HP41 & EMU41
Uploading plug in module ROM-image-files

HP41 & EMU41
Uploading plug in module ROM-image-files

This document gives you some details about uploading ROM-images (plug in modules) from HP41 to EMU41 by using **ZENROM** or **HEPAX** or **W&W RAM-Box** or **ERAMCO RAM-Box** or **MBK-ProfiSET**. Two appendixes gives knowledge about ROM image file format conversion between HEPAX, MBK-ProfiSET and RAM-Box. Two appendixes describes the generation of image files which are needed for programming **CLONIX** plug in modules and burning **ZEPROM** plug in modules. Last not least you get some details about creating back up, about storing ROM-image files on tape and about email transfer.

EMU41 virtual loop configuration:

First check and set the virtual loop configuration of EMU41: Open the EMU41.INI file and check / set / save the following configuration:

DISPLAY	1. Device
LPRTER1	2. Device
HDRIVE1	3. Device
HDRIVE2	4. Device
;FDRIVE1	not used
DOSLINK	5. Device
XIL 700 (or 300)	HP-IL/PC interface card

Wire HP41 to PC (IL-Module and HP-IL/PC interface card). Start EMU41. Switch on the real handheld and execute AUTOIO, SF 15. CCD-Module list you the devices above with CAT 0 to EMU41 screen display.

Part I : Working with ZENROM

The following parts are needed: HP41, IL-Module, Zenrom, (CCD Module), HP-IL/PC interface card, EMU41 full version. Possible HP41 port configurations are:

ZENROM	
ROM	IL-Module

or with CCD-Module :

CCD	ZENROM
ROM	IL-Module

If you work with CCD and Zenrom, deactivate the low letter mode of the CCD-Module as described in the CCD-Module manual (software routine TLC).

ZENROM disassembler test :

Execute 1 SELECT MANIO SF 15 and start the Zenrom MCode editor with XEQ MCED. After the COMMAND? prompt press the ENTER key for activating the disassemble mode : You get the following HP41 screen display

DIS : _ _ _ _ , _ _ _ _

Enter the start address of the page you want to disassemble, for example the lower page of port 3

DIS : C000 , _ _ _ _

Now enter the stop address for a 4Kbyte module :

DIS : C000 , CFFF

Or the stop address for a 8Kbyte module :

DIS : C000 , DFFF

Start disassembling by pressing R/S key. Now you get a test listing on EMU41 screen display. Stop listing by pressing R/S again. Now you get the COMMAND? prompt. Terminate MCED by pressing the ON key.

Writing inside a DOS text file:

Now we disassemble the module data inside EMU41OUT.DAT text file: 5 SELECT MANIO activates the DOSLINK of EMU41. Again start the disassemble mode with SF15 XEQ MCED After the COMMAND? prompt press ENTER and input start and stop address of the module :

DIS : _ _ _ _ , _ _ _ _

After pressing R/S the procedure runs, but you get no EMU41 screen response about this (refer also to Appendix IV). Remember: Disassembling a 4KByte page or a 8KByte page takes some time. Therefore wait until transfer is finished completely and you are presented again the COMMAND?

prompt. Then stop MCED by pressing the ON key. Deactivate your handheld and EMU41. Open the EMU41 folder and open the EMU41OUT.DAT file with any text editor. Store this file in same folder under NAME.LST. Open NAME.LST and edit the text :

```
MCED
DIS : C000,CFFF
C000  005   XROM 05
C001  00D  FCNS 13
            :
            :
```

Remove the first two leading lines, only the start address and following text lines are used. Save the file and close the editor. Now finish work with Part III.

Part II : Working with HEPAX

Also by using the HEPAX module it would be easy possible to create a disassembled ROM-image . LST file. The following parts are needed: HP41, IL-Module, HEPAX, HP-IL/PC interface card, EMU41 full version.

