

Emu41 documentation

1. Presentation:

Emu41 is a software emulator of the HP-41C machine and HP-IL system.

Main features:

- run under DOS, DOS box in Windows 9x, command box in Windows 2000, NT, etc,
- emulate ROM and XMEM modules,
- support for advanced modules: Hepax, MLDL, RamBox,
- emulation at machine code level: can execute synthetic programs and micro-code,
- run on all 80x86 MS-DOS systems, even with only 512Kb RAM.
- FREEWARE: free for non commercial usage,
(a short mail with your opinion to the author would be appreciated),
- emulation of the HP-IL loop and 8 HP-IL devices (one video display, three mass storage units, one printer port, one DOS interface, two serial COM),
- support of 43 and 50 lines display modes,
- improved display and keyboard management,
- automatic shift key management in ALPHA mode for easier alpha character entry, automatic BST key management without shift,
- manual speed switching with TAB key (between auto-speed and fast)
- compatible with standard ROM image format (.rom, .41r),
- emulation of the W&W 64K RAMBOX II (or HP-41CY),
- emulation of the Eramco RAM Storage Unit (RSU) up to 128kb,
- support of the PC serial ports for easy program download to the HP-41,
- NEW: the free version now includes the control from Emu41 of an external HP-IL loop, and/or control of the internal devices by an external HP-41C, using either the classic HP-IL ISA board or a HP-IL/serial converter (tba).

Note: in all this document, HP41C or 41C means any of the 41 serie: HP-41C, HP-41CV or HP-41CX.

2. Preparation for use:

Emu41 uses ROM images of HP41C ROM and modules (X-F, Time, ...).

You can find ROM images on other HP-41 sites like www.hp41.org

To help you, you will find on my site the ROM images of the HP-41CV machine, as well as the X-functions, Time and HP-IL modules. Although I don't have a formal authorization from the Hewlett-Packard company, I considered that this is consistent with the publication by HP of the source listings of these ROM. If it happens that the Hewlett-Packard company should be opposed to this, I would immediately remove these files from my site. Don't ask me to provide other HP or third party ROM images.

THESES FILES ARE NOT PROVIDED WITH EMU41
FOR OBVIOUS COPYRIGHT REASONS

You also can transfer these ROM on PC from your calculator.

FOLLOWING INSTRUCTIONS ASSUME THAT YOU HAVE
ADVANCED KNOWLEDGE ABOUT THE HP41C.

Instructions:

- make a link between the HP41C and the PC with the HP82164A HPIL/RS232 converter or with the HP82973A HP-IL interface card for PC (you have to know how to make these interfaces work),
- use a module with a ROM disassembler function (Zenrom, MLDL, ...) to transfer the ROM to PC as ASCII files with one word by line like this:

address word instruction (instruction is not used)

ex:

```
0000 210 *
0001 006 NCGO 0180
...
```

save the data in *.lst files, e.g. ROM41CV.LST

- after the transfer, use the CLBIN41 utility to build the binary files *.bin that can be used by Emu41.

e.g. CLBIN41 ROM41CV.LST ROM41CV.BIN

ROM41CV.BIN must be 15360 bytes long.

(Note: 15360 bytes correspond to 12288 words of 10 bits)

- Note to HP-41CX users: the HP-41CX has 24kwords in it: 12kwords as the HP-41CV, plus Xfunctions, plus TIME, plus CX functions. See the example of emu41.ini file below.

Note: Emu41 also uses pack format (.bin files) that can save space on small systems.

o recognize the packed or unpacked format, just look at the file size: a 4k ROM uses 8kbytes in unpacked format (.rom), but only 5kbytes in packed format.

To convert .rom format to .bin format, use the CLBIN41 utility with the 'r' option:

xample for a xio.rom file:

```
CLBIN41 xio.rom xio.bin /r
```

3. Emu41 basic functions

3.1 Running the emulator:

```
emu41 [option]
```

options:

/d : call debugger at startup
/m : slow host machine
/sx: emulation speed

/m option is used to adjust the emulation speed to the host processor.
It has to be used only with processor 8088 to 286.
With 386 processors and higher, it is not useful.

