -as, 32kB RAM memóriával (ReMake with 32kB RAM -> RM32). Ehhez az RM32 jumpert kell zárni (FIGYELEM: 'rendes' HomeLab3/4 esetén TILOS a jumpert rövidre zárni!)
- Elkészült egy ún. header nyáklap, ami Kollár Zoltán féle HL5 -ös alaplapján teszi elérhetővé a szükséges érintkezőket, így azon lap tulajdonosai számára

is elérhető az SD kártya olvasás (a HL5 fix, 32kB -os memória konfigurációjával)

Úgy vélem eléggé impozánsra sikerült ez a lista, szóval a nevet illetően lehet nagyképűnek tartani, de van mire büszke legyek! 😊

Korlátok

A cartridge Aircomp és HomeLab2 -es gépekkel **nem kompatibilis**. Kollár Zoltán által készített HomeLab5 -ös géppel csak a HL5 adapter közbeiktatásával együtt.

A kártya használatához a HL3/4 gépekben a PROM -ot át kell programozni (lásd alább).

A cartridge nem tud olyan programot betölteni, futtatni, aminek 36kB -nál több memóriára van szüksége (fájlméret>36800 byte). Szerencsére a HL3 -ra megjelent programok túlnyomó többségének elegendő a 32kB RAM is.

A cartridge elvárja a ROM1 és ROM2 rendszer ROMok meglétét a \$0000-\$1FFF címen. (Csak a hexadecimális számok használatához szükséges a bővített BASIC, a cart elindítható ezen bővítés nélkül is).

A cartridge használatakor a ROM5 IC foglalatnak üresnek kell lennie a HL3/4 alaplapon.

A cart bekapcsoláskor **nem** indul el, ahhoz a felhasználó közreműködése szükséges (ilyen szolgáltatást a HL3 -as nem nyújt - a ROMok lecserélése nélkül).

A cart menüje hibát jelez és kilép, ha a micro SD kártya nem elérhető vagy használat közben elérhetetlenné válik. A menü csak akkor működik helyesen, ha közben az SD kártya végig a foglalatban van.

Az SD kártyán az első partíciót FAT32 vagy FAT16 fájlrendszerre kell formázni, a betöltendő fájloknak itt kell elhelyezkedniük.

Figyelem! A cartridge-et minden esetben kikapcsolt HomeLab-as géphez csatlakoztassuk, ellenkező esetben sérülhet a gép és a cart is. Működés közben - ha egy mód van rá - ne mozgassuk a cart-ot az élcsatlakozón (pl ne csatlakoztassunk joy -t, audio kábelt, stb...)

PROM

A kártya működéséhez olyan PROM kell, aminek hatására a HL3 egyetlen memória lapon éri el a rendszer memóriát (ROM és RAM) és a \$D000-\$DFFF tartományban a

ROM5 tartalma látható. Ebből következik, hogy legfeljebb a \$4000-\$CFFF -ig terjedő terület lehet RAM a HL3 a carttal együtt, ezért a gép max kb 36kB memóriát láthat összefüggő területként, ami a BASIC számára is elérhető.

RM32 -es gépek esetében is szükség van a PROM átprogramozására, mivel a 'gyári', Attila által kiadott PROM még nem volt felkészítve a cartridge fogadására. HL5 -ös gépeknél nincs PROM, ott fixen be van állítva a memória és tervezett adapter megoldja a memória-elérési gondokat.

Rövid technikai kitékintő

A cartridge ROM és RAM szolgáltatást is nyújt a HomeLab gépnek. Amikor az a ROM5 területről (helyes PROM kódolás esetben a \$D000-\$DFFF memória tartomány) olvas vagy oda ír, akkor a HL gép engedélyezi a ROM5 IC foglalatban a /CE jelet. Ez az engedélyező jel ki van vezetve a hátsó bővítő csatlakozóra, ezt tudja a cartridge felhasználni a saját céljaira, de csak akkor, ha a foglalatban nincs ROM/RAM IC, mert akkor az ütközést eredményezne az adatbuszon - és vélhetőleg fizikai károsodást is az IC-kben.

Az emulált ROM-ban található a cart menüje, a HTP file-ok olvasása és írása, a \$D003-as címen található parancsértelmező és a hozzá tartozó parancsok, az interrupt kezelő példaprogram, a memória teszt program illetve a 3 BASIC nyelven írt demo program is.

Úgy szerettem volna, hogy a cartridge mind az egy- és a két lapos szervezésű HomeLab3 -as gépekkel együtt tud működni, de sajnos a fejlesztés közben ezt el kellett vessem.

A BASIC POKE utasítás tud címezni a második lapon található címeket, de a BASIC CALL csak az első lapon található címekre tud ugrani.

A cart menüjébe egy sima CALL utasítással nem lehet belépni, ha a cartridge a 2. lapon elérhető csak. Ezt egy rövid assembly rutinnal ki lehet ugyan kerülni, de az nagyon körülményessé tette volna a menü elérését - a cart alapfunkcióját (arról nem is beszélve, hogy elbonyolította volna a már egylaposra elkészült cart rutinok átírását).

Felhasználóknak

Belépési pontok a CART-ba

Bekapcsolás után a HomeLab3 -as számítógépen semmi jelét nem szabad lássuk a cart jelenlétének: a cart-on a LED békésen pislog - jelezve, hogy az kész az utasítások fogadására - de máskülönben a képernyőn csak a kurzor villog a HomeLab3 -as bejelentkező képernyője alatt.

A cart programjait minden esetben egy BASIC CALL utasítással kell meghívunk. A v0.12 -es firmware revízióban 4 futtatható program található a cartban, ezek belépési pontjai: \$D000, \$D003, \$D006 és \$D009 (decimális jelöléssel rendre: 53248,

53251, 53254 és 53257), és a \$D00C-\$D00D és a \$D00E-\$D00F címeken pedig a HTP file betöltő és kimentő API rutinjainak a címe (ezekről később lesz szó).