HEPAX	
ROM	IL-Module

HEPAX disassembler test :

Start EMU41, then switch on your HP41. First we disassemble to the EMU41 screen: Execute AUTOIO and start the HEPAX disassemble mode with XEQ DISASM. You get the following display message :

DIS : _ _ _ _

Enter the start address of the lower page of port 3 (= page C) :

DIS : C000 – _ _ _ _

Now HEPAX generates a new prompt for entering the stop address. For a 4Kbyte module enter:

DIS : C000 – CFFF

For a 8Kbyte module enter :

DIS : C000 – DFFF

HEPAX starts the process automatically. After looking to the screen display stop the listing by pressing R/S.

Writing inside a DOS text file :

Now we disassemble to a DOS file: Select the EMU41OUT.DAT text file by pressing 5 SELECT MANIO , this activates the DOSLINK of EMU41. Again execute XEQ DISASM and enter start and stop address :

```
DIS : C000 – CFFF
```

During the writing into the EMUOUT.DAT file you get no EMU41 screen response about this (refer also to Appendix IV). Remember: Disassembling a 4KByte page or a 8KByte will take some time. Therefore wait until transfer is finished completely. Then deactivate your handheld and EMU41. Open the EMU41 folder and open the EMU41OUT.DAT file with any text editor. Store this file in same folder under NAME.LST. Open NAME.LST and see the text :

```
C000  005   XROM 05
C001  00D  FCNS 13
           :
           :
```

There is no need to edit the HEPAX output. Close the file.

Now finish the work with Part III.

Part III : Generating a .BIN file

With help of ZENROM or HEPAX we create the NAME.LST file. For generating a .BIN file we need the MS DOS prompt. Jump to the EMU41 folder with DOS command C:/ CD EMU41. Than execute the following command line :

```
clbin41   name.lst   name.bin
```

Close the MS DOS window and open the EMU41 folder. Now you find the new NAME.BIN file. Size is 5,00KB for a 4Kbyte ROM-image-file or 10.00KB for a 8Kbyte ROM-image-file. Move the NAME.LST file to an other folder (for example HP41 modules) and delete the contents of the EMU41OUT.DAT file.

Installing the virtual ROM to EMU41 :

In the last step we install the new virtual ROM to EMU41. For this open the EMU41.INI file with a text editor. Add the following line :

```
; X  4  NAME.BIN  ; comment text which includes the XROM Number
```

for a 4KByte ROM-image file, or

```
; X  8  NAME.BIN  ; comment text which includes two XROM Numbers
```

for a 8Kbyte ROM-image-file. X is the port number and runs from 8 up to 15. By removing the leading ; you activate the virtual module for EMU41. Avoid doubling identical XROM numbers and do not place two modules on the same port address. Save the EMU41.INI file.

Start EMU41 AUTOIO (press F8 F4 AUTOIO F4) SF15 (press F5 7 15) and execute CAT 2 (press F5 N 2). Now the actual plug ins are listed. For generating .ROM files now jump to Appendix III....

Part IV : Working with W&W RAM-BOX

Alternatively to Part I (ZENROM) and Part II (HEPAX) you can use a RAM-BOX. Then the ROM-image file upload to EMU41 is much easier and faster. The following parts are needed: HP41, with W&W RAM-Box (or HP41CY), IL-Module, HP-IL/PC interface card, EMU41 full version. Possible HP41 port configurations with W&W Ram-Box or HP41CY are :

W&W OS Page 9	Page A (=10) Page B (=11)
Page C + D IL-Module	Page E (=14) Page F (=15)

Write the ROM-image file on virtual drive :

Check the first virtual drive of EMU41 by executing SF15, AUTOIO, 3 SELECT (=HPRIVE1) and DIR (= CAT 7 with CCD-Module). This lists the existing files stored on the drive to the EMU41 screen. Now enter a NAME for your ROM-image-file to the ALPHA-Register.
For writing/saving the lower Port 3 with W&W RAM-BOX you enter 12 to X-Register and execute WRTPG. Enter 13 to X-Register for writing/saving the upper Port 3 and execute WRTPG.

DIR (CAT 7) shows you the new 640 register ROM-image file on the virtual drive. Now finish the work with Part VI.