/s option is used to set the emulation speed:
/s0 : automatic (default value), emulation is done at the same speed than the HP41C while messages are displayed, then at maximum speed when the display does not change any more. This mode, active by default, let the CAT and the messages be displayed at normal speed, and let the other calculation or processing parts be executed at maximum speed.
/s1 : force the standard speed, same than the HP41C. It is useful in some cases.

/d option call the debugger at startup (see below).

The /d option passes control to the built-in debugger (see below) at startup.
If the MFENTRY.ADD file is present, entry points addresses are displayed with symbolic mnemonics.

3.2 Exiting the emulator:

Normal exit with the F1 key (ON/OFF) or with the OFF function, with automatic saving of main and extended memories.

Emergency exit (emulated CPU stuck, or user choice not to save RAMs) by ctrl-Z, then 'q' (see debugger description below).

In case of difficulty, you can make a master reset (memory lost) by deleting the mem41.dat file.

3.3 Files used by Emu41 (minimum configuration):

```
emu41.exe      : emulator  
clbin41.exe   : utility to build *.bin ROM files  
emu41.ini     : emu41 initialization file, see description below  
rom41.bin     : image of HP41C 12Kwords ROM  
mem41.dat     : main memory (320 registers) and status registers  
xmem41.dat    : extended memory (600 registers) and X-M status  
registers
```

The mem41.dat and xmem41.dat files are automatically created by Emu41.

3.4 Debugger:

A debugger is available in Emu41.

You call it with ctrl-Z, or with the /d option at startup.

To return to emu41, use the 'r' (run) command.

The 'q' command is used to exit from emu41. In this case, RAM are NOT saved.

'?' can be used to get the list of all available commands.

This debugger has been written to study the micro-code, it lacks a lot of commands, you can't change the values of registers or memory locations (because I didn't need it).

3.5 emu41.ini file

This file describes the ROM modules plugged into the emulated machine, as well as the virtual HP-IL devices.

This file is made of two sections, each declared by a header:

[modules]: start of the ROM modules declaration section,

[devices]: start of the HP-IL devices declaration section.

- ROM modules

ROM modules are only limited by the addressing space of the Nut processor (64 Kwords total, i.e. 16 pages of 4Kwords).

The internal ROM use the 3 first pages 0 to 2 (12 Kwords).

Each module use 4, 8 or 12 Kwords, in 1 or 2 pages.

Format of the lines in emu41.ini:

page(0-15) size(Kwords) file

Size is 4Kwords for one page module, 8 Kwords for 2 pages module, and 12 Kwords for modules with bank switching such as the Advantage module.

Example of [modules] section for an HP-41CV with X-Function (port 1, page 8), Time (any port, page 5) and HP-IL (any port, pages 6 and 7)

```
0 12 rom41cv.bin
5  4 time1c.bin
6  4 prter2e.bin
7  4 hpil1h.bin
8  4 xfcn1c.bin
```

Example of [modules] section for an HP-41CX with Advantage module (port 1, pages 8 and 9). X-F module is in page 3, and there are in page 5 two ROM of 4 Kwords with bank switching (time and cx)

```
0 12 rom41cx.rom
3  4 cxfcn2d.rom
5  8 cxtime2c.rom
8 12 adv41b.rom
```

Note that the HP41CX use bank switching for the page 5.

- DEVICES HP-IL

The [devices] section is used to declare what are the active internal HP-IL devices, and what is the device order, as well as the external HP-IL interface (XIL).

The internal device order is chosen by the user, but it is not allowed to declare twice the same device. It is recommended to put the XIL declaration after all the others, so the external devices are after the internal devices.

See more details on HP-IL at chapters 4.3 and 4.4.

Example:

```
[devices]
DISPLAY
HDRIVE1
LPRTER1
DOSLINK
XIL
```

3.6 HP41C keyboard emulation:

The special keyboard of the HP41C is difficult to emulate with a PC keyboard.