A menü

A cartridge menüjébe a `CALL $D000` utasítással léphetünk be (amennyiben a HL3 -ban nem áll rendelkezésre a BASIC bővítés úgy a decimális értéket - itt 53248 - kell használni).

Ez a BASIC utasítás rajta is van a cartridge hátoldalán, hogy a nyájas felhasználó ezen felhasználói kézikönyv olvasása nélkül is be tudja tölteni kedvenc programjait.

A parancs hatására egy fájl doboz jelenik meg a képernyő bal oldalán, benne az SD kártyán található könyvtárak és fájlok neveivel utóbbiak méretével is. A cart 8.3 -as fájlnev konvenciót követ, hosszabb nevű fájlokat lehetőség kerüljük el. Az MSDOS -ban megszokott ~ jellel való rövidítés itt is jelen van, de a ~ jel kódja a HomeLab3 -on az 'Ó' betű. Ez elsőre kissé furán fog hatni a listában.

Figyeljük meg a fájldoboz két szélén található > és < jeleket! Ezek jelzik az adott sor kijelölését. A kijelölést a nyilakkal tudjuk változtatni: LE és FEL nyilakkal egyet tudunk lefelé és felfelé mozogni a listában, míg a JOBB és BAL nyilakkal egy teljes oldalt tudunk mozogni előre illetve vissza.

Ha egy könyvtáron állunk, akkor a RETURN billentyűvel tudunk a könyvtárba beváltani, (.. a szülőkönyvtárt jelöli).

Ugyanígy a CR (RETURN) billentyűvel lehet a kijelölt .HTP állományt betölteni. A program betöltés előfeltétele a .HTP kiterjesztés és a helyes HTP fájl header megléte. Bármelyik hiánya betöltéskor hibaüzenetet von maga után.

Amint a program betöltődött a képernyő bal felső sarkában megjelenik a 'Program betöltve' üzenet, majd elindul a program (ha a program rövid, vagy nem csinál semmit, akkor könnyen lehet, h betöltésre közvetlenül a fenti üzenet alatt egy Ok szócskát is látunk a képernyőn, hiszen a lefutott BASIC program után a HomeLab3 ROM azt írja ki).

A kimentett (basic) HTP fájlok sajátja, hogy tartalmazzák a rendszer RAM területet is. Ez azt is eredményezi, hogy egy 64kB -os (2 lapos) HL3 gépen kimentett program nem tud elindulni, vagy hibásan futna a cart alapjául szolgáló 32 vagy 36kB -osra konfigurált HL3 -as gépen.

Nos, a cart betöltés után okoskodik kissé és a valóban rendelkezésre álló memóriát beállítja a betöltött program indítása előtt (a rendszer RAM területen), ezért a programok - nagy része - el fog indulni. Több program fog elindulni, mint sima, kazettás BASIC betöltés esetén.

Azon programok viszont, amik *valóban* 64kB RAM-ot igényelnek (vagy simán több, mint 36kB RAM-ot az első lapon), nos azok a programok sajnos nem fognak működni.

A Menü futása során használ RAM-ot. Ez a már a memóriában lévő, rövidebb BASIC programok esetében nem okoz gondot, de ha a program 'belelóg' a \$6000 cím fölötti RAM területre, akkor a Menü futtatása során a HL3 memóriájában lévő program sérülni fog. (A Menüt érdemes úgy futtatni, h előtte egy NEW basic utasítást futtatunk, a későbbi félreértések elkerülése érdekében.)

A \$6000 -es cím fölötti RAM területen a Menü a saját változóit tárolja, \$600D-től kezdve pedig az egyes könyvtárak fájl-lista elemeit. Egy listaelem 16 byte helyet foglal, így kiszámolható, hogy egy 16kB -os gép csak olyan könyvtárat tud beolvasni, amiben legfeljebb 511 bejegyzés található. 32 és 64kB -os gépek esetén ez a határ jelentősen kitolható - de milyen könyvtár már az, amiben 100-nál több fájl található?!

Parancsértelmező

```
Ez itt a HomeLab3-The Cart kártya parancsértelmező leírása v1.01

A parancsot mindig a
CALL $D003,"<P>(<file>)"(<startcim>,<végcim>)
formában kell megadni, ahol <P> a parancs, a (< >) közötti
kifejezések opcionálisak.

A lehetséges <P> parancsok:

I vagy M - az SD kártya inicializálása
P - az aktuális könyvtár lekérdezése
$ - az aktuális könyvtár kilistázása
H - ez a HELP (vagy a $D003 paraméter nélkül)
D <file> - file vagy Üres könyvtár törlése
R <file1> <file2> - <file1> átnevezése <file2>-re
C <path> - könyvtár váltás
L <file> - file betöltése
B <sorszám> - BASIC példaprogram (1-3) betöltése
S <file> - HTP file mentése.

Mentéskor ha a file neve + jellel kezdődik, akkor a felülírás
engedélyezett.
Start- és végcim csak az S (SAVE) parancsnál adható meg!
A start- és a végcim megadható decimális(4096) és hex($CAFE)
formában is!
A file nevet az S parancsnál .HTP nélkül kell megadni!

Ok
```

A \$D003-as belépő ponton egy többfunkciós parancsértelmező rutin található. Attól függően, hogy hogyan paraméterezzük a hívást, más-más parancsot tudunk futtatni.

Ha nem paraméterezzük a \$D003 -as hívást (sima CALL \$D003 hívás esetén) akkor egy képernyőnyi segítséget, ún. HELP üzenetet kapunk.

