

## HP Pioneers ROM Dumping

(Tested case: Create a HP-27S ROM image file to run the emulator on Windows)

### Contents

Machine code programming (Memory Editor) .....	2
HP17BII and similar machines .....	2
Pioneer (17BII, 27S, 42S) ROM image download .....	3
HP-42S: New Facts .....	4
ROM dump .....	4
HP17B/17BII/19BII/27S/28S/32SII/42S ROM dumping .....	5
CPROMUPL ROM uploading tools.....	5
Content of CPROMUPL archive.....	5
How to make a HP17B/17BII/27S/42S ROM image .....	5
1) General .....	5
2) Memory scanner .....	5
3) Required equipment .....	6
4) Prepare the HP-48 Receiver calculator.....	7
5) Prepare the Pioneer calculator .....	7
6) Transfer the ROM image from the Pioneer calculator to the HP48.....	10
7) Getting the final ROM image on the PC .....	11
Dumping Clamshell calculator ROM images.....	13
1) General .....	13
2) HP28C/S.....	14
3) HP48 .....	19
4) Transfer the ROM image from the HP28C/S to the HP48 .....	19
5) Transferring the data from the HP48 to the PC .....	20
6) Getting the final ROM image .....	20
Transferring the ROM image in smaller parts .....	21
Test results .....	23
HP-48 Serial interface .....	24
HP-48 serial port connector pin out.....	24
HP-48 to PC serial cable.....	24
Testing the Serial cable .....	25
Transferring and Running Files .....	25

## Machine code programming (Memory Editor)

Machine code programming on the HP-17B, 27S, and 42S.

### HP17BII and similar machines

This machine has a **memory editor** just like the 48 does (so does the 17, 27, and 42.)

You get in the same way:

**ON+4th** menu key, then <- to enter editor.

Press . immediately after entering to show your version number.

Keys are:

- **^v** Move thousands
- **x /** Move hundreds
- **+ -** Move units
- **0-9 and menu keys A..F** Enter hex
- **PRT** sends line to IR (potentially to 48 running INPRT, then to PC for disassembly)
- **.** Executes at the current address (Call instruction)

Note that **Execute** is a call, doing a RTN goes back to the editor from your ML code.

Basic architecture:

- 64KB (128K nybbles!) of ROM at 00000
- Display at 40000
- 8K of RAM at 50000

The first part of RAM changes on the fly, so around 51000 or 52000 is good for experimental code.

One can do small machine language programs simply by writing the code on the PC, using SASM on it, and typing in the hex digits in the .L file.

If one creates a Solver equation with a recognizable pattern, then scans from 54000 (end of RAM) back-wards, one can easily find it.

This is a way to enter ML without wrecking normal memory.

They are RPL machines internally; could you write system RPL as well as machine code?

## Pioneer (17BII, 27S, 42S) ROM image download

Message #1 Posted by [Christoph Giesselink](#) on 22 May 2003, 5:59 p.m.

I'm wondering if there are any interests in a quicker download of ROM images from the High End Pioneer calculators HP17B, HP17BII, HP27S, HP42S.

The ROM dump way from **Jean-Francois Garnier** published in [HP-42S: New Facts](#) is quite slow and needs lot of manual work.

My way needs also a HP48 as receiver with 70KB or 103KB (depends on the ROM image size (64KB/96KB)) of free memory, so a HP48G+, HP48GX or any other HP48 with merged 128KB RAM card is necessary.

Further you need of course the calculator which ROM you want to copy.

The trick for the faster download is that I use a binary data capable receiver program on the HP48 and machine written code on the Pioneer calculator. This machine code must be entered by the **integrated memory scanner** and may destroy the current RAM content. It takes about 5 - 10 min. to enter and verify this code.

**The program (~50 bytes) needs the ROM point entry of the IR transmitter function.** So far I know the entry of:

- HP17B rev. B
- HP17BII rev. B
- HP27S rev. B
- HP42S rev. A
- HP42S rev. B
- HP42S rev. C

The automated download from the Pioneer calculator to the HP48 takes about 16 - 17 min. for a 64KB ROM image. Finally you have to transfer the captured string from the HP48 to a PC and decode the string on the PC with a command line program into the final ROM image.

The whole process without searching for the ROM entry on unknown software revisions takes about 40 - 45 min, most time automated.

The battery consumption is quite low on both calculators comparing to Jean-Francois way. ROM images are necessary if you want to explore the operating system internals or you want to work with an emulator.

## HP-42S: New Facts

Posted by [J-F Garnier \(France\)](#) on 10 Apr 2002, 4:32 p.m.

### ROM dump

In order to study the HP-42S internals, or to even think about an emulator, the first thing to do is to get a ROM image.

**Fortunately, there is a way to do it thanks to the internal memory browser.**

You will need:

- A HP-42S
- A HP48 (with the INPRT program from HP)
- A PC.