Here is the chosen mapping with minimum inconvenience:

```
PC    HP41C
F1   : ON
F2   : USER
F3   : PRGM
F4   : ALPHA
F5   : SHIFT
F6   : SST      ( BST  after F5:SHIFT)
F7   : R/S      ( VIEW after F5:SHIFT)
F8   : XEQ      ( ASN  after F5:SHIFT)
F9   : MODE     (if the HP82143 printer is emulated)
F10  : PRX      (if the HP82143 printer is emulated)
F11  : ADV      (if the HP82143 printer is emulated)
F12  : reset    (as the HP41CX ON/<- sequence)
Enter: ENTER    (!! ) (CAT after F5:SHIFT)
<--: <-        (backspace = delete or CLX)
<-  : <-        (cursor left= delete or CLX)
^   : SHIFT     (cursor up = SHIFT, same as F5)
v   : RDN       (cursor down = RDN)
->  : X<>Y      (cursor right = X<>Y)
PgDn: SST       (Page Down = SST)
PgUp: BST       (Page Up = BST)
ctrl-A: ALPHA   (same as F4)
ctrl-R: R/S     (same as F7)
ctrl-X: XEQ     (same as F8)
```

There is also the following mapping (as the HP41C keyboard):

A: s+ / s-	B: 1 / X / Y ^ X	C: SQRT / X ^ 2	D: LOG / 10 ^ X	E: LN / E ^ X
F: X <> Y / CLs	G: RDN / %	H: SIN / ASIN	I: COS / ACOS	J: TAN / ATAN
	K: XEQ / ASN	L: STO / LBL	M: RCL / GTO	
N: ENTER / CATALOG	O: CHS / ISG	P: EEX / RTN		
- : X = Y ?	7 : SF	8 : CF	9 : FS ?	
+ : X <= Y ?	4 : BEEP	5 : P - R	6 : R - P	with
* : X > Y ?	1 : FIX	2 : SCI	3 : ENG	SHIFT
/ : X = 0 ?	0 : PI	. : LASTX		

The BST key is simulated with the page up PC key without shift.

In ALPHA ON mode, Emu41 handles automatically the shift key, for instance numeric keys don't need to be shifted, as it should with the HP41C.

Alpha characters like <, >, +, -, /, * can be keyed directly from the PC keyboard.

This is also valid in alpha function argument like XEQ.

Example: to key in XEQ "PROG1", type:

"K" (or F8), F4 (for alpha), "P", "R", "O", "G", "1" and F4. No need to shift "1" !

The following special characters can be keyed in with:

```
sigma      : SHIFT F
different: #
angle      : SHIFT O
```

3.7 Display emulation:

Emu41 now supports 43 and 50 lines display mode. To set this mode, use the MODE CO80,43 or CO80,50 command (DOS, MS-DOS command box in Win95/98), or set the number of lines in the Command Box Properties (WinNT, Win2000, etc, ...) before launching Emu41.

The display is emulated in text mode. It is made of:

- a display field of 12 characters and 12 separators
(' ' , ' . ' , ' : ' or ' , ') ,
- a field with 12 flags (BAT USER G RAD SHIFT 0 1 2 3 4 PRGM ALPHA),
- a flag showing the emulator state:
' ' : sleeping processor,
' o ' : normal speed processor,
' * ' : fast speed processor.

The HP41C special characters are replaced by ASCII equivalents:

Code	character	display
-	flying goose	>
-	starburst	*
0	overbar	~
1-6	little hangman	x
12	micro	u
13	angle	a
29	different from	#
96	alpha	'
126	sigma	s
127	append	`

4. Advanced functions

4.1 Emulation speed

Emu41 automatically manages the emulation speed. Emu41 is designed to provide a average speed slightly greater than HP41C (independently of the PC speed) in most cases.

The switching to the maximum emulation speed (which depends on the actual PC) is done when there is no more key stroke, nor any change on the display for about 1s. This allows to provide a speed similar to the HP-41C during interactive operations (ALPHA, CAT, DIR, ...), and also a maximum speed during long processing. Especially in case of a large number of ROM modules, executing a function with XEQ alpha is slow on the HP41C, because it has to poll all the modules. Emu41 switches to fast speed after about 1s and limits the search time.