Part V : Working with ERAMCO RAM-BOX

Alternatively to Part I (ZENROM) and Part II (HEPAX) you can use a RAM-BOX. Then the ROM-image file upload to EMU41 is much easier and faster. The following parts are needed: HP41, ERAMCO RAM-Box, IL-Module, HP-IL/PC interface card, EMU41 full version. Possible HP41 port configurations with ERAMCO Ram-Box are :

Page +1 Page -1	Page +2 Page -2
Page +3 / -3 IL-Module	RSU 1 OS RSU 2 OS

Write the ROM-image file on virtual drive :

Check the first virtual drive of EMU41 by executing SF15, AUTOIO, 3 SELECT (=HDRIVE1) and DIR (= CAT 7 with CCD-Module). This lists the existing files stored on the drive to the EMU41 screen. Now enter a NAME for your ROM-image-file to the ALPHA-Register.

For writing/saving the lower Port 3 with ERAMCO RAM-BOX you enter +3 to X-Register and execute WRTRAM. Enter -3 to X-Register for writing/saving the upper Port 3 and execute WRTRAM.

DIR (CAT 7) shows you the new 640 register ROM-image file on the virtual drive. Now finish the work with Part VI.

Part VI : Working with Virtual RAM-BOX

For using ROM-image files, stored on virtual drive, use the advanced W&W RAM-BOX II operating system (= HP41CY), which includes two software selected 32 KByte banks = 64Kbyte RAM. Open the EMU41.INI file and activate the W&W RAM-BOX II operating system on port 8 :

```
8  64   ramboxii.dat   ;  W&W RAMBOX II   XROM 31   64KByte = HP41CY
```

This file includes also the belonging RAM. Save the EMU41.INI file. Start EMU41, execute SF15, AUTOIO, 3 SELECT, enter the NAME of the ROM-image-file on the first virtual drive to ALPHA-Register, enter 12 to X-Register and execute READPG. Now CAT 2 lists you the imported module, and you are able to execute functions of the new virtual plug in module.

When using the virtual 64KByte RAM-Box II, you have no direct access to selected pages, by opening the EMU41.INI file. For creating .BIN (or .ROM) files work with the following and more basic EMU41 configuration : Open the EMU41.INI file and activate the 32KByte W&W RAM-BOX operating system on port 8. Activate the MLDLRAM.DAT file to port 12. Copy the MLDLRAM.DAT file, rename it to MLDLRAM.OLD and store it to the EMU41 folder :

```
8    4  rambox.bin          ;  W&W RAMBOX   XROM 31   32KByte
12  8  MLDLRAM.DAT   ;  8KByte RAM
```

Than save the EMU41.INI file. Now your have available a virtual RAM-Box with only 8KByte RAM on Port C. Start EMU41 and enter the NAME of the ROM-image-file on the virtual drive to ALPHA-Register. Enter 12 to X-Register and execute READPG. Now CAT 2 lists you the imported module, and you are able to execute functions of the new virtual plug in module.

Save the ROM-image-file as .BIN file :

Enter the EMU41 folder and rename the MLDLRAM.DAT file to NAME.BIN file. Than rename the existing MLDLRAM.OLD file to MLDLRAM.DAT file. Open the EMU41.INI file deactivate the MLDLRAM.DAT file on port 12 and add the new NAME.BIN file to the list :

```
; X  4  NAME.BIN  ; comment text which includes the XROM Number
```

for a 4Kbyte ROM-image. In this case the upper 4KByte block is not used / needed. Or enter

```
; X  8  NAME.BIN  ; comment text which includes the XROM Number
```

for an 8Kbyte ROM-image-file. Activate the new .BIN file on port 12 by removing the leading ; you activate the virtual module for EMU41. Avoid doubling identical XROM numbers and do not place two modules on the same port address. Last save the EMU41.INI file. Start EMU41 and work with the new virtual plug in module application.

Compared with Part I....Part III (ZENROM or HEPAX) the process of renaming the MLDLRAM.DAT file is an easy and fast solution for creating .BIN files. For generating .ROM files now jump to Appendix III....