Here is the procedure:

- enter the test mode on HP42S: EXIT-LOG
- enter the memory browser on HP42S: <- (back arrow)
- here, you can check your software version: press '.' (decimal point). Version is displayed at the upper left (A, B or C). At the upper right, you can see the **last warm start log** (similar to the WSLOG command on HP48 serie)
- use the ^, v (up and down arrows), /, \*, +, - keys to go to address 0
- now, align your HP42S and HP48 IR leds
- start INPRT on the HP48
- press the **COS** key on HP42S and keep it down
- monitor the address displayed on HP42S, and release the COS key when reaching 1000h (you can release the COS key a little bit before and add a few keystrokes to stop just at 1000h)
- wait for INPRT timeout (about 1s)
- now, you have your first 1000h nibbles on the HP48! Store this string in an object, and transfer it to the PC.
- Continue up to the end of the ROM (1ffffh). I recommend to transfer by 1000h blocs for convenience, so you will need 32 transferts. Be patient, it takes a lot of time ...
- To leave the HP42S test mode, press EXIT-Sqrt

## HP17B/17BII/19BII/27S/28S/32SII/42S ROM dumping

(c) Christoph Gießelink

<http://hp.giesselink.com/emu42.htm>

## CPROMUPL ROM uploading tools

A collection of tools for uploading the necessary ROM images very quickly comparing to other methods. Necessary hardware:

- The calculator with the ROM you want
- A HP calculator as a receiver
  - A HP48GX/G+ for a Pioneer
  - A HP48G/GX/G+ for a Clamshell image upload as receiver.
  - The necessary free size of memory on the HP48 depends on the size of the ROM image you want to upload.

## Content of CPROMUPL archive

<b>CLAMSHEL.TXT</b>	Document how to make a HP28C/S ROM image upload.
<b>PIONEER.TXT</b>	Document how to make a HP17B/17BII/27S/42S ROM image upload.
<b>BINPRT.ZIP</b>	IR receiver software for the HP48 (This is a modified INPRT version).
<b>CAPC2BIN.ZIP</b>	Conversation tool to convert captured into binary ROM data.
<b>CHAMPCHK.ZIP</b>	Checksum checker for Centipede chip based calculators.
<b>LEWISCR.C.ZIP</b>	CRC checker for Sacajawea/Lewis chip based calculators.

## How to make a HP17B/17BII/27S/42S ROM image

### Dumping High End Pioneer calculator ROM images

#### 1) General

This is a ROM dump utility for uploading the HP17B, HP17BII, HP27S or HP42S ROM image to a PC. The only useful ability to do this without opening the calculator is to use the infrared output port. But here we have a problem; the used printer protocol format isn't compatible with standard PC redeye receivers. So you must have a special hardware for your PC or use a HP48 with a special program derived from the **INPRT** program to capture the data. Because PC redeye receivers for the HP redeye protocol are very rare and the **HP48** is more common, I assume that you use a HP48 as receiver.

This procedure destroys or corrupts your calculator memory content!

#### 2) Memory scanner

All of these calculators have a build in memory scanner.

Legend:

SK = Soft Key -> SK1 is the top left key ... SK6 is the top right key

A = SK1, B = SK2, C = SK3, D = SK4, E = SK5, F = SK6

Enter the memory scanner with

**EXIT/ON + SK4**  
**<- (backspace)**

Commands (incomplete) are:

**^** move by + #1000h nibbles (only in 17BII/42S versions)  
**v** move by - #1000h nibbles (only in 17BII/42S versions)  
**\*** move by + #100h nibbles  
**/** move by - #100h nibbles  
**+** move by + #1h nibbles  
**-** move by - #1h nibbles  
**.** execute current address  
**0-9,A-F** enter a digit into memory, move by + #1h nibbles

Press **EXIT/ON + SK3** to exit.

### 3) Required equipment

#### 3.1) Supported calculators

- HP17B (english or international) ROM Rev. A or B
- HP17BII (english or international) ROM Rev. B
- HP27S ROM Rev. A or B
- HP42S ROM Rev. A, B or C

To check the ROM revision recall the memory scanner of the calculator with

**EXIT/ON + SK4**  
**<-**

You see something like **023F5:710D...** The address and content differs from ROM to ROM.

Press the <.> key to evaluate this address.

In the top left corner of the display you see one or two alphanumeric characters:

- The first character is the **ROM Rev.** of the main ROM.
- Notice this number for use in one of the next steps.
- If you see two alphanumeric characters, which may happen on the HP17B or HP17BII, you have an international version of this calculator. Notice this also, because international versions have a different ROM size.

#### 3.2) Receiver calculator

- A HP48G+, HP48GX or a HP48SX with merged RAM card
- The disadvantage of this method is you need at least **~70KB** (for a normal ROM) or **~100KB** (for an international ROM) of free RAM on the HP48 to hold the captured ROM.
- Check the amount of free RAM with the MEM command.

### 3.3) Receiver program

The **INPRT** program has one big problem, it don't accept **0** characters, so it cannot be used for binary transfers.

**Raymond Del Tondo** made a derivation of this program without this limitation. I called it **BINPRT** and is used on the HP48 for capturing the IR output of the calculators.

### 3.4) Fresh or good batteries in both calculators

Errors can occur due to insufficient current supply.

The complete process takes less than 1 hour.

## 4) Prepare the HP-48 Receiver calculator

Clear the calculator memory and merge a RAM card if necessary to your user memory to get at least ~70KB or ~100KB (depends on the ROM image you want to dump) of free memory. Refer to the calculator manual how to this please.

Transfer the **BINPRT** program from this packet to the HP48. The program itself is a derivation from the original **INPRT** program (c) by Hewlett Packard which allows capturing of binary data. A big thank to Raymond Del Tondo for doing this.

Check the calculator's free memory with the **MEM** command. You must have at least the necessary free bytes before continuing!

## 5) Prepare the Pioneer calculator

The upload program on the calculator itself is written in assembler for maximum speed. Therefore you have to start the memory scanner and enter the binary data of the assembler program into the calculator memory.

A good address for the assembler program is **#52000**.

Start the memory scanner and go to address #52000 with the keys described in section (2).

The following process destroys the memory content!

This is the list file of the dump assembler program.

The program is free for personal use only.

Ask the author for permission in all other cases please.

**You have to enter the digits in the 2nd column just after the address.**

The file has been compiled for HP42S Rev. C ROM image.