The additional annunciator on the display (after the regular HP41C annunciators) lets the user knows the emulation state (see chapter 3.2).

To automatically speed up the execution of a program, just add a VIEW or AVIEW at the beginning of the program to freeze the display, this avoids the 'flying goose' and let Emu41 switch to maximum speed.

You can switch manually between auto-speed and fast speed with the TAB key.

You can also force normal speed by using shift-TAB. This can be useful in some particular situations where fast speed has to be disabled.

4.2 Support for various modules

The various ROM modules must be declared in the [modules] section of the initialization file (see chapter 3.5).

- Standard ROM modules

Emu41 supports standard 4k, 8k and 12k ROM modules.

A 4k ROM uses one page. A 8k ROM uses two consecutive pages starting from an even page.

A 12k ROM uses two pages like a 8k ROM (the second and third 4k blocks share the same odd page with bank switching).

Standard ROM modules must be declared in page 8 to 15.

Examples:

```
8 4 math1a.rom ; standard 4k ROM
10 8 navig.rom ; standard 8k ROM, uses pages 10 and 11
12 12 adv41b.rom ; standard 12k ROM, uses pages 12 and 13
```

(bank switching)

Note: you may find on the Web 8k and 12k ROM images as separated 4k blocs.

To use them with Emu41, just merge the blocs into an unique file:

example for Advantage ROM:

```
copy AdvL1-1B.rom /b + AdvU1-1B.rom /b + AdvU2-1B.rom adv1b.rom
```

- X-functions (built-in in the HP-41CX) and X-memory modules

These modules are fully supported.

On the HP-41CV, X-function ROM module can be declared in any page from 8 to 15.

On the HP-41CX, X-function ROM module must be declared in page 3.

- TIME module (built-in in the HP-41CX):

The clock and calendar functions work, except:
CLOCK function, as well as the shift-ON sequence,
alarms are not automatically launched.

The Time module clock is synchronized with the PC clock at Emu41 startup.

The Time module must be declared in page 5. The HP-41CX internal time module uses 2 banks in page 5.

- Card reader:

Although Emu41 doesn't support the card reader, it is possible to use the ROM for example to use the HP-67 compatibility functions.

It is mandatory to declare the card reader ROM in page 14:

```
14 4 CrdCdr1E.rom
```

- HP82143 printer:

Emu41 partially emulates the HP82143 printer, i.e. in text mode only.

The graphic mode is not supported (functions such as ACCOL, SKPCOL, ...).

Double width characters are emulated by inserting a space between each character.

The HP82143 printer emulation shares the PC screen with the HP-IL 'DISPLAY' device.

The Printer ROM must be declared in page 6:

```
6 4 Prter1e.rom
```

It is possible to use the HP82143 ROM in page 6, and at the same time the part of the HP-IL ROM for mass storage and HP-IL control in page 7. This corresponds to the configuration of using the HP82143 printer together with the HP-IL module with the switch in 'disable' position.

When using the HP82143 printer ROM, the F9, F10 and F11 function keys are used for:

F9: printer mode (OFF, NORM, MAN, TRACE)

F10: PRX

F11: ADV

- HP-IL module

The HP-IL module is supported. The HP-IL module ROM is 8k long, and is made of 2 parts:

Printer functions (page 6),

Mass storage and control function (page 7).

The Printer part is not mandatory, and can be omitted or replaced by the HP82143 ROM.

The other HP-IL related modules (IL development, X-IO, Ext-ROM) are supported as well.

Some very advanced or very low level features of the IL development (like launching a program in case of HP-IL interrupt) may not work correctly.

The emulated HP-IL devices are described in the next section.

- IR printer module

This module is not supported due to the lack of information on IR module hardware.

Using this ROM will likely corrupt Emu41 operations.

Any information on the internals of this module is welcome.

- HEPAX module:

Emu41 emulates a 8k RAM Hepax module (Hepax Standard Module).