Paraméterezés

A parancsértelmezőt az alábbi módon kell meghívni:

CALL \$D003, "<P>[<param>]" [, <kezdőcím>, <végcím>]

- a belépési pont a \$D003 (decimálisan 53251)
- a [] közötti kifejezések/paraméterek nem kötelező elemei a parancs hívásának
- kezdő- és végcímet csak a SAVE parancsnál lehet megadni
- a <param> paraméter parancstól függően lehet fájlnev (.HTP kiterjesztéssel vagy anélkül), lehet könyvtár név vagy a betöltendő BASIC demó program sorszáma is

Egy példa egy fájl kimentésére:

CALL \$D003, "S TUZOLTO"

Parancsok

Az alábbi lehetséges parancsok adhatóak meg (a <P> helyett):

- I vagy M
SD kártya (újra-) inicializálása, becsatolása. Ilyenkor az aktuális munkakönyvtár újra a gyökérkönyvtár lesz. Ezt a parancsot egy esetleges SD kártyacsere után érdemes meghívni (azt nem veszi észre a CART). Mivel a parancsok hívásakor a parancsértelmező megvizsgálja, hogy RESET óta volt-e már inicializálva az SD kártya és ha nem, akkor automatikusan inicializálja, becsatolja az SD kártyát, az egyes parancsok előtt nem szükséges kiadni az INIT parancsot.
Példa: CALL \$D003, "I"

- H vagy paraméter nélküli parancs hívás
HELP kiírása a képernyőre. Gyakorlatilag a parancsértelmező használati utasítása kerül a képernyőre az összes lehetséges parancs rövid leírásával, ez a HELP látható a fejezet elején. Ez a parancs SD kártya nélkül is működik (nincs szükség SD initre).
Példa: CALL \$D003, "H"

- \$ Könyvtár listázása
Direktori kiírása a képernyőre a jelenlegi munkakönyvtár tartalmáról. Ha a listázás közben lenyomjuk az F1 billentyűt, akkor lassul a kiírás, ha az F1 mellett az F2 billentyűt is, akkor megszakad a listázás és a parancsértelmező kilép.
Példa: CALL \$D003, "\$"

- P A jelenlegi munkakönyvtár kiírása a képernyőre
Példa: CALL \$D003, "P"
- C <path>
A jelenlegi munkakönyvtár megváltoztatása. Mivel az alacsony szintű API csak teljes elérési útvonalat és a .. fogadja el, relatív útvonal megadása esetén ez a parancs először lekérdezi a jelenlegi munkakönyvtárat és ahhoz illeszti a megadott relatív útvonalat. Ha lehet, akkor kerüljük a .. path-on belüli használatát!
Példa: CALL \$D003, "C /jatek/a"
- D <file>
A megadott file vagy alkönyvtár (amennyiben az üres) törlése.
Példa: CALL \$D003, "D WOW.HTP"
- R <file1> <file2>
Fájl vagy könyvtár átnevezése, <file1>-ről <file2>-re. A két fájl- vagy könyvtárnevet SPACE karakter választja el egymástól, így a parancsértelmező olyan fájlokat nem tud átnevezni, aminek nevében van SPACE karakter is.
Példa: CALL \$D003, "R WOW.HTP WIZOFWOR.HTP"
- B <sorszám>
A ROMban található 3 BASIC példaprogramok egyikét (cart felismerés, joy-kezelés, polifónikus zongora az AY-3-8910 csippel) betölti és elindítja.
Ez a parancs SD kártya behelyezése nélkül is működik (nincs szükség SD initre). Rossz paraméterezés esetén egy rövid help üzenet jelenik meg arról, hogy mivel lehet paraméterezni a parancsot (1, 2 vagy 3) és hogy az egyes paraméterek milyen BASIC példa programokat jelentenek (1 - kártya felismerő, 2 - joy lekérdező, 3 - polifónikus zongora AY-3-8910 -re)
Példa: CALL \$D003, "B 2"
- S <file>
File mentése. Ennek a parancsnak a kifejtése historikus okokból egy külön fejezetben található. Ez az egy parancs, ahol a fájlnev .HTP nélkül értendő és további paraméterek is adhatóak.
- L <file>
File betöltése. Ezzel a paranccsal van lehetőség arra, hogy a memória különböző helyeire, több .HTP file-ból adatot, programot töltsünk be.

Amennyiben a program a BASIC területre tölt (a \$4018 és \$4019 címek tartalma megváltozik), akkor a cart a betöltött programot el is indítja.

Példa: `CALL $D003, "L WOW.HTP"`

Ments meg - avagy SAVE

Az `S` (SAVE) parancsot meghívva lehetőségünk van egy .HTP file létrehozására (vagy felülírására) a memóriából. A parancs formátuma:

`CALL $D003, "S [+]<fájlnév>"[, kezdőcím, végcím]`

Ahol:

- a belépési pont a \$D003 (decimálisan 53251)
- vesszővel mögötte dupla-idézőjelek között kell lennie a parancsnak (`S`) és a fájlnevnak, pl: `"S test-program"`. Amennyiben egy már létező fájlt kívánunk felülírni, akkor a fájlnevünket `+` jellel kell kezdeni. Ebből az adódik, hogy ha azt szeretnénk, hogy egy fájl `+` jellel kezdődjön, akkor két `db +` jellel kell kezdeni a file nevét kimentéskor! Javaslom: inkább kerüljük az ilyenféle különcködést
- az SD kártyán a fájl neve a megadott fájlnev első 8 karaktere lesz (amennyiben meg van adva annyi), a kiterjesztés pedig .HTP, továbbá a névben a space karakterek `_` (underscore) karakterre cserélődnek
- a .HTP fájlban a program neve a megadott fájlnev első 32 karaktere lesz
- a fájlnev megadása kötelező
- a kezdő- és végcím megadása nem kötelező
 - ha nem adjuk meg, akkor a cart a jelenlegi BASIC programot menti ki a memóriából. Kezdőcím a \$4016 lesz, végcím pedig a \$4018 által mutatott cím (ez a BASIC adatterület kezdete)
 - ha megadjuk, akkor a cart a megadott címek közötti memória területet menti ki a .HTP file-ba. A kezdőcím igen, végcím már nem lesz része a kimentett file-nak
- a kezdő- és végcím megadható decimális és hexadecimális formában is. Értékük 0 és 65535 (\$0 - \$FFFF) között lehet. Ebből adódik, hogy ha valaki a \$FFFF címet is ki akarja menteni (mondjuk utolsó byte-ként), akkor \$0 címet kell megadnia végcímként (ezeket a számokat nem a ROM olvassa fel, ROM bővítéstől függetlenül lehet megadni tetszőleges számrendszerben)
- amennyiben kezdőcím meg van adva, úgy végcímet is meg kell adni

Kezdő és végcím nélkül az utasítás hatására egy sima BASIC `SAVE` fog lefutni, BASIC programokat így lehet kimenteni az SD kártyára.