If you have a different ROM image or another revision, notice the changes behind the listing.

### 5.1) Upload program

Modify the data at the address positions **0000D** (ROM size) and **0005B** (Send chars over IR) depending on the calculator model to dump the ROM.

Saturn Assembler Dump Lewis Pioneer ROM over IR Fri Feb 13 23:01:03 2004  
V3.0.8 (12/06/2002) ROM42.a Page 1

\* Send character in A[B] over IR  
\*OUTBYT EQU #0133C (HP17B rev. A)  
\*OUTBYT EQU #01378 (HP17B rev. B)  
\*OUTBYT EQU #01445 (HP17BII rev. A/B)  
\*OUTBYT EQU #0136A (HP27S rev. A)  
\*OUTBYT EQU #0142E (HP27S rev. B)  
\*OUTBYT EQU #0318D (HP42S rev. A)  
**OUTBYT EQU #031D6 (HP42S rev. B/C)**

\* no. of 128 byte blocks

**PRGLEN EQU 512 (HP27S/42S) = 64KB / 16 min**  
\*PRGLEN EQU 768 (HP17B/17BII) = 96KB / 24 min

Addr:	Code:	Assembly:	Remarks:
00000	<b>D2</b>	C=0 A	
00002	<b>135</b>	D1=C	ROM Start address
00005	<b>D1</b>	B=0 A	init block no.
00007	<b>E5</b> -	B=B+1 A	inc no. of blocks
00009	<b>7110</b>	GOSUB SBLOCK	send block
0000D	<b>32FF1</b>	LC(3) (PRGLEN)-1	no. of 128 byte blocks <b>(See Note 1)</b>
00012	<b>B39</b>	C=C-B X	
00015	<b>51F</b>	GONC -	
00018	<b>D2</b>	C=0 A	restart
0001A	<b>81B3</b>	PC=C	
0001E	<b>D0</b> SBLOCK	A=0 A	SOH block header
00020	<b>E4</b>	A=A+1 A	
00022	<b>7D20</b>	GOSUB SBYTE	
00026	<b>D4</b>	A=B A	block number
00028	<b>7720</b>	GOSUB SBYTE	send data byte
0002C	<b>D4</b>	A=B A	1' complement of block number
0002E	<b>FC</b>	A=-A-1 A	
00030	<b>7F10</b>	GOSUB SBYTE	send data byte
00034	<b>AF2</b>	C=0 W	reset checksum
00037	<b>34000F7</b>	LCHEX #7F000	loop 128 bytes, chk = 0
0003E	<b>14B</b> -	A=DAT1 B	get data
00041	<b>171</b>	D1=D1+ 2	next address
00044	<b>A62</b>	C=C+A B	add checksum
00047	<b>7800</b>	GOSUB SBYTE	send data byte
0004B	<b>A5E</b>	C=C-1 M	next byte
0004E	<b>5FE</b>	GONC -	
00051	<b>DA</b>	A=C A	send checksum, fall into SBYTE
00053	<b>10B</b> SBYTE	R3=C	save byte counter/checksum
00056	<b>D9</b>	C=B A	save block counter
00058	<b>10C</b>	R4=C	
0005B	<b>8F6D130</b>	GOSBVL OUTBYT	Send char. in A[B] over IR <b>(see Note 2)</b>
00062	<b>11C</b>	C=R4	
00065	<b>D5</b>	B=C A	
00067	<b>11B</b>	C=R3	
0006A	<b>01</b>	RTN	



### 5.2) Program modifications for each calculator model

To use other ROM versions, adjust the lines above with the information below that corresponds to your ROM version.

#### Note 1 – Choose the ROM size

- Normal ROM version (64KB)

**0000D 32FF1**      **LC(3) (PRGLEN)-1**      **no. of 128 byte blocks**

- International ROM version (96KB)

**0000D 32FF2**      **LC(3) (PRGLEN)-1**      **no. of 128 byte blocks**

#### Note 2 – Choose the calculator model

- HP17B ROM Rev. A

**0005B 8FC3310**      **GOSBVL OUTBYT**

- HP17B ROM Rev. B

**0005B 8F87310**      **GOSBVL OUTBYT**

- HP17BII ROM Rev. A + B

**0005B 8F54410**      **GOSBVL OUTBYT**

- HP27S ROM Rev. A

**0005B 8FA6310**      **GOSBVL OUTBYT**

- HP27S ROM Rev. B

**0005B 8FE2410**      **GOSBVL OUTBYT**

- HP42S ROM Rev. A

**0005B 8FD8130**      **GOSBVL OUTBYT**

- HP42S ROM Rev. B + C

**0005B 8F6D130**      **GOSBVL OUTBYT**

### 5.3) Entering the program machine code

Take the 2<sup>nd</sup> column values and enter the data very carefully and verify it.

It's absolutely necessary that you type in the program without any errors before using!  
Any typo will crash you calculator and will force you to begin from scratch.

If you are absolutely sure that you typed in the data correctly, go to the first address of the program (**#52000** if you used my suggestion) inside the memory scanner.



*Picture: Jebem's HP-27S ROM dumping*

## 6) Transfer the ROM image from the Pioneer calculator to the HP48

Put the IR transmitter LED of the Pioneer calculator and the IR receiver LED of the HP48 together.

Therefore the calculator transmitter LED should point to the arrow marker on top of the HP48. The LED's of both calculators should have a similar height.

I got best results with calculator distance of about 5 mm (0.2 inch). Avoid external IR light.

- Now start the **BINPRT** program on the HP48;
- Then start the program on the Pioneer calculator with the <.> key before the timeout of the BINPRT program occur!