The Hepax ROM is 16k long and must be declared as:

```
'page' 16 hepax1c.rom
```

page must be an even page. After Emu41 startup, the Hepax ROM is moved to an other free 4k page, and the 8k at page and page+1 are used for the Hepax RAM.

The Hepax RAM is saved in the Hepaxram.dat file. This file doesn't have to be declared in the initialization file, it is used automatically when the Hepax ROM is declared, and is automatically created at the first Hepax module use.

It is possible to build a ROM module from the Hepax RAM by copying the hepaxram.dat file to an other name with the .bin extension.

Note: if the Hepax module ROM is declared in an odd page, the ROM is not moved to an other page, and the Hepax RAM can not be used.

Note: you can extend the Hepax RAM to 16k by using the MLDLRAM (see below) just after the Hepax RAM pages:

```
8 16 hepax1c.rom ; pages 8 & 9 hold the Hepax RAM (hepaxram.dat file)
10 8 mldlram.dat ; 2 more pages for Hepax
```

The only difference is that it is not possible to write protect the MLDL RAM with the RAMTOG function.

- MLDL:

Emu41 supports MLDL (Machine Language Development Lab) in the sense that Emu41 supports a 8k RAM block that must be declared as:

```
'page' 8 mldlram.dat
```

'page' is an even page number. MLDL RAM uses page and page+1.

The reserved name mldlram.dat is mandatory.

This MLDL RAM can be used with Zenrom, David Ass, or any other MLDL OS.

A empty mldlram.dat file is available in the Emu41 package.

- W&W 64K RAMBOX II

Emu41 fully supports the W&W 64K RAMBOXII.

A blank 64k Ram Box file is provided in Emu41 package.

It must be declared as:

```
8 64 ramboxii.dat
```

The name "ramboxii.dat" is reserved for RAMBOXII emulation.

If you rename it to an other name (e.g. myrambox.dat), then it will act like a "Rom Box", it means that it will no more be possible to write into the pages. It is a way to write protect your Ram Box.

Like the real RAMBOXII, you can override some pages by "plugging" modules. Example:

```
8 64 ramboxii.dat
14 8 navig.rom ; override pages 14 & 15
```

Page 8 must not be overridden, otherwise the RAMBOXII will no more be recognized by Emu41 and Ram Box content will not be saved at Emu41 exit.

- Note: Using the RAMBOXII with other OS

The RAMBOXII can be used with other OS like ESRSU or Hepax:

You will have to clear page 8 to get rid of RAMBOX OS.

With ESRSU

```
8 64 ramboxii.dat ; pages 8 to 13 available for ESRSU
14 8 es41.rom ; override pages 14 & 15
```

Initialize the pages either with INIDATA or INIPRGM.

With Hepax:

```
8 64 ramboxii.dat ; pages 8 to 13 available for Hepax
14 16 hepax1c.rom ; Built-in Hepax RAM in pages 14 & 15.
```

With CX ROM, leave page 7 free (i.e. no HP-IL) to let Hepax install itself

Clear all the pages with CLRAM, then restart Emu41 to initialize the Hepax file system.

If you want to use Hepax, RamBox and HP-IL at the same time, you may declare the Hepax ROM in page 15 after proper initialization, but Hepax file system will be reduced to 24k, as built-in Hepax Ram (hepaxram.dat file) will not be used (see Hepax description above). Only the RAMBOX block A can be used (i.e. 24k effective with ESRSU, 24k to 32k with Hepax).

- Eramco "128k" RSU

Emu41 supports the Eramco "128K" Ram Storage Unit (RSU).

First, note that the last two pages (14 and 15) are used for the RSU OS, and are not available for data or program storage. It means that the actual RSU capability is 96k.

The RSU system must be declared as:

```
8 96 es_rsu.dat ; 96k RAM = 4 banks of 24k in pages 8 to 13
14 8 rsu_os.rom ; the RSU OS (supporting the multiple banks)
```

A blank 96k RSU file is provided in Emu41 package.

The name "es_rsu.dat" is reserved for RSU emulation.