Part VII : Working with MBK-ProfiSET

May be you have the MBK-ProfiSET available : This RAM-BOX product, created by Mathematisches Büro Köln / Germany is housed inside an small external box, and includes a 8KByte operating system (EPROM) plus 16KByte or 32KByte RAM . For the ROM-image file upload to EMU41 the following parts are needed: HP41, MBK-ProfiSET, IL-Module, HP-IL/PC interface card, EMU41 full version. By setting the left rotary switch of the MBK-ProfiSET, you place the internal operating system to port 4 = page E + F. Insert the adapter terminal of the MBK-ProfiSET to port 1 of your HP41. Insert the IL-module to port 2 or port 3 of your HP41. Now you have available the following Block configuration :

Block 1	Block 3
Block 2	Block 4
Block 5	MBK OS
Block 6	MBK OS

Write the ROM-image file on virtual drive :

Check the virtual drive of EMU41 by executing SF 15, AUTOIO, 3 SELECT (= HDRIVE1) and DIR. This lists the existing files stored on the first virtual drive to the EMU41 screen. Now enter a NAME for your ROM-image-file to the ALPHA-Register.

For writing/saving Block 3 with the MBK operating system, load the Block number to X-Register and execute SLCT**.** Enter the file name to ALPHA-Register and execute SAVEBL or GETBL. DIR lists you the new 640 register ROM-image file on the virtual drive. Now finish the work with the following Part VIII :

Part VIII : Working with virtual MBK-ProfiSET

Open the EMU41.INI file and activate the MBK-ProfiSET operating system on port 8. Activate the MLDLRAM.DAT file to port 10. Copy the MLDLRAM.DAT file, rename it to MLDLRAM.OLD and store it to the EMU41 folder :

```
8   8  mbkprof.bin        ;  MBK PROFISET  XROM 27 + XROM 31 version 2G
10  8  MLDLRAM.DAT  ;  8KByte RAM
```

Than save the EMU41.INI file. Now your have available a virtual MBK-ProfiSET with small 8KByte RAM on page 10 (= Block 3) and page 11 (=Block 4). Start EMU41 and enter the NAME of the ROM-image-file on the virtual drive to ALPHA-Register. Execute 3 SLCT followed by GETBL. Now CAT 2 lists you the imported 4KByte module, and you are able to execute functions of the new virtual plug in module.

Appendix I : ROM file conversation HEPAX / RAM-Box

The RAM-Box ROM file format of W&W and ERAMCO is identical and also compatible to the ZEPROM file format (command ILBURN). Unfortunately this format is **not** compatible to the HEPAX ROM-file format.

With help of EMU41 it is possible to convert ROM-files between the RAM-Box and HEPAX format. Install the W&W RAM-Box operating system and the HEPAX operating system to EMU41 by editing EMU41.INI file :

```
8   16  hepax1d.bin  ;  hepax          XROM 07 bank switched module
11  4   rambox.bin   ;  rambox w&w     XROM 31
```

When starting EMU41, HEPAX places the belonging HEPAXRAM.DAT file automatically to port 8 and port 9, the operating system is placed to port 10. It is possible to use the HEPAX RAM on port 8 and port 9 also with the W&W RAM-Box operating system.

Therefore you are able to transfer ROM-files between page 8 page 9 and the virtual drive of EMU41 by executing the HEPAX commands WRTROM or READROM or the W&W RAM-Box commands WRTPG and READPG. Place the filename to Alpha-Register and the port number (in this configuration 8 or 9) to the X-Register. Than execute one of the four commands.
Last you transfer the ROM-files from the virtual drive to your HP41 handheld with RAM-Box or HEPAX.

Appendix II : ROM file conversation MBK-ProfiSET / RAM-Box

Like Appendix II, the MBK-ProfiSET ROM file format is different from the common W&W and ERAMCO file format, which is also compatible to the ZEPROM file format (command ILBURN).