The transmitting annunciator on the HP48 should blink now.

The complete upload process takes about 16 min. for a 64KB and 24 min. for a 96KB ROM.



*Picture: Jebem's ROM transfer via IR from HP-27S to HP-48G+*

After the upload the Pioneer calculator is doing a warm start by jumping to address #00000 (it displays a Machine Reset message).

The ROM dump program in the HP42S normally persist the warmstart at program end when you're using my suggested base address.

After the upload process you should finally have on your HP48:

- A **string** at stack level 2
- And **<1d>** or **<1h>** at stack level 1.

The content at stack level 1 is the error result of the BINPRT program.

**Drop** (left shift - backspace) the return value at stack level 1 and then save the string at stack level 2 into a variable (**'ROMIMG' STO**).

## 7) Getting the final ROM image on the PC

### 7.1) Transfer the ROM image to the PC

Now transfer the string (**ROMIMG**) from the HP48 to the PC in **binary mode**.

The time for this depends on the transfer program you use. I prefer **Conn4x** (not for the HP48SX) for this purpose.



Picture: Jebem's HP-27S ROM transfer to a PC using XModem protocol. Windows 7 64-bit running Hyperterminal.

### 7.2) Strip off the HP48 XModem header and protocol information

Run the PC **CAP2BIN** program with:

**CAP2BIN.EXE ROMIMG ROMIMG.ROM**

to strip off the HP48 binary header and protocol information (XModem) inside the uploaded file. The last displayed message should be

"ROM size = 10000h = 64KB" or

"ROM size = 18000h = 96KB"

Or else you got a transfer error on the steps before.



Picture: Jebem's HP-27S ROM - Stripping the extra data from the ROM file at PC.

### 7.3) Check ROM integrity

Finally you should check the internal CRC's of the ROM image with the **LEWISCRC** program by doing:

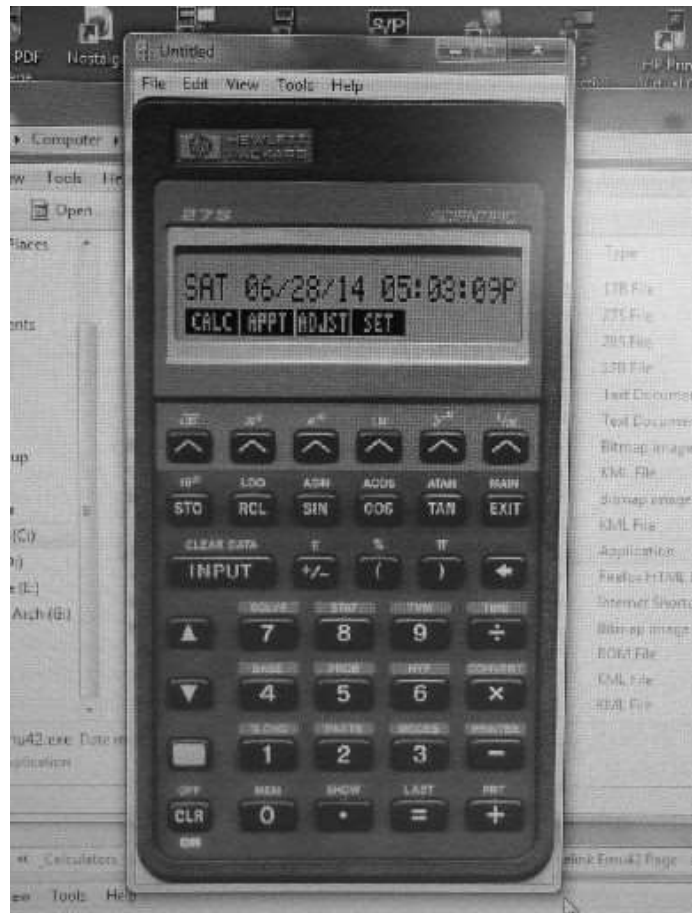
**LEWISCRC.EXE ROMIMG.ROM**

© Christoph Giesselink, Oct. 5th, 2009 c dot giesselink at gmx dot de

```
28-06-2014 16:48 <DIR>
27-08-2004 00:02      32.768 LEWISCRC.EXE
27-08-2004 00:01       1.230 LEWISCRC.TXT
28-06-2014 16:47     65.536 ROMIMG.ROM
               3 File(s)      99.534 bytes
               2 Dir(s)  63.160.516.608 bytes free

G:\Arquivo_Electronica\Service Manuals\_Calculato
MUPL\LEWISCRC>LEWISCRC.EXE ROMIMG.ROM
SACAJAWEA/LEWIS ROM Test U1.00
Unpacking data...
1st Lewis CRC Pass 0 = FFFF
1st Lewis CRC Pass 1 = FFFF
CRC ok!
```

Picture: Jebem's HP-27S ROM - ROM file checksum OK.



Picture: Jebem's HP-27S ROM image - Testing on the excellent Christoph Gießelink's Emu42.

## Dumping Clamshell calculator ROM images

By [Christoph Giesselink](#)

### 1) General

This is a ROM dump utility for uploading the HP28C and HP28S ROM images to a PC. The only useful ability to do this without opening the calculator, is to use the infrared output port. But here we have a problem, the used printer protocol format isn't compatible with standard PC redeye receivers. So you must have a special hardware for your PC or use a HP48 with a special program derived from the INPRT program to capture the data. Because PC redeye receivers for the HP redeye protocol are very rare and the HP48 is more common, I assume that you use a HP48 as receiver.