Menten megszakadok!

Egy rövid demonstrációs program található a \$D006 -os címen (decimális alakban 53254). Ez a program egy megszakításkezelő rutint definiál, ami 25ms -onként hívódik meg (kb minden képernyő frissítéskor egyszer). Ez a képernyő első sorába ír ki majd töröl le egy, a carton található teszt szöveget, miközben a kurzor villog a képernyőn. A kurzor mozgatható közben a nyilakkal. Mivel a program nem túl bonyolult, ezért érdemes a \$D006 -os címet úgy meghívni, hogy a képernyő első sora teljesen üres. Én szóltam.

Valamilyen okból kifolyólag a megszakítások és a BASIC értelmező nem tud egyszerre futni: ez egy eléggé sérülékeny - és nem túl jó - demo. Bármilyen BASIC utasítás hatására lefagy a gép.

(Nem jártam teljesen utána, de az a gyanúm, hogy a 2. lap kicsi stack területének lehet hozzá köze)

Memória teszt

A \$D009 -es címen (decimálisan 53257) egy teszt program található, ami a cart RAM-ot teszteli vég nélkül. A képernyőn 4 sornak felel meg egy teljes teszt ciklus. A program + karakterrel jelöli az éppen teszt alatt álló memória címet, * kerül a képernyőre hiba (nem egyezés) esetén, máskülönben . (pont). Semmi extra.

Egy jól működő cart esetében ez egy sima, hétköznapi képernyő pontozó program...

Programozóknak

CART API

A cartridge-el assembly-ben kétféle módon kommunikálhatunk:

- 1) a (\$D00C) és a (\$D00E) címeken definált 16 bites ugró kódok a LOAD_HTP_FILE és a SAVE_HTP_FILE cart szubrutinokra mutatnak.
- 2) alacsony szintű CART parancsokkal kommunikálni az assembly IN/OUT utasításokkal illetve a feljebb már említett \$DF00-\$DFFF ún. cart-RAM területen

Hívható API szubrutinok

LOAD_HTP_FILE

Ez a szubrutin bemenő paraméterként a HL regiszterben a betöltendő fájl nevére mutató pointert várja, null-terminated (azaz C típusú) sztringként.

A szubrutin végrehajtása közben az összes regiszter tartalma megsemmisül, ha szükség van rájuk a szubrutin hívás után, gondoskodjunk azok mentéséről.

Amennyiben a betöltés sikeres, akkor visszatéréskor CPU Z flag beállítódik, egyébként törölve lesz. A rutin nem ír a képernyőre.

A .HTP file-ban található adat a .HTP headerben meghatározott címtől lesz bemásolva a memóriában.

A microSD kártyának a hívás előtt már csatolva kell lennie.

SAVE_HTP_FILE

Ez a szubrutin bemenő paraméterként a HL, DE és BC regisztereket használja.

Amennyiben a kimentés sikeres, akkor a Z flag be lesz állítva visszatéréskor. Hiba esetén a Z flag tartalmát törli a szubrutin.

A microSD kártyának már csatolva kell lennie.

Paraméterezése:

- HL: pointer a file nevére kell mutasson. Ugyan ezt az alaprutint használja a BASIC-ből is hívható \$D003-as kód, ezért a stringet lezárhatja \$00, dupla idézőjel vagy a \$60 kódú (sorvég) karakter is. Érdemes sima \$00 -vel zárni.
- DE: tartalmazza a kimentendő terület kezdőcímét
- BC: tartalmazza a határcímet, amit már éppen nem kell kimenteni (pl. \$5000-\$5FFF között mentés esetén a DE regiszter \$5000, míg a BC regiszter \$6000 értékeket kell tartalmazzon)
- Ez az API hívás megegyezik a BASIC parancsértelmezőből meghívott SAVE rutinnal, de annyi különbség azért van, hogy az API hívás nem ír semmit a képernyőre.

Alacsony szintű I/O Cart parancsok

Az assembly OUT utasítással parancsot lehet küldeni a cartnak, a cart-RAM területen esetleg paramétereket beállítani a parancshoz, az assembly IN utasításokkal pedig ellenőrizni a parancs végrehajtását, esetleg a cart-RAM területen megtalálni egy sikeresen lefutott parancs kimenetét, válaszát.

Minden egyes parancsot a cart vezérlő (CTRL) portjára írt OUT utasítással indítunk, arra a választ a CTRL port olvasásával - IN - kapjuk meg.

A cartridge CTRL portja: \$40.

Parancsot a cart $\$7F$ kóddal nyugtáz (jelezve, hogy megkezdte a feldolgozását), amit a CTRL port olvasásával (IN) lehet elérni. Ez minden parancsvégrehajtás egyik első utasítás a carton, azaz nagyon hamar kell tudni olvasni. A HomeLab3 -as gépen általában azonnal, de legyünk megengedőek így feltehetjük, hogy legfeljebb 16 IN utasítás után olvasnunk kell egy $\$7F$ értéket. Ha nem érkezik a $\$7F$ válasz időben, akkor baj van: vagy nincs cart, vagy lefagyott....
A $\$7F$ értéket csak egyszer adja vissza a cart.

Amíg a cart a fent nyugtázott parancsot dolgozza fel, addig a CTRL porton a $\$FF$ értéket kell olvasni.

Amint készen van a parancs végrehajtásával (fájl megnyitás, blokk kiírás, stb...), a CTRL porton egy feldolgozási státusz kódot kell lássunk. A kód 7. bitje jelzi, h baj van-e. Minden $\$80$ -nál nagyobb érték valamilyen hibát jelöl, ellenkező esetben a parancs sikeresen lefutott.

A legtöbb utasítás $\$00$ -val tér vissza sikeres végrehajtás esetén.