With help of EMU41 it is possible to convert ROM-files between MBK-ProfiSET and RAM-Box format. Install the W&W RAM-Box operating system, the MBK-ProfiSET operating system and the MLDLRAM.DAT file to EMU41 by editing EMU41.INI file :

```
8    4   rambox.bin       ;  rambox w&w  XROM 31
14   8   mbkprof.bin      ;  MBK PROFISET  XROM 27 + XROM 31 version 2G
10   8   MLDLRAM.DAT  ;  8KByte RAM
```

Now you are able to transfer ROM-files between page 10 (= Block 3), page 11 (=Block 4) and the virtual drive of EMU41 by executing the W&W RAM-Box commands WRTPG or READPG and the MBK-ProfiSET commands SLCT plus SAVEBL or GETBL : Place the filename to ALPHA-Register and the port or Block number to the X-Register. Than execute the commands listed above.

Last you transfer the ROM-files from the virtual drive to your HP41 handheld and use them with your real RAM-Box or MBK-ProfiSET.

Appendix III : ROM file conversation .BIN / .ROM

The former version of EMU41 from Jean-Francois Garnier works with ROM-image files stored in the .bin format. V41 from Warren Furlow works with the different .rom format. His web page www.hp41.org contains some .rom files for downloading. Actual Jean-Francois Garnier added the .rom format to EMU41, and you are able to use both formats in parallel. Now EMU41 includes the updated CLBIN41.EXE software tool (version V1.8 JFG 2004) for converting 4KByte .bin files to .rom files :

```
clbin41     name.lst    name.bin   =  convert .lst file into .bin file
clbin41     null        name.bin   =  creates empty .bin file
clbin41 /r  name.rom  name.bin     =  convert unpacked .rom file into packed .bin file
clbin41 /b  name.bin   name.rom    =  convert packed .bin file into unpacked .rom file
clbin41 /?                         =  view help text  and software version
```

.bin files need less memory space and include 4KByte or 8KByte ROM-images and advanced bank switched ROM applications. .rom files need more memory space and contain only 4KByte ROM-images. For programming CLONIX you also need the .rom format...

Appendix IV : Creating a user code software ROM for EMU41

May be you want saving main memory by storing your own user code program applications to a 4KByte ROM-image page. For creating this, EMU41 plus virtual RAM-Box are the needed software tools : Transfer your user code program solutions from HP41 to the virtual drive of EMU41 (AUTOIO, 3 SELECT, WRTP) and load them to EMU41 main memory (AUTOIO, 3 SELECT, READP). Next install the virtual RAM-BOX by editing the EMU41.INI file :

```
8    4   rambox.bin          ;  RAMBOX W&W  XROM 31
10  8   MLDLRAM.DAT  ;  8KByte RAM
```

Working with RAM-Box : First initialise 4KByte page 10 with XROM number. A list of XROM numbers you find in the EMU41.INI file of the author. Avoid doubling XROM numbers with your configuration. Enter a page name to ALHPA-Register. This page name is the module name and listed with CAT 2. Load the XROM number to Y-Register and page number 4 to X-Register. Now execute INITPG.

Next load your user code programs from main memory to page 10. Enter the program name in the ALPHA-Register and page number 10 in X-Register. Execute LDPGM. The FAT of your 4KByte module application accept maximally 63 commands = global ALHPA LBL´s. Therefore replace not needed global LBL´s of your software routines by numerical LBL´s before uploading to RAM-Box. Next enter the program name in ALPHA-Register and execute XQ>XR. This transforms the XEQ commands and jump distances to an fast speed XROM command. Remove the uploaded programs from main memory.

List your virtual plug in module application with CAT 2 and execute some software tests inside EMU41. If anything works fine, calculate the check sum or your module application : Enter the page number 10 to X-Register and execute PGSUM. Close the page by entering the page number 10 to X-register, a four letter ROM version to ALPHA-Register and execute ENDPG.

If you want to alter your ROM page later – do not execute ENDPG. By entering the page number 10 to X-register and executing CLLSTFL you are able to remove the last program from your ROM page. When your virtual user code software module is finished, copy the MLDLRAM.DAT file and rename it to NAME.BIN for former use with EMU41.