You can override some pages by "plugging" modules over it. Example:

```
8 96 es_rsu.dat
14 8 rsu_os.rom
10 8 navig.rom ; override pages 10 & 11
```

Page 8 must not be overridden, otherwise the RSU will no more be recognized by Emu41 and RSU content will not be saved at Emu41 exit.

- Notes about the Eramco RSU:

The Eramco RSU uses 4 banks. Use the SETBANK function to switch banks. Note that Eramco bank switching scheme is not standard (doesn't use the ENBANK opcodes).

The RSU OS must be in pages 14 and 15 because it uses absolute address coding. It will not run correctly in other pages.

At Emu41 startup, the bank 1 is active.

The PORTON/PORTOFF functions are not supported.

4.3 HP-IL support - internal virtual devices

Emu41 emulates the HP-IL interface, and provides 6 internal (or virtual) devices.

These devices must be declared in the [devices] section of the Emu41.ini initialization file (see chapter 3.5).

Below are described the internal devices:

- DISPLAY

The DISPLAY device emulates the HP82163 video interface.

```
AID=32
ID="DISPLAY"
```

Declaration in Emu41.ini:

```
DISPLAY
```

Display devices are automatically used by the 41C for print functions. To enable printing on the DISPLAY, just set flags 15 and/or 16.

Note: the escape sequences are partially supported. These sequences are not used by the HP41C (except with the Paname module).

- HDRIVE1

The HDRIVE1 device emulates a mass storage unit compatible with the HP9114 by using a DOS image file.

```
AID=16  
ID="HDRIVE1"
```

Declaration in Emu41.ini:

```
HDRIVE1 {name}
```

'name' is optional, and set the name of the image file. Default is: hdrive1.dat.

DDT and DDL command set compatible with the HP82161 and HP9114 units.

The data are recorded in a image file in the Emu41 working directory. Maximum size of this file is 640k.

This file is created with NEWM. When created, the file is 32k long, its size is then increased when needed, up to 128k with the HP-IL module standard functions, or up to 640k with special instructions or with the Extended IL ROM from SKWID (don't confuse it with the HP X-IO module).

The format of this file is compatible with LIF disk images that can be found in some Web archives. So it is possible to directly read these files from these archives to Emu41 (or to an external HP41C with the XIL option).

- HDRIVE2

The HDRIVE2 is similar to HDRIVE1.

Main usage of HDRIVE2 is to access LIF image archive, whereas HDRIVE1 is used as working drive.

Typical HDRIVE1/HDRIVE2 usage example:

```
HDRIVE1 mydisc.dat           my personal disc  
HDRIVE2 swap\lexfl1.lif     path to a LIF archive image
```

Then it is possible to read files from HDRIVE2, and save them on HDRIVE1 if needed.

Tips: It may be convenient to set the archive as read-only file, so Emu41 will not overwrite any archive files by mistake.

It is easy to get confused with the HP-41 AUTOIO behaviour, see the note below FDRIVE1.

To do so, use the File Properties under Windows, or the ATTRIB utility under DOS.

- FDRIVE1

The FDRIVE1 device emulates a mass memory unit compatible with HP9114 by using the PC 3 1/4 floppy drive. The goal is to allow reading the old LIF discs created with the HP9114 unit.

```
AID=16  
ID="FDRIVE1"
```

Declaration in Emu41.ini:

```
FDRIVE1 {A|B}
```

The optional A (default) or B parameter set the PC drive.

DDT and DDL command set compatible with the HP82161 and HP9114 units.

This device doesn't allow to format discs. Nevertheless, it can read and write on discs previously formatted with the HP9114 unit.

Note: It is recommended to use this device only to read old LIF discs, it is also recommended to write protect the discs.

Note: This emulation doesn't work on some modern PC. Reason is that this emulation relies on low level BIOS calls to manage LIF formatted discs (256 bytes/sector). These calls are historic (there are not used by DOS/Windows) and seem to have been removed (or at least are no more supported) in modern PC BIOS.

- Note about the use of multiple mass storage units:

If more than one HDRIVE1, HDRIVE2 and FDRIVE1 drives are simultaneously declared, the Emu41 (like the HP-41) behaviour is such as this: in AUTOIO mode, Emu41 searches on all drives for the file to read or write.