The following calculators are supported

- HP28C with ROM Rev. 1BB
- HP28C with ROM Rev. 1CC
- HP28S with ROM Rev. 2BB

To check the ROM revision put the calculator in HEX mode and do

# A SYSEVAL

You should get "Version 1BB", "Version 1CC" or "Version 2BB" in the top line of the display. Notice your calculator ROM version to select the corresponding program version later.

You need also

- a HP48 calculator of any type

The greater the amount of free RAM inside the HP48 the easier and faster the ROM upload gets. Best is a HP48GX with a merged RAM card. For uploading the complete ROM we need at least ~133KB (~136000 bytes!) of free RAM on the HP48 to hold the captured ROM by this way. There's a variant of the HP28 upload software allowing the upload in pieces. This reduce the necessary amount of free RAM in the HP48 receiver but makes things more inconvenient and increase the time for the complete upload process. Check the amount of free RAM with the MEM command.

The INPRT program, originally made by HP for capturing infrared printer data, has one big problem, it don't accept 0 characters, so it cannot be used for binary transfers. Raymond Del Tondo made a derivation of this program without this limitation. I called it BINPRT and is used on the HP48 for capturing the IR output of the HP28C/S.

- fresh or good batteries in both calculators

Do you have the necessary equipment? The complete process takes about 1 hour when you have a HP48GX with merged RAM as receiver, longer when you have to do

the upload in pieces. Are you ready? Lets start.

## 2) HP28C/S

The upload program on the HP28C/S itself is written in assembler for maximum speed. There are two program versions available. The first one uploads the complete ROM content, whereas the second one use two arguments specifying the upload range. So it's possible to use receivers with less RAM capabilities which cannot handle the complete ROM content. To get such a program into the calculator you need an external tool. It's absolutely necessary that you type in every program \_without\_ errors before using! Therefore each program here is published with the corresponding checksum in brackets [] behind the program name.

Use the corresponding programs that are belonging to your HP28 version. The programs ASS, MROM28 and MROM28P use internal unprotected functions, they will crash the calculator if your are using the wrong version or if they are typed in wrongly.

This is the absolute necessary checksum program. The number in [] behind the name is the checksum. To verify it, type 'CHK' CHK.

CHK [6C50] for HP28C Rev. 1BB or 1CC

```
<< RCLF STD HEX 64 STWS 48 CF SWAP RCL ->STR 16 STWS
DUP # 0 1 ROT SIZE
FOR j
  OVER j j SUB NUM R->B XOR RL
NEXT
->STR 3 OVER SIZE SUB ROT STOF SWAP DROP
>>
```

CHK [1142] for HP28S Rev. 2BB

```
<< RCLF STD HEX 64 STWS 48 CF SWAP RCL ->STR 16 STWS
DUP # 0h 1 ROT SIZE
FOR j
  OVER j j SUB NUM R->B XOR RL
NEXT
->STR 3 OVER SIZE 1 - SUB ROT STOF SWAP DROP
>>
```

This is the assembler to generate the System Object from the given string.  
Verify the ASS program with 'ASS' CHK before calling please.

ASS [BEB8] for HP28C Rev. 1BB

```
<< -> s
```

```
<< HEX "" 1 s SIZE
FOR i
  "#" s i DUP2 1 + DUP SUB 3 ROLLD
  DUP SUB + + STR-> B->R CHR + 2
STEP
# 3EEA0 SYSEVAL # 4D9E SYSEVAL
>>
>>
```

ASS [2EB8] for HP28C Rev. 1CC

```
<< -> s
<< HEX "" 1 s SIZE
FOR i
  "#" s i DUP2 1 + DUP SUB 3 ROLLD
  DUP SUB + + STR-> B->R CHR + 2
STEP
# 3EFC1 SYSEVAL # 4D9E SYSEVAL
>>
>>
```

ASS [BE22] for HP28S Rev. 2BB

```
<< -> lm
<< HEX "" 1 lm SIZE
FOR i
  "#" lm i DUP2 1 + DUP SUB 3 ROLLD
  DUP SUB + + STR-> B->R CHR + 2
STEP
# 3CEAAh SYSEVAL # 4F3Dh SYSEVAL
>>
>>
```

### ***2.1) Complete ROM upload***

This is ROM image upload program for uploading the complete ROM image. You need at least ~133KB (~136000 bytes!) of free RAM on the HP48 for this version. Enter the following string without any whitespace characters please. They are only for better reading. Verify the MROM28 string with 'MROM28' CHK before further use please.

MROM28 [F074] for HP28C Rev. 1BB

```
"69C20A80008F2EE4
08FA4E10D2135D1E
57C1032FF3B3951F
8F91F40142164808
CD0E47D20D47720D
4FC7F10AF234000F
714B171A627800A5
E5FEDA10BD910C8F
```

C8C1011CD511B010"

MROM28 [A030] for HP28C Rev. 1CC

```
"69C20A80008F2EE4
08F27E10D2135D1E
57C1032FF3B3951F
8F91F40142164808
CD0E47D20D47720D
4FC7F10AF234000F
714B171A627800A5
E5FEDA10BD910C8F
4BC1011CD511B010"
```

MROM28 [1BB8] for HP28S Rev. 2BB

```
"69C20080008F1805
08F26E10D2135D1E
5721032FF3B3951F
8DB2B30D0E47D20D
47720D4FC7F10AF2
34000F714B171A62
7800A5E5FEDA10BD
910C8F00D1011CD5
11B010"
```

Now we are ready to "assemble" the image upload program. Do

MROM28 ASS

please. As result you should get a "System Object" in stack level 1. Save this program with

'ROM28' STO

please.

For this way you need at least ~133KB (~136000 bytes!) of free RAM on the HP48 for upload. This can only be done by merging a RAM card to the HP48 user memory.

