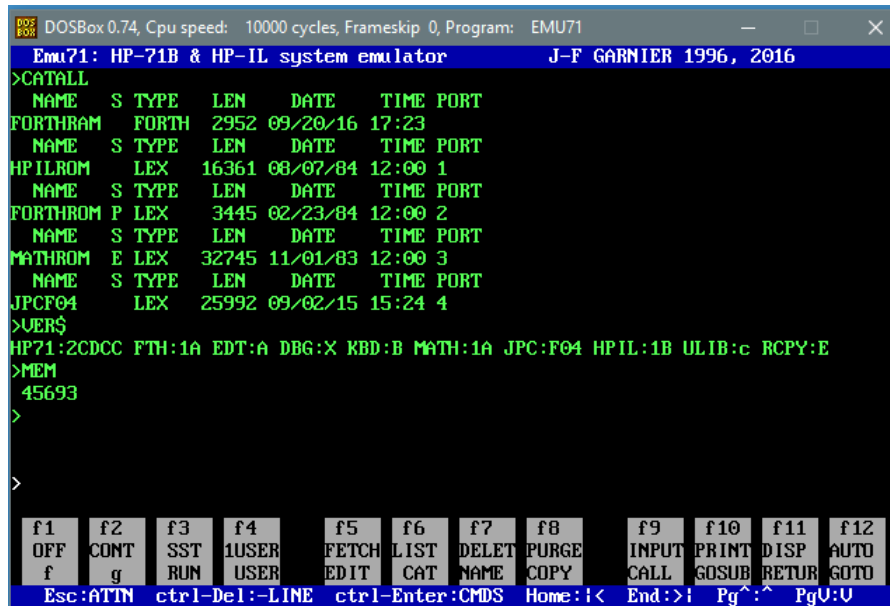


Emu71/DOS: 20th Anniversary!



Emu71 v1.1 (1996) on a Windows-98 computer



Emu71 v2.45 (2016) in DOSBox on a Windows-10 computer

Emu71/DOS history notes

■ The idea of Emu71 started in ... 1993
but work really started only on end of 1995

■ Motivations:

- build a HP-71B environment with the ability to run modules that I only owned as ROMCOPY images,
- have lot of memory (at the time my configuration was a HP-71B + a HHP 32k RAM/EPROM module – not enough!),
- and possibly be faster than the real thing.

■ First stable, usable version in **1996**: single-line display emulation, but already full ROM and RAM module emulation.

About 3x the native HP-71B speed on a 386SX at 25 MHz

```
*****  
;  
;          émulation du processeur Saturn par le 8086  
;          J-F Garnier 1993, 2004  
;-----  
; Module saturn.asm : définitions des registres, moteur d'exécution  
;-----  
; Version 0.0 16/03/93  
;   prototype, faisabilité  
; Version 1.0 04/11/95  
; Version 2.0 07/09/01 extensions pour HP42  
; Version 2.1 08/09/02 maj  
; Version 3.0 03/10/02 12 bits jump table
```



The Saturn emulation core

- Hand-coded, highly-optimized assembly code
- Heavy use of tables, macros, unrolled loops to get the highest speed. My PC of the time was a 386SX, I wasn't sure to get at least the HP-71B speed.
- Use the packed BCD format (2 nibbles per byte) and the Intel x86 native BCD support provided by the DAA/DAS opcodes, for speed (again).
- Emulation core source file (before macro expansions): ~ 200 kB
- Emulation core object code: ~ 30 kB,
- many hours (I didn't count...) of optimizing, testing, benchmarking on different machines and host CPU, ...