Vegyük észre, hogy egy parancs kiadása utáni $\$7F$ pontosan egyszer, míg a végrehajtás során az $\$FF$ egyszer sem vagy többször is 'jöhet vissza' a CTRL porton, akár végtelenségig is (programhiba esetén). Érdemes egy racionális határt szabni a várakozásra (10ms, 10sec, stb...)

A parancsok nem futhatnak egymással átfedésben. Az előző parancsnak teljesen le kell futnia (ki kell olvasni az $\$FF$ értékek utáni visszatérési értéket) ahhoz, hogy a következő parancsot el tudjuk indítani!

A cart-RAM tulajdonképpen tekinthető egy olyan RAM területnek, amit a cart és a HomeLab3 is tud írni-olvasni (jó esetben nem egyszerre). Ezen a memória területen tud a két CPU (HomeLab3 Z80 -as processzora és cart-on található ARM alapú mikrokontroller) adatot cserélni.

Megjegyzés: BASIC -ben is elérhetjük a $\$40$ -es CTRL (I/O) portot a POKE 64,x és PEEK(64) utasításokkal.

A parancslista

A be- és kimenő paraméterek minden esetben a cart-RAM -on helyezkednek el. String esetben C típusú, NULL-terminated string-et vár a parancs. Emlékeztetőül: az üres string az egy db $\$00$ byte.

16 vagy 32 bites számok esetében a legalacsonyabb helyiértékű byte van a legalacsonyabb címen (pl a $\$01F5$, 16 bites szám letárolása a $\$DF00$ címre => $\$DF00$ címen: $\$F5$, $\$DF01$ címen: $\$01$)

Könyvtár-szeparátor karakter / (forward slash - a sima per jel)

Fájl és könyvtárnév megadható abszolút vagy relatív címzéssel. Az első esetben a név / jellel kezdődik és a gyökérkönyvtárhoz képest értékelődik ki a név. Utóbbi esetben a jelenlegi könyvtárból kezdődik kiértékelődni a név.

És akkor következzenek a meghívható parancsok:

\$00, MOUNT

Ezt a parancsot az első fájlműveletünk előtt meg kell hívni. Ez inicializálja a belső struktúrákat, keresi meg a fájlrendszert az SD kártyán, stb.. Ha egy SD kártyát eltávolítottak a gépből, akkor is újra kell hívni a MOUNT parancsot. A fgv hívása után az aktuális könyvtár a gyökérkönyvtár (azaz a /).

Bemenő paraméter: -

\$01, OPEN DIR

Egy könyvtár listázása esetén meg kell 'nyitni' a szülő könyvtárat, amit ezzel a paranccsal tehetünk meg (majd egyenként lekérdezni az adott könyvtár elemeit). Ez a parancs egyúttal beállítja a megadott könyvtárat az aktuális könyvtárnak.

Bemenő paraméter:

- üres string esetén az aktuális (current) könyvtárat nyitja meg listázásra.
- a megadott könyvtárat nyitja meg
- ha csak kettő pontot adunk meg paraméterül, akkor az aktuális könyvtár szülőkönyvtára lesz megnyitva

\$02, READ DIR ENTRY

Egy már megnyitott könyvtár elemeit kérdezi le sorban, egyenként. Nyilván a könyvtárra az OPEN DIR parancsot korábban meg kellett már hívni, a listázáshoz a könyvtárat meg kellett nyitni.

Kimenő paraméter:

A \$DF00 címre másolva a FILINFO FatFS struktúra egy része.

- \$DF00: attribútum byte (\$01-read only, \$02-hidden, \$04-system, \$10-directory, \$20-archive)
- \$DF01-\$DF04: file méret, \$DF04 tartalmazza a legkisebb helyiértéket
- \$DF05- fájl/könyvtár neve, legfeljebb 12 byte (8 + 1 + 3), de alapvetően NULL-terminated sztring.

Ha a listázásnak a végére értünk, akkor a \$DF05 (fájlnev első karaktere) \$00 lesz, azaz üres sztring jön vissza fájlnev gyanánt.

\$03, CLOSE DIR

Lezárja a listázáshoz megnyitott könyvtár struktúrát és felszabadítja a listázáshoz lefoglalt erőforrásokat. Új könyvtár listázásához minden esetben le kell zárni a régi, már megnyitott könyvtárat.

\$04, OPEN HTP FILE

Egy HTP file megnyitása olvasásra. Amennyiben a megnyitandó fájl nem létezik vagy a htp header hibás, a \$04 -es fgv hibát ad vissza.

Hibakódok:

- \$81 és \$93 között a FatFS library hibakódjai (+\$80)
- \$B0: nincs meg a 128 (!) db \$00 byte a file headerben. Igen, a gyári ROM LOAD rutin is csak ennyit igényel
- \$B1: nem sikerült megfelelő mennyiségű RAM-ot lefoglalni az MCU-n belül a beolvasáshoz. Vagy a fájl túl nagy (>50kB) vagy az MCU-n található firmware kód leakel..
- \$B2: nem sikerült beolvasni a blokkheader-ben leírt blokkméretet beolvasni a fájlból
- \$B3: hibás a fájlban az első blokk checksum értéke

Kimenő paraméterek (amennyiben a megnyitás sikeres volt):

- \$DF00 - \$DF01: kezdőcíme a HL3 programnak (innenről kezdve kell bemásolni a memóriába)
- \$DF02 - \$DF03: az első blokk mérete (checksum nélkül). A beolvasott byte -ok számát a kliens programnak kell nyilvántartania
- \$DF04: A jelenlegi blokk sorszáma (1. blokk esetén: \$01)
- \$DF05: Blokkok száma a file-ban.
- \$DF06: Az első blokk checksum értéke

\$05, READ HTP BLOCK

Egy már megnyitott HTP file-ból olvas be adatot.

A HTP file megnyitása során az első blokk adatai kerülnek beolvasásra, 256 byte-onként, ameddig tart az éppen olvasás alatt álló blokk.