Espacially:

You can't know on which drive the HP-41 will write in AUTOIO mode!

In case of no disc in the FDRIVE1 floppy drive, an error is returned, even if the operation was targeted to the HDRIVE1 or HDRIVE2 drive!

So it is recommended to use both HDRIVE1 et HDRIVE2 with care, and to declare FDRIVE1 only when it is needed.

Look at the HP82160 manual under "Using Multiple Mass Storage Devices", page 39 in the english edition (feb.81).

This note also applies when using external drives (HP9114, HP82161), or when using an external HP41C (see chapter 4.4).

- DOSLINK

The DOSLINK device provides an interface with the DOS file system.

It is possible to import data from a DOS file, and export data to a DOS file.

```
AID=78
```

```
ID="DOSLINK"
```

Declaration in Emu41.ini:

```
DOSLINK
```

Data sent to DOSLINK are written into the emu41out.dat file.

Data received from DOSLINK are read from the emu41in.dat file.

These files are in the Emu41 working directory.

- LPRTER1

The LPRTER1 device is a pure interface that uses the parallel port.

Although its name is LPRTER1, it is recognized as an interface similar to the HP82165 GPIO interface or the HP82166 converter that are commonly used as parallel printer interface.

```
AID=64
```

```
ID="LPRTER1"
```

Declaration in Emu41.ini:

```
LPRTER1
```

To use it as printer, you must switch to MANIO mode and do a SELECT:

e.g.:

```
MANIO
```

```
"LPRTER1"
```

```
FINDID
```

```
SELECT
```

This interface is especially useful with the Ext-ROM and SKWINBC-ROM from SKWID.

These modules allow you to print a program in multicolumn format, and also print bar codes on standard ink jet or laser printers.

- SERIAL1

The SERIAL1 device is a pure interface that uses a serial port.

```
AID=66
```

```
ID="SERIAL1"
```

Declaration in Emu41.ini:

```
SERIAL1 [ComPort[, InitByte]]
```

ComPort and InitByte are two optional parameters. Initbyte is a hex number.

ComPort specifies the COM port. Default value is 1, for COM1.

InitByte specified the communication parameters. It is a hex value with the meanings:

bit	7	6	5	4	3	2	1	0				
speed				parity		stop		length				
	010	=	300	x0	=	none	0	=	1bit	10	=	7bits
	011	=	600	01	=	odd	1	=	2bits	11	=	8bits
	100	=	1200	11	=	even						
	101	=	2400									
	110	=	4800									
	111	=	9600									

Default value is E3(hex) = 9600 bauds, no parity, 1 stop, 8 bits.

Actually, this is the value send to the INT14, function 0

- SERIAL2

Similar to SERIAL2, but ID="SERIAL2" and defaults to COM2.

- Important notes about SERIALx:

Emu41 uses the BIOS functions (INT 14) to communicate with the COM port.

The BIOS functions are very limited and some points must be understood:

- Usually, PC BIOS expects the CTS and DSR lines to be active to be able to send a character. If these lines are inactive, no character will be transmitted.

See the correct cable wiring in an associate document on my site.

- In Windows environment, Emu41 must be active i.e. not in idle state, but in a running program or running a I/O function (INAN, INP for instance) for reliable reception of characters from the Windows OS.

- INA function from HPIL module terminates when there is no more bytes are available.

Functions from the X-I/O module are much more powerfull and wait for the correct end condition (number of bytes or terminator). X-I/O module is highly recommended for use with SERIALx.

- Virtual Communication Ports (VCP) provided by USB/serial bridges seem to work. If a USB/RS232 interface installs itself as COM4 for instance, declare it as:

```
SERIAL1 4 (or SERIAL2 4)
```

- The serial support is provided for experimental or hobby purpose only.

It is believed to be reasonably reliable for downloads from PC to HP41 through the HP82164A.

Upload from the HP41 to the PC is not guaranteed.

I can't provide any support or help for problem solving.