Appendix V : Creating a user code software ROM for CLONIX

The new CLONIX plug in module from Diego Diaz is a fair price solution for HP41 users, who do not own a RAM-Box, HEPAX or Zeprom. CLONIX includes 6 x 4KByte ROM pages. Below you find an advanced HP41 port configuration, the CLONIX contains 6 ROM-image files, programmed from pages 8 to page 13.

CLONIX	Double X-Memory
IL-Module	free

For programming a CLONIX, you need a PC (WIN98 or WIN2000/XP) and the special programming adapter (interfaced to the RS232 COM port) and software from Diego Diaz. His detailed user manual describes the programming procedure including some screen displays. By loading/programming ROM-image files to CLONIX now you also get high priced or exotic plug in solutions like CCD, ZENROM, PPC, ADVANCED, EXT-I/O, EXT-IL, SKWIDINK.... Some of this .rom files you find on Warren Furlows web page www.hp41.org . The conversion of existing 4KByte .bin files to .rom files (needed for programming CLONIX) is described in Appendix III.

May be you want saving main memory of your HP41 by storing your own user code program applications to a CLONIX page. Than you need a special .rom file containing your own software application. For creating this, EMU41 plus virtual EXT-I/O and virtual RAM-Box and are the needed software tools. Start with Appendix IV, as result you get a MLDLRAM.DAT file on port 10 containing your virtual module application.

The MLDLRAM.DAT files have a size of 8KByte, containing the lower 4KByte page and the upper 4KByte page. We are able to rename the MLDLRAM.DAT to a 8KByte .bin file – but it is not possible to convert this to a .rom file (Appendix III) because .rom files limited to 4KByte pages. Therefore we need a software tool for extracting the lower and upper 4KByte page from the MLDLRAM.DAT file. Enter the following user code software to EMU41 main memory :

```
01  LBL LOWER
02  “DOSLINK”
03  FINDID
04  SELECT
05  “LOWER 4K”
06  AVIEW
06  256
08  20
```

```
22  “UPPER 4K”
23  AVIEW
24  256
25  20
26  LBL 02
27  INAN
28  CLA
29  DSE Y
```

```
09  LBL 01
10  INAN
11  OUTAN
12  DSE Y
13  GTO 01
14  CLD
15  CLA
16  CLX
17  RTN
18  LBL UPPER
19  “DOSLINK”
20  FINDID
21  SELECT
30  GTO 02
31  256
32  20
33  LBL 03
34  INAN
35  OUTAN
36  DSE Y
37  GTO 03
38  CLD
39  CLA
40  CLX
41  END
```

Now start extracting 4KByte blocks from the MLDLRAM.DAT file : First clear the contents of the EMU41OUT.DAT file. Second rename and save the EMU41in.DAT file. Third copy and rename the MLDLRAM.DAT file to EMU41in.DAT file. Fourth start EMU41, press TAB for activating the high speed mode, and execute LOWER for extracting the lower 4KByte block from EMU41in.DAT file to EMU41OUT.DAT file. Alternatively execute UPPER for extracting the upper 4KByte block from EMU41in.DAT file to EMU41OUT.DAT file. Last close EMU41 and copy and rename the EMU41OUT.DAT file to NAME.BIN file.
Size of the new NAME.BIN file is 5,00KB for a 4Kbyte ROM-image-file. Move the NAME.LST file to an other folder (for example HP41 modules) and delete the contents of the EMU41OUT.DAT file. Next we convert the NAME.BIN file to a .rom file by using the MS DOS prompt command line :

```
clbin41 /b  name.bin   name.rom
```

Close the MS DOS window and open the EMU41 folder. Now you find the new NAME.ROM file. Size is 8,00KB for a 4Kbyte ROM-image-file. Transfer the .rom file to the CLONIX programming tool folder and start programming procedure as described in the user manual of Diego Diaz.