Amennyiben a fájl több blokkból áll, akkor az első blokk teljes beolvasása után nem a következő blokk byte-jai következnek (azonnal), hanem a következő blokk metaadatai pont úgy, ahogyan az OPEN HTP FILE során, annyi különbséggel, hogy a jelenlegi blokk száma itt már nem 1 lesz, hanem az éppen aktuális blokk sorszáma (2, 3, stb...). A \$DF05 mindig ugyanazt az értéket fogja tartalmazni: a fájlban található blokkok számát. Ez nyilván nem változik olvasás során.

Kimenő paraméterek:

(Ha az adott blokknak még vannak valid adatbyte-jai)

- \$DF00- a beolvasott byte-ok

(Ha az előző blokkot teljesen beolvastuk és még van beolvasatlan blokk a fájlban)

- Ugyanazon kimenő paraméterek, amik a HTP_OPEN_FILE -nél találhatóak a megnyitás során, ha egy új blokk olvasása kezdődik (a http file-ban található blokkok számán kívül minden változik - valószínűleg)

(Ha a file végére értünk, azaz nincs több blokk):

- \$DF00 - \$DF05: \$00

\$06, CLOSE FILE

Lezárja a megnyitott fájlt, HTP file-ok esetben felszabadítja a munka-memóriaterület.

\$07, GET CURRENT PATH

Visszaadja az aktuális path -t (working directory).

Visszatérési érték: \$DF00 - az aktuális munkakönyvtár

\$08, SET CURRENT PATH

Beállítja az aktuális path -t, de csak ha az abszolút path adható meg.

Ha a kiválasztott új path nem abszolút (nem '/' jellel kezdődik), akkor a fgv hívás hibát ad vissza (0x81).

\$09, SET AND START TIMER

Ezzel a hívással egy timert - periodikus megszakítás kérést - lehet bekonfigurálni és elindítani. A megszakítás hatására a Z80 CPU egy IRQ handlerre ugrik. A HomeLab3 operációs rendszerben a \$0038 -as című IRQ handler hívódik meg, ami leteszi stack -be a HL 16 bites regisztert és a \$403F-\$4040 címpáros által címzett területre ugrik.

Az IRQ handlernek a felelőssége, hogy az a kilépés (RET) előtt:

- jóváhagyja (ACK) az IRQ kérést (lásd \$0B)
- visszaállítsa a HL regiszter állapotát a stack-ből
- és újra engedélyezze a megszakítást az EI assembly utasítással

A fgv bemeneti paraméter egy 16bit előjel nélküli szám, ami megadja, hogy mennyiszor 100us periódusidővel generálódjon IRQ kérés (azaz 100us felbontással állítható a periódusidő).

Az IRQ időzítése a cart belső órája szerint működik, az pedig nincs szinkronban a HL3 -as órajelével, technikai okokból kifolyólag.

Bemenő paraméter:

\$DF00-\$DF01: 16 bites szám, ami meghatározza a periódusidőt.

1-65535 közötti számot lehet megadni (azaz 100us és 6.5536s közötti periódusidőt lehet beállítani)

A fgv \$81-es hibával tér vissza, amennyiben a paraméter 0 us lenne.

\$0A, STOP TIMER

Ezzel a fgv hívással egy futó timert lehet leállítani. Több megszakítás nem fog érkezni a carttól.

\$0B, ACK TIMER

Ezzel a fgv hívással lehet jóváhagyni - acknowledge-olni - egy megszakítási kérelmet. Ennek hatására az /IRQ vonal ismét magasra állítódik. Az interrupt handler rutinban a IRQ flag engedélyezésekor az IRQ hardver vonalnak magasban kell lennie, különben az IRQ handler azonnal meghívódik újra.

Mindenképp célszerű az EI hívás előtt meghívni.

Mivel egy cart parancs kiadása és feldolgozása sokáig tart - az IRQ handlereknek pedig határozottan gyorsnak kell lenniük - készült egy alternatív ACK TIMER módszer is. Ez pedig nem más, mint egy tetszőleges byte írása a \$D000-\$DEFF cart ROM területre. A memória szempontjából hatása nincs (a ROM nem írható!), viszont egy pending IRQ kérést jóvá hagy. A két módszer közül ez utóbbi a javasolt ACK TIMER módszer..

\$0C, OPEN BINARY FILE FOR READ

Bináris (tetszőleges) file megnyitása olvasásra. Amennyiben a file nem nyitható meg (nem file, nem létezik, stb...) a megfelelő FatFS (+\$80) hibakódot kapjuk vissza.

Ha a file megnyitható, akkor a parancs visszatérési értéke \$00 lesz és a kimenő paraméter is beállítódik.

Kimenő paraméter

\$DF00-\$DF03 : a file hossza

\$0D, READ BINARY BLOCK

Egy már megnyitott bináris file tartalmából olvas be legfeljebb 255 byte -ot és helyezi azt el a cart-RAM területre.

Kimenő paraméter:

- \$DF00: beolvasott byte -ok száma (0-255)

- \$DF01-\$DFFF: a beolvasott byte -ok, de legfeljebb annyi, ami a \$DF00 címen szerepel

\$0E, SEEK BINARY FILE

Már megnyitott bináris file-ban lehet az aktuális olvasási (és gondolom írási) pozíciót állítani.

Bemenő paraméter:

\$DF00-\$DF03: fájl pozíció, 32bites előjel nélküli egész szám.

\$0F, OPEN BINARY FILE FOR WRITE

Ez a parancs egy bináris file-t nyit meg, írásra. Ha a fájl már létezik, akkor a parancs hibát ad vissza.

Bemenő paraméter:

\$DF00-tól: a file neve. Amennyiben a file már létezik, a parancs hibakóddal tér vissza. Ha a file neve '+' (plusz) jellel kezdődik, akkor a felülírás engedélyezett, létező file esetén a file tartalma törlődik és újra adatot lehet bele írni.

Írás után (\$10) a file-t mindenképpen le kell zárni (\$06) !

\$10, WRITE BLOCK TO BINARY FILE

Ezzel a paranccsal egy írásra megnyitott file-ba lehet adatot írni.