4.4 Support of the external HP-IL interface

The free Emu41 version now allows also to manage real external devices through either the HP82973A HP-IL board for PC (or the equivalent board rebuilt by Christoph Klug), or a HP-IL/serial converter. It also allows an external HP41C to control and use the internal devices.

The declaration of the external loop is made with the XIL declaration in the [devices] section of the emu41.ini file:

```
XIL {addr}
```

or

```
XIL COMx
```

If the parameter is a hexadecimal number, then it specifies the address of a HP-IL/PC ISA board, default value is 1700(hex).

If the parameter is the string "COMx" and 'x' is a number from 1 to 9, then it specified a HP-IL/serial converter connected to a COM port, either a native COM1 or COM2, or a virtual communication port (VCP) provided for instance by a USB/RS232 converter.

Note: a HP-IL/serial converter is NOT a HP82164 HP-IL/RS232 interface. At the time being, the only existing HP-IL/serial converter is a custom hardware made by the Author.

Although not mandatory, it is recommended to put the XIL declaration after all the internal device declarations.

There are two usages of the external HP-IL interface:

- control of external devices from Emu41

Emu41 manages the devices connected to the PC. The external HP-IL loop should be closed, and all devices should be turned on.

Emu41 includes a very nice feature: the external loop integrity is checked each time the 41C send a identify (IDY) message, i.e. very often. If the IDY message doesn't return in a short time, then the external loop is assumed to be broken, and is bypassed until a next IDY message shows to Emu41 that the external loop is valid.

In this way, we can use Emu41 with an open external loop, or with shutdown devices, and still have access to the internal devices. To have access to the external devices, just close the loop and turn on all devices.

- control of the internal devices from an external controller

An external HP-IL controller (41C, 71B, 75C/D, ...) can access the Emu41 internal devices.

To do so, Emu41 must be inactive (waiting for key stroke, no running program or function).

The 41C, as well as Emu41, assign HP-IL addresses at each loop access, so no special operation is needed to pass control from/to Emu41 or 41C. For other controllers, a loop initialization is needed (e.g. RESTORE IO).

A possible usage is data exchange between Emu41 and HP41C through a disc drive, e.g. HDRIVE1.

Here is a possible way:

- start Emu41,
- connect and turn on the external HP41C,
- record a program from the HP41C to HDRIVE1 with WRTP,
- turn off the HP41C (no need to physically disconnect it),
- read the program in Emu41 from HDRIVE1 with READP.

The procedure is very easy and convenient thanks to the 41C AUTOIO mode.

- Note:

If an external HP41C is used at the same time that the internal VIDEO device (or a external display or printer), then it is mandatory to turn off the HP41C (but no need to physically disconnect it) before Emu41 can recover the control of the internal devices. This is due to the fact that the HP41C, even in idle state (light sleep) still continuously access the loop (IDY messages) when a display or printer device is on the loop.

5. Limitations and known bugs:

5.1 Limitations:

- BEEP and TONE commands are useless.
- It is not possible to keep a key down to see the name of the function followed by "NULL".
- Auto shutdown is not supported.
- Alarm functions of time module don't work, as well as clock display (CLOCK or shift-on sequence).
- the IR printer module is not supported
- at Emu41 startup, the W&W 64K RAMBOXII always starts with block A enabled.

5.2 Known bugs:

None...contact Author if you find unexpected features...

6. Thanks:

I would like to thank some people:

- first, Hewlett-Packard for this wonderful little machine named HP-41C,
- W.C.Wickes, for his book "Synthetic Programming ..." which makes me really discover the HP-41C world,
- PPC Toulouse (France), and Jean-Daniel Dodin for his works in the '80s,
- more recently, Emmanuel Compes for his enthusiasm and his help in this project,
- Christoph Klug for suggesting me to add the external HP-IL support, and pushing me to support the RS232 link (I was quite reluctant to it...)
- and all the people who are still playing or working with this machine, and who encouraged me.

Author:

EMU41 was written by Jean-Francois Garnier

For more information or any update, please visit my home page at:

<http://membres.lycos.fr/jeffcalc>

J-F Garnier, December 2007.