## ***2.2) ROM upload in parts***

This is ROM image upload program for uploading the ROM image in parts. Enter the following string without any whitespace characters please. They are only for better reading. Verify the MROM28P string with 'MROM28P' CHK before further use please.

MROM28P [35C7] for HP28C Rev. 1BB

```
"76C205A3C1307C19
```



78F376C20853405F  
0208534069C20F90  
008FA6C60068F2EE  
401318FA4E1007DA  
137E2F6F606D1E57  
B100706B3952F8F9  
1F40142164808CD0  
E47D20D47720D4FC  
7F10AF234000F714  
B171A627800A5E5F  
EDA10BD910C8FC8C  
1011CD511B0109F2  
009F20"

MROM28P [A629] for HP28C Rev. 1CC

"76C20194C1FE7C1A  
99F376C20853405F  
0208534069C20F90  
008FA6C60068F2EE  
401318F27E1007DA  
137E2F6F606D1E57  
B100706B3952F8F9  
1F40142164808CD0  
E47D20D47720D4FC  
7F10AF234000F714  
B171A627800A5E5F  
EDA10BD910C8F4BC  
1011CD511B0109F2  
009F20"

MROM28P [4C6A] for HP28S Rev. 2BB

"76C204C3C05A7C02  
1DE376C20DA3405E  
020DA34069C20490  
008FDB070068F180  
501318F26E1007DA  
137E2F6F606D1E57  
0100706E953F8DB2  
B30D0E47D20D4772  
0D4FC7F10AF23400  
0F714B171A627800  
A5E5FEDA10BD910C  
8F00D1011CD511B0  
109F2009F200"

Now we are ready to "assemble" the image upload program. Do

MROM28P ASS

please. As result you should get a "System Object" in stack level 1. Save this

program with

'ROM28P' STO

please.

Now we have to think about the size of the parts. The Clamshell ROM has a size of 128KB where the uploaded data is a little bit larger because of some protocol information.

### *2.2.1) Using a HP48G+ or a HP48GX with 128KB RAM*

The easiest way for this receiver is doing the complete process in two parts. Lets identify us the address ranges.

$128\text{KB} / 2 = 64\text{KB}$

64KB are  $64 * 1024 * 2 = 131072$  nibbles (# 20000h)

So the upload ROM areas are

# 00000h - # 1FFFFh for ROM image part 1

# 20000h - # 3FFFFh for ROM image part 2

For this way you need at least ~67KB (~68600 bytes!) of free RAM on the HP48 for each upload.

### *2.2.2) Using a HP48S(X) or a HP48G with 32KB RAM*

Here are several options for splitting. I suggest to split the process into 8 pieces of 16KB each. Lets identify us the address ranges.

$128\text{KB} / 8 = 16\text{KB}$

16KB are  $16 * 1024 * 2 = 32768$  nibbles (# 08000h)

So the upload ROM areas are

# 00000h - # 07FFFh for ROM image part 1

# 08000h - # 0FFFFh for ROM image part 2

# 10000h - # 17FFFh for ROM image part 3

# 18000h - # 1FFFFh for ROM image part 4

# 20000h - # 27FFFh for ROM image part 5

# 28000h - # 2FFFFh for ROM image part 6

# 30000h - # 37FFFh for ROM image part 7

# 38000h - # 3FFFFh for ROM image part 8

For this way you need at least ~17KB (~17400 bytes!) of free RAM on the HP48 for each upload.

### 3) HP48

Transfer the BINPRT program from this packet to the HP48. The program itself is a derivation from the original INPRT program (c) by Hewlett Packard which allows capturing of binary data. A big thank to Raymond Del Tondo for doing this.

### 4) Transfer the ROM image from the HP28C/S to the HP48

Check the calculators free memory with the MEM command. Make sure that you have enough free memory for your selected upload size on the HP48! Put the IR transmitter LED of the HP28C/S and the IR receiver LED of the HP48 together. Therefore the HP28C/S transmitter LED should point to the arrow marker on top of the HP48. To get a similar height of both LED's I fold the alphanumeric keyboard side of the HP28C/S under the LCD. I got best results with calculator distance of about 5 mm (0.2 inch). Avoid external IR light please.

#### 4.1) Complete ROM upload

Now start the BINPRT program on the HP48 and then the ROM28 (System Object) on the HP28C/S before the timeout of the BINPRT program occur.

The transmitting annunciator on the HP48 should blink now. The complete upload process takes about 37 min. now.

After the upload process you should finally have a string at stack level 2 and <1d> or <1h> at stack level 1 on your HP48. The content at stack level 1 is the error result of the BINPRT program. Drop the return value at stack level 1 and save the string at stack level 2 into a variable ('ROM' STO).

#### 4.2) ROM upload in parts

First put your HP28C/S into HEX mode by executing the HEX command. Then enter the start and end address for the address area to upload from one of the lists above. Because of the used protocol the minimum size for each block must be 256 nibbles (# 100h). So the given address range must be a multiple of this size!

Enter for example

```
#20000 <ENTER>  
#27FFF <ENTER>
```

for a 16KB part (part 5) on your HP28.

Now start the BINPRT program on the HP48 and then the ROM28P (System Object) on the HP28C/S before the timeout of the BINPRT program occur.

The transmitting annunciator on the HP48 should blink now. The upload process

time depends on the upload size. 16KB are uploaded in less than 5 min.

After the upload process you should finally have a string at stack level 2 and <1d> or <1h> at stack level 1 on your HP48. The content at stack level 1 is the error result of the BINPRT program. Drop the return value at stack level 1 and save the string at stack level 2 into a variable naming the part ('ROM5' STO for part 5).