Bemenő paraméter:

\$DF00: a kiírandó byte -ok száma (1-255)

\$DF01-\$DFFF: a kiírandó byte -ok

\$11, DELETE FILE OR DIRECTORY

Egy fájlt vagy egy üres könyvtárat lehet letörölni az SD kártyáról.

Bemenő paraméter:

\$DF00-tól: a törlendő fájl vagy könyvtár neve.

\$12, RENAME FILE OR DIRECTORY

Egy fájl vagy könyvtár átnevezése.

Bemenő paraméterek

\$DF00-tól: a régi (forrás) fájl/könyvtár neve után az új (cél) fájl- vagy könyvtárnevet kell a cart-RAM -ba másolni (mindkét név null-terminated C string).

\$13, <API fgv belső használatra>

\$14, CHECK MOUNTED

A cart számon tartja, h az utolsó mount hívás sikeres volt-e vagy sem. Reset után 0 értéket (sikertelen) ad vissza. Ezzel az API fgv -el ellenőrizhetjük, h már volt-e inicializálva a kártya. Mivel a cart azt nem veszi észre, ha a uSD kártyát eltávolítják, ez a megoldás csak félmegoldás...

Visszatérési értékek

- 0: A cart RESET óta még nem volt inicializálva, vagy az utolsó MOUNT hívás sikertelen volt
- 1: Sikeres MOUNT hívás történt, a cart *valószínűleg* ki tudja szolgálni az SD kártyára vonatkozó parancsokat

\$15, GET API VERSION

Ez a fgv minden esetben \$00 -val tér vissza, jelenleg. Ha a kártyának lesz egy következő API verziója, akkor a visszatérési érték majd változni fog. Ebből lehet majd tudni, h használhatóak-e az esetlegesen újonnan bevezetett függvények.

Joystick kezelés

A carton található 2 joy állapotát a \$50 és \$51-es port (decimálisan 80 és 81) olvasásával lehet lekérdezni (hátról nézve, azaz a cart HL3-hoz csatlakoztatott állapotában bal és jobb oldali csatlakozó sorrendben).

Ez a két port csak olvasható, írásra nem történik semmi. A visszaolvasott adat alsó 5 bitje a lényeges. Ezek magas-aktívak, azaz 0, ha a megfelelő irány nem aktív és 1 amennyiben igen. Egy nyugalomban hagyott (vagy nem csatlakoztatott) joy port visszaolvasott értékének alsó 5 bite mindig 0.

A bitek jelentése:

bit 0 (1): FEL

bit 1 (2): LE

bit 2 (4): BAL

bit 3 (8): JOBB

bit 4 (16): TŰZ GOMB

AY-3-8910 kezelés

Az AY-3-8910 -es részletes programozására nem térek ki, arról az interneten lehet találni elegendő dokumentumot.

A PSG chip programozása regiszter írásokkal történik. Egy OUT utasítással ki kell jelölni a kiválasztott 16 regiszter egyikét (0-15) majd egy újabb OUT utasítással az adatot kell kiírni az előbb kiválasztott regiszterbe.

A cart-on az AY-3-8910 IC cím portja a \$47 (decimálisan 71), az adatportja pedig a \$46 (decimálisan 70) címeken található.

Az AY-3-8910 IC a HL3től független 1.5MHz -es órajelen üzemel. A hangok frekvenciáit így kell kiszámolni.

Érdekességként említendő, hogy az AY chip specifikációban szereplő regiszterek címezése nem folytonos. R0-R7 a 0-7 című regiszterek. R10 a 8. regiszter, R11 a 9. regiszter, stb...

Ami számomra kellemetlen meglepetés volt, hogy a hang-boríték egyszerre a 3 hangra vonatkozik, azaz itt nem lehet csatornánként ADSR paramétereket állítani, csak egyszerre, a már összekevert 3 hangra. Elkényeztetett a SID...

Függelék

megjegyzés: az itt ismertetett programokat nem próbáltam ki,

Példa: egy rövid bináris file beolvasása, assembly-ben

```
CTRL_PORT equ $40
CMD_MOUNT equ $00
CMD_OPEN_BIN equ $00
CMD_READ_BIN equ $00
CMD_CLOSE equ $00
ADDR_CART_RAM equ $DF00
STR_FILE_NAME:
    db "filename.bin", $00
START:
    ld a, CMD_MOUNT ; SD card must be mounted first
    out (CTRL_PORT), a
    call WAIT_FOR_RESPONSE
    ret nz

    ld hl, STR_FILE_NAME
    ld de, ADDR_CART_RAM
COPY_FILE_NAME:
    ld a, (hl)
    ldi
    or a
    jr nz, COPY_FILE_NAME
OPEN_FILE
    ld a, CMD_OPEN_BIN
    out (CTRL_PORT), a
    call WAIT_FOR_RESPONSE
```

```

ret nz
; let's drop the file's metadata returned on $DF00
ld a, CMD_READ_BIN
out (CTRL_PORT), a
call WAIT_FOR_RESPONSE
ret nz
; the first 255 bytes of the file is on $DF01 - $DFFF
; $DF00 contains the num of valid bytes in buffer
ld a, ($DF00)
ld c, a
ld b, 0
ld hl, $DF01
ld de, $6000
ldir
ld a, CMD_CLOSE
out (CTRL_PORT), a
call WAIT_FOR_RESPONSE
ret ; I would either way return

; The wait routine, returns Z flag set if OK, cleared otherwise.
; Register A is overwritten.
WAIT_FOR_RESPONSE:
    push bc
    ld b, $10
WAIT_FOR_FIRST_ACK:
    in a, (CTRL_PORT)
    cp $7F
    jr z, COMMAND_ACKD
    djnz WAIT_FOR_FIRST_ACK
    inc b ; clear Z-flag for error (b->1)
    jr LEAVE_WITH_CLEAR_STACK
COMMAND_ACKD:
    ld b, $ff
CART_BUSY_LOOP:
    in a, (CTRL_PORT)
    cp $FF
    jr nz, RETVAL_ARRIVED
    nop ; kis extra várakozás
    djnz CART_BUSY_LOOP
    ; nothing to be done here. Register A contains $FF, Z flag will be cleared (nz)
RETVAl_ARRIVED:
    and $80 ; Z flag is cleared on error values, set on success
LEAVE_WITH_CLEAR_STACK:
    pop bc
    ret