Appendix VI : Creating a user code software ROM for ZEPROM

May be you want saving main memory of your HP41 by storing your own user code program applications to a ZEPROM page. First go to Appendix IV and finish your ROM application by using EMU41 and virtual RAM-Box. By burning ROM-image files to ZEPROM now you also get high priced or exotic plug in module solutions like CCD, ZENROM, PPC, EXT-I/O, EXT-IL, SKWIDINK.... Some of this .rom files you find on Warren Furlows web page www.hp41.org .

When your own ROM application is finished and tested with virtual RAM-Box of EMU41, you store the complete ROM image to virtual mass storage device of EMU41, for example HDRIVE1 : Enter the page number 10 (lower 4KByte page) or 11 (upper 4KByte page) to X-register, the module name in ALPHA-Register and execute WRTPG which saves the image.

Now you are able to copy the 640 register file from virtual tape to a real RAM-Box by using the READPG (W&W) or READRAM (Eramco) command. Alternatively copy the file to a real IL-Mass storage drive, by using the COPYFL command of the EXT-I/O module.

For burning a ZEPROM you execute the ILBURN command which is part of the PROGRAMMER ROM. Insert a ZEPROM with PROGRAMMER ROM application to port 1 of your HP41. Place the IL-module to port 2 of your HP41 for controlling the IL-mass storage drive. Insert the ZEPROM VOLTAGE CONVERTER = ZVC (One AT A TIME PROGRAMMER) burning tool to port 3 of your HP41. Port 4 is free. Place a second, with UV light erased ZEPROM to the ZVC. Because the needed supply current for the burning process is high, only use a new set of Lady batteries inside your HP41. The HP41 accumulator pack is not qualified for burning ZEPROMS.

PROGRAMMER ROM	IL-Module
ZVC ZEPROM	free

When you pull above the small switch of the ZVC, the ZEPROM is routed from page 12 up to page 15 of your HP41. Now you are able to burn 4 x 4KByte pages = 16KByte. For this enter the page number 12…15 to X-register, the mass medium file name to ALPHA-Register and execute ILBURN.

When complete burning process is finished, press down the switch of the ZVC tool. Last run CAT 2 and you will see the listing of your ROM image files.

Appendix VII : Writing inside DOS file with EMU41 screen response

May be you prefer to get a real-time screen response during uploading great files (like a disassembled ROM-image-page) inside a DOS text file, as we know this from TRANS41 software.

As Jean-Francois Garnier told me, this is also possible with EMU41 : At DOS prompt start EMU41 with the /lg extension (= logger). Than EMU41 automatically generates a DISP.LOG file and transfers the screen output also to this file. When the data upload is completed leave EMU41 (press F1) and open the DISP.LOG file which contains the data.

Appendix VIII : Storing ROM-image files on cassette tape

If you want to store ROM-image files with RAM-BOX on tape (IL-Cass-Drive) execute NEWM 027. After writing 12 ROM-image files on tape add a 300 register data file with CREATE, named ENDTAPE. Now you are able to store 12 more ROM-image-files on the second track of the tape.

Appendix IX : Back up and email transfer of ROM-image files

The actual version of EMU41 now includes two virtual mass storage drives HDRIVE1 and HDRIVE2. For creating a back up copy from your real tape or disk use the EXT-I/O commands MCOPY or COPYFL. Now this advanced functions are also run correctly with the virtual drives of EMU41.

For easy and fast exchange of HP41 ROM-image files by email copy the HDRIVE1.DAT or the HDRIVE2.DAT file from the EMU41 folder (= virtual tape or disk) and transfer them to other HP41 enthusiasts.

Appendix X : EMU41 and HP82161 IL-Cass Drive

Activate the slow down mode of EMU41 (press SHIFT and TAB) when using the real IL-Cass Drive. Without the slow speed mode of EMU41 sometime you get a timeout failure (and EMU41 hang up) during the IL-Cass Drive searches the tape for detecting files.

The slow speed mode of EMU41 suppresses any hang up effect of EMU41. Although in slow speed mode, the file exchange between EMU41 and IL-Cass Drive needs unknown long transfer times, the file exchange would be finished correctly.