## 5) Transferring the data from the HP48 to the PC

Now transfer the saved string (ROM or ROMx) from the HP48 to the PC in `_binary_` mode. The time for this depends on the transfer program you use. I prefer Conn4x (not for the HP48SX) for this purpose. If you are doing the ROM upload in parts repeat step 4 and 5 until you have all parts on the PC.

## 6) Getting the final ROM image

To get the final ROM image we have to strip off the HP48 binary header and protocol information (XModem) inside the uploaded file(s) and merge them together to a single file. Therefore we use the CAPC2BIN program.

For a complete ROM upload call the CAPC2BIN program with

```
CAP2BIN.EXE ROM ROMIMG.ROM
```

For a ROM upload in two parts call the CAPC2BIN program with

```
CAP2BIN.EXE ROM1 ROM2 ROMIMG.ROM
```

For a ROM upload in eight parts call the CAPC2BIN program with

```
CAP2BIN.EXE ROM1 ROM2 ROM3 ROM4 ROM5 ROM6 ROM7 ROM8 ROMIMG.ROM
```

where ROM1 ROMx ... are the source ROM parts and ROMIMG.ROM is the final ROM image name. The last displayed message in all cases should be "ROM size = 20000h = 128KB" else you got a transfer error in the steps before.

Finally you should check the internal checksums of the ROM image with the CHAMPCHK program (HP28C) or with the LEWISCRRC program (HP28S) by doing

```
CHAMPCHK.EXE ROMIMG.ROM
```

or

```
LEWISCRRC.EXE ROMIMG.ROM
```

Happy grabbing...

**Christoph Giesselink, Feb. 21th, 2007**  
**c dot giesselink at gmx dot de**

## Transferring the ROM image in smaller parts

Posted by [Christoph Giesselink](#) on 17 Jan 2008, 6:34 p.m.,

As I described in the **Clamshel.txt** file it's possible to load the ROM in pieces and finally put them together on the PC.

For the **HP48SX** four pieces with **16KB** each are best.

You can call the program four times; the only difference in program is the start address where to begin.

The ROM dump program in the HP42S normally persist the warmstart at program end when you're using my suggested base address.

So you only have to recall the program, modify the start address and upload the next 16KB block. The converting and merging the blocks is described in the Clamshel.txt file.

The following is a modified dump (sorry not tested) for uploading the HP42S ROM in 16KB blocks

Saturn Assembler Dump Lewis Pioneer ROM over IR Thu Jan 17 01:52:53 2008

V3.0.8 (12/06/2002) ROM42.a Page 1

```
* Send character in A[B] over IR
*OUTBYT EQU #0133C (HP17B rev. A)
*OUTBYT EQU #01378 (HP17B rev. B)
*OUTBYT EQU #01445 (HP17BII rev. B)
*OUTBYT EQU #0136A (HP27S rev. A)
*OUTBYT EQU #0142E (HP27S rev. B)
*OUTBYT EQU #0318D (HP42S rev. A)
OUTBYT EQU #031D6 (HP42S rev. B/C)
```

```
* no. of 128 byte blocks
PRGLEN EQU 128 16KB block
```

```
00000 1F000 D1=(5) #00000 ROM Start address
00
00007 D1 B=0 A init block no.
00009 E5 - B=B+1 A inc no. of blocks
0000B 7110 GOSUB SBLOCK send block
0000F 32F70 LC(3) (PRGLEN)-1 no. of 128 byte blocks
00014 B39 C=C-B X
00017 51F GONC -
0001A D2 C=0 A restart
0001C 81B3 PC=C
00020 D0 SBLOCK A=0 A SOH block header
00022 E4 A=A+1 A
00024 7D20 GOSUB SBYTE
```

```

00028 D4      A=B  A      block number
0002A 7720    GOSUB SBYTE  send data byte
0002E D4      A=B  A      1' complement of block number
00030 FC      A=-A-1 A
00032 7F10    GOSUB SBYTE  send data byte
00036 AF2     C=0  W      reset checksum
00039 34000   LCHEX #7F000  loop 128 bytes, chk = 0
      F7
00040 14B -   A=DAT1 B      get data
00043 171     D1=D1+ 2      next address
00046 A62     C=C+A B      add checksum
00049 7800    GOSUB SBYTE  send data byte
0004D A5E     C=C-1 M      next byte
00050 5FE     GONC -
00053 DA      A=C  A      send checksum, fall into SBYTE
00055 10B SBYTE R3=C      save byte counter/checksum
00058 D9      C=B  A      save block counter
0005A 10C     R4=C
0005D 8F6D1   GOSBVL OUTBYT
      30
00064 11C     C=R4
00067 D5      B=C  A
00069 11B     C=R3
0006C 01      RTN

```

- Block1 (16KB)

```

00000 1F000   D1=(5) #00000  ROM Start address
      00

```

- Block2 (16KB)

```

00000 1F008   D1=(5) #00800  ROM 2nd block
      00

```

- Block3 (16KB)

```

00000 1F000   D1=(5) #01000  ROM 3d block
      10

```

- Block4 (16KB)

```

00000 1F008   D1=(5) #01800  ROM 4th block
      10

```

- HP42S ROM Rev. A

```

0005D 8FD81   GOSBVL OUTBYT
      30

```

- HP42S ROM Rev. B + C

```

0005D 8F6D1   GOSBVL OUTBYT
      30

```

## Test results

*Message #9 Posted by [Mike Reed](#) on 19 Jan 2008, 4:37 p.m.,  
in response to message #8 by Christoph Giesselink*

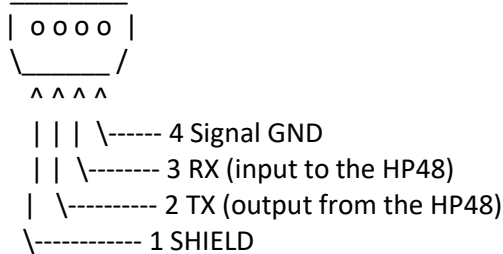
It worked perfectly - after i found and downloaded Kermit for the HP48sx - CONN4x doesn't work with the 'sx... (learned the hard way, LOL).