```

Példa: egy rövid BASIC program SD kártya listázására

```

10 ca=64
20 poke ca, 0 : rem MOUNT
30 gosub 1000 : rem wait for ack
40 if rv<>0 then print "nem sikerult az SD-t felcsatolni!!": end
50 print "mount ok."
60 poke $DF00,0 : rem gyokerkonyvtar
70 poke ca,1 : rem open DIR
80 call 1000
90 if rv<>0 then print "nem sikerult megnyitni a konyvtarat.":end

```

```

100 poke ca, 2
110 call 1000
120 if rv<>0 then print "bejegyzes nem olvashato" : end
130 if peek($DF05) = 0 then goto 200 : rem vege a listanak
140 i=0
150 c=peek($DF05 + i)
160 if c<>0 then print chr$(c); : i=i+1 : goto 150
170 print : goto 100
200 poke ca, 3 : rem close DIR
210 gosub 1000
220 end
1000 C=20
1010 C=C-1: If C=0 then Print "Nincs cart...": End
1020 RV=peek(CA): If RV<>127 then Goto 1010
1030 C=20
1040 C=C-1: If C=0 then Print "Hiba olvasaskor": End
1050 RV=peek(CA): If RV=255 then Goto 1040
1060 Return

```

Példa: SAVE_HTP_FILE hívása assembly-ben

megjegyzések:

- 1) a cart már inicializálva van.
- 2) ez a program csak RAM-ból tud futni az önmódosító kód miatt

```

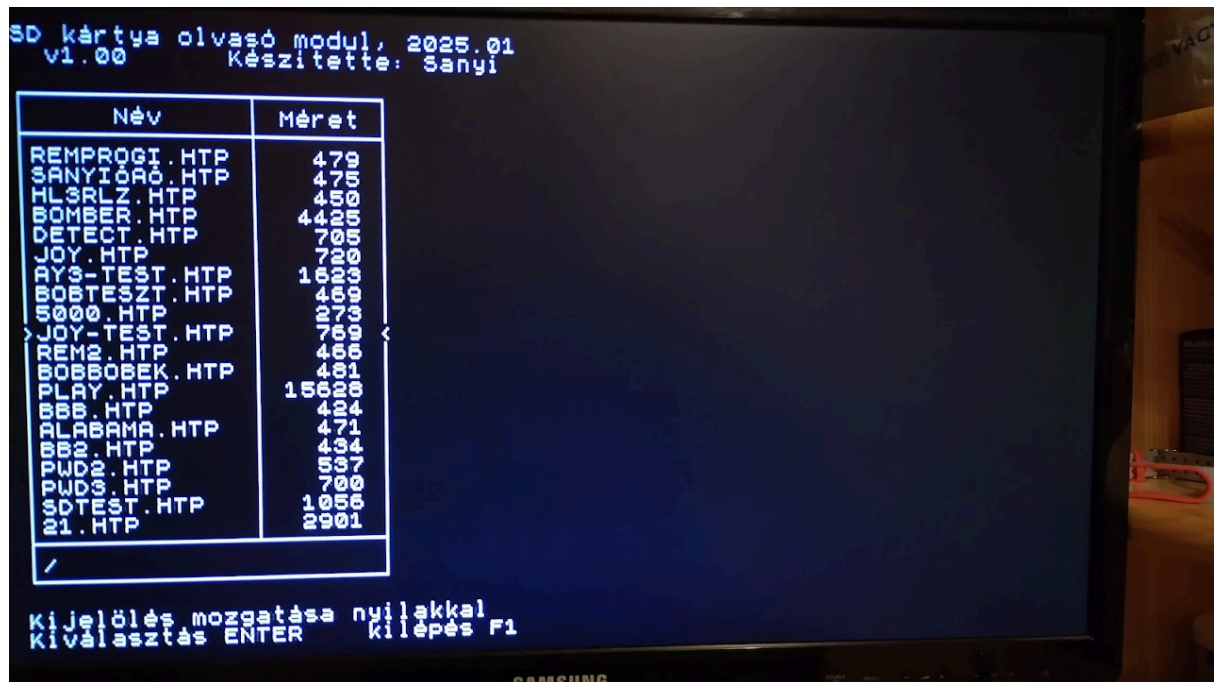
ENTRY
    ld hl, FILENAME
    ld de, $5000
    ld bc, $6000
    call CALL_API
    ret

CALL_API
    push hl
    ld hl, ($D00E)
    ld (JUMP_TO_SAVE_API + 1), hl
    pop hl
JUMP_TO_SAVE_API
    jp $0000                ; a címet beállítja a CALL_SAVE rutin

FILENAME
    db "myname", $00

```

Galéria



A Menü, file listával. Eredetileg a 32 és 64 oszlopos mód támogatását is terveztem...



Néhány összerakott példány, tesztelés után



Tesztelés, az SD kártyáról betöltött Bombázó c. játékprogrammal (Nagy Attila műve)



HL5 -höz adapter

Köszí!

Köszönettel tartozom családomnak, mert elnézték az el-el tünevezéseimet az otthoni közös életből.

Köszönet illeti Hajdú Zoltánt, aki egy repro HomeLab3 -as számítógépet is a rendelkezésemre bocsátott a cart fejlesztéséhez (ez a gép látszódik a fényképeken is), az eredeti ötlet (kérés) is az ő fejéből pattant ki.

Köszönet illeti Nagy Attilát is, aki az RM32 fejlesztése során szerzett tapasztalataival segített, joy és hang képességek körül adott sok-sok hasznos tanácsot, visszajelzései alapján lett a cart egyre jobb.

Köszönet illeti Rákos-Zichy Pétert, aki rendelkezésemre bocsátotta a HL5 gépét, így alkalmam nyílt leellenőrizni a HL5 cart adaptert.

Édesanyám emlékére

Vass Sándor
2025.február 8.