- bonus:
extra R5-R7 registers and
64-level hardware stack:

```
Emu71 Ready.
A=000000400000f00 B=001da2eaf004aeff C=0001da2eae000004 D=ff00000000000002
R0=0000032623015102 R1=00000007f7f3765e R2=d4d4d4d4d4d4d0e214 R3=003262301f0f0001
R4=0a02834ddc200000 R5=0000000000000000 R6=0000000000000000 R7=0000000000000000
D0=2f6e3 D1=2f761 PC=006ed P=0 CY=0 HEX ST=0010 IN=0000 OUT=00f SB
Stack: d01cf 00378 378c3 02660 378c3 02660 00365 378c3
006ED 83860 ?MP=0 / GDYES #006F6
Cmd: vs
Extended Stack:
d01cf 00378 378c3 02660 378c3 02660 00365 378c3
378c3 00365 378c3 378c3 378c3 02660 378c1 02660
37730 00365 378c1 378c1 378c1 378c1 02660 00365
378c1 378c1 378c1 00365 378c1 378c1 378c1 378bd
0291b 02660 3795b 379ed 00378 379ed 02660 37730
37a7d 37730 37a73 37a73 02f4c 37a83 37730 06af3
37a91 027e1 06af3 37afd 027e1 37afd 37730 00378
06af3 37afd 00000 00000 00000 00000 00000 00000
Cmd:
```

```
; fonctions d'addition decimal
add_byte macro
    mov al,[di]
    adc al,[si]
    daa
    mov [di],al
    inc si
    inc di
endm

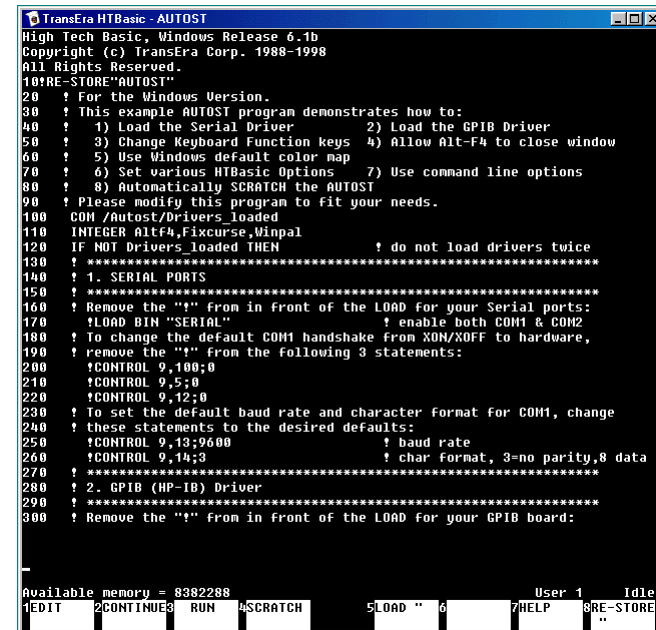
_addd_hinib:
;
mov al,[di]
mov ah,[si]
mov dl,al ; copie
and ax,0f0f0h ; masque
add al,ah
daa
lahf
and dl,0fh
or al,dl
sahf
mov [di],al
ret

_addd_16: add_byte
_addd_14: add_byte
_addd_12: add_byte
_addd_10: add_byte
_addd_8: add_byte
_addd_6: add_byte
_addd_4: add_byte
_addd_2: add_byte
update_carry
ret

_addd_15: add_byte
_addd_13: add_byte
_addd_11: add_byte
_addd_9: add_byte
_addd_7: add_byte
_addd_5: add_byte
_addd_3: add_byte
_addd_1: add_byte
_addd_lonib:
mov al,[di]
mov bh,al
adc al,[si]
daa
update_halfcarry
mov ah,bh
and ax,0f0f0h
or al,ah
mov [di],al
ret
```

Emu71/DOS specificities...

- Since speed was the top priority, some choice have been made since the beginning, especially for the I/O system emulation.
- As a result, Emu71/DOS is not an exact, accurate emulation at register level, especially for the display and keyboard management (the Diagnostic ROM fails). A benefit of this implementation is the 78-char display line instead of the 22-char LCD.
- Emu71/DOS also has some limitations on RAM modules (e.g. no support of 4 kB RAM modules, RAM ports other than port 0 not usable as main memory).
- Emu71/DOS is a text-mode, DOS-style application, and this will not change! The model of the Emu71/DOS user interface was the HP Basic aka RMB (still existing as [HT Basic for Windows](#)).
- Emu71/DOS includes a few special features:
 - support of the 8 ports (0-7) managed by the HP-71B OS,
 - support of the DEBUG command of the Forth/Assembler ROM,
 - support of the EEPROM module type.



```
TransEra HTBasic - AUTOST
High Tech Basic, Windows Release 6.1b
Copyright (c) TransEra Corp. 1988-1998
All Rights Reserved.
10:RE-STORE"AUTOST"
20 ? For the Windows Version.
30 ? This example AUTOST program demonstrates how to:
40 ? 1) Load the Serial Driver      2) Load the GPIB Driver
50 ? 3) Change keyboard Function keys 4) Allow Alt-F4 to close window
60 ? 5) Use Windows default color map
70 ? 6) Set various HTBasic Options  7) Use command line options
80 ? 8) Automatically SCRATCH the AUTOST
90 ? Please modify this program to fit your needs.
100 COM /Autost/Drivers_loaded
110 INTEGER AltF4,Fixcursr,Winpal
120 IF NOT Drivers_loaded THEN      ! do not load drivers twice
130 ! *****
140 ! 1. SERIAL PORTS
150 ! *****
160 ! Remove the "!" from in front of the LOAD for your Serial ports:
170 !LOAD BIN "SERIAL"              ! enable both COM1 & COM2
180 ! To change the default COM1 handshake from XON/XOFF to hardware,
190 ! remove the "!" from the following 3 statements:
200 !CONTROL 9,100;0
210 !CONTROL 9,5;0
220 !CONTROL 9,12;0
230 ! To set the default baud rate and character format for COM1, change
240 ! these statements to the desired defaults:
250 !CONTROL 9,13;9600              ! baud rate
260 !CONTROL 9,14;3                 ! char format, 3=no parity,8 data
270 ! *****
280 ! 2. GPIB (HP-IB) Driver
290 ! *****
300 ! Remove the "!" from in front of the LOAD for your GPIB board:

Available memory = 8382288
User 1 Idle
1EDIT 2CONTINUE3 RUN 4SCRATCH 5LOAD 6 7HELP 8BRE-STORE
```

HTBasic - a model for the Emu71/DOS user interface

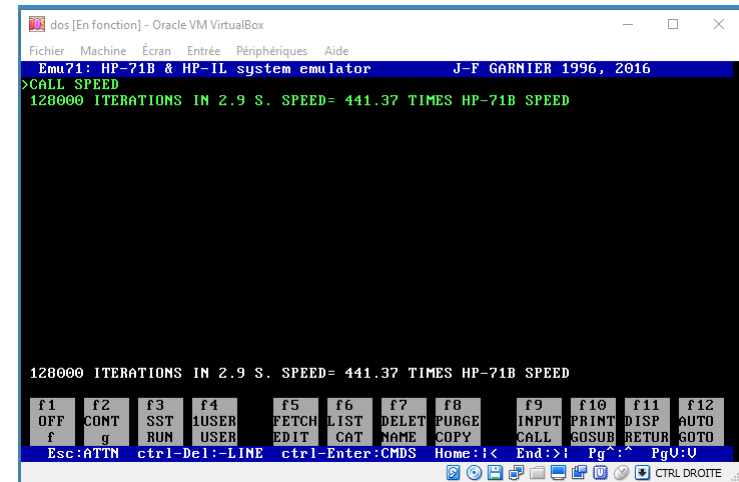
Emu71/DOS status in 2016

- Still usable:

Despite its 16-bit DOS nature, Emu71/DOS can still be run on modern 64-bit OS. [DOSBox](#) is the simplest solution. [VirtualBox](#) gives higher performance but is more complex.

- Still powerful:

x5 the HP-71B speed with DOSBox (2 emulation layers),
x400+ the HP-71B speed with VirtualBox ¹.



Emu71/DOS v2.45 in VirtualBox

- The HP-IL board and the PIL-Box are no more supported natively due to the modern OS constraints. ILPer is now the natural gateway between Emu71/DOS and the real HP-71B.

- Still maintained:

The last update was rev 2.44 on dec. 2014, to support large HP-IL drives up to 16 MB.

- And for the first time, the full Emu71/DOS source files including the Saturn emulation core in assembly language are available (under GNU license).

¹ Tested on a Intel core i3 1.8GHz computer.

Benchmark: number of empty loops done in 10s (FOR I=1 TO 1E6 @ NEXT