Thank you VERY much for the additional instructions for the ROM dump.  
I now have a fully functional emulated HP42S on my desktop, and it works perfectly!

## HP-48 Serial interface

### HP-48 serial port connector pin out

Looking at the pins of the HP48 (the following diagram is showing the pins on the HP48, as you look at the calculator):



Stated another way:

HP48	IBM 9 PIN	IBM 25 PIN
(From the outside edge -> inward)		
SHIELD-----	SHIELD-----	SHIELD
TX (Output)<-----	2 RX (Input)-----	3 RX (Input)
RX (Input)----->	3 TX (Output)----->	2 TX (Output)
SGND-----	5 GND-----	7 GND

This information is also on page 27-7 of the G/GX manual.

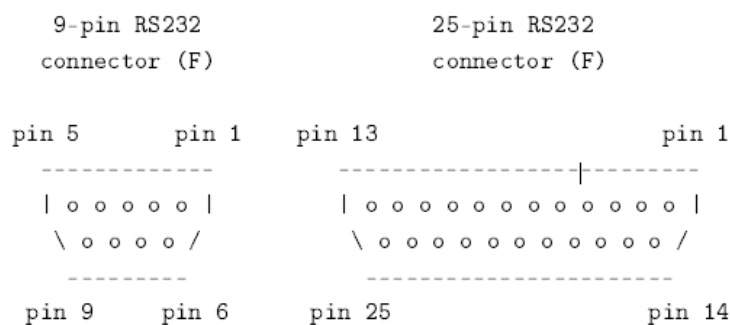
### HP-48 to PC serial cable

HP to PC cable

HP48 | RS232-9 | RS232-25

HP48	RS232-9	RS232-25
1	shield	shield
2	2	3
3	3	2
4	5	7

You can use either a 9 or a 25 pin female sub-D socket for the PC-side of the cable:





## Testing the Serial cable

Download some programs (such as those in the Best Program List) that you wish to try.

At some point you should go to <http://www.columbia.edu/kermit/> to obtain the version of Kermit that suits you best.

However, if you have an alternate communications program (for example, Windows 95 comes with HyperTerminal which you can use) you can delay downloading Kermit. However, it is a highly recommended to obtain because certain programs for the HP48 use special features in Kermit (such as server mode) not available in other communications programs.

With this done, you can begin the actual transfers.

Start your communications program, and set the port to COM1 (or whatever your link is plugged into).

In Kermit, you would type "SET PORT COM1" and in HyperTerminal you would set the dialog box with the choice of what modem you want to use to "Direct to COM1".

Then change the speeds of the ports to 9600. In Kermit, type "SET SPEED 9600" and in Hyperterminal click on Advanced.

On the HP, go to **I/O**, and go to **Transfer**. Set the calculator to **Wire, 9600 baud**, and Kermit (or **X-Modem**, if you using it instead). Then, get ready for to receive a file.

Note that X-Modem is much faster than Kermit in most situations, especially in long transfers. However, it is not available built-in on the S/SX.

On the PC start sending, and on the HP, start receiving. You should see the transfer arrow on the upper right of the screen on the HP, and it should be flashing.

On the PC, you should see a progress indicator to show how much of the file has been transferred.

When the transfer is finished, check that you received what you expected. If it what you expected, your HP PC link works!

## Transferring and Running Files

1. Connect your calculator to a serial port on your computer using a serial link cable.
2. Load your favorite communications program that supports either the Kermit or Modem protocol. If you use Windows, a suitable program is already supplied: either Terminal or HyperTerminal.
3. Set the communications program to use the serial port that you plugged your calculator into, and set it to 9600 bits per second, 8 data bits, no parity, and 1stop bit. I'm not sure what flow control should be, but Xon / Xoff works just fine for me.
4. If you have an HP48 G-series calculator, press [RightShift] [1] on your HP48 and choose **Transfer**. If you have an HP49G calculator, press [APPS] [2] [ENTER] [5] [ENTER] to select "Transfer..." on your HP49.

5. Now you have to decide whether to use **XModem** or Kermit. XModem is by far the fastest, but it requires you to type in a filename to receive and only works with binary files. Kermit is much slower, but it automatically receives the file name and can receive ASCII files.
6. After you have selected a protocol on the calculator, type in the filename (if needed) or select Binary or ASCII (if needed). In most cases, you can leave the other settings as they are. Notice that the HP48 and HP49 incorrectly say "BAUD" when they mean "BPS" or bits per second. \
7. In your communications program on your computer, choose the **Send File** command and choose the **same protocol (XModem)** as you chose on your calculator. You may then select the file that you want to send.
8. Press either **RECV** or **XRECV**, depending on the protocol, on your calculator, and then tell your computer to start sending the file.
9. The file is now stored as a **variable** on your calculator.
10. In the **VAR** menu, press the **softkey corresponding to the file** that you just transferred to your calculator.
  - If the file you transferred was a directory, it will change to the directory for that program and you should continue from there.
  - If it is a variable, the program will load right away and no further installation is needed.
  - If text that says "Library ###: ..." appears on stack level 1, proceed to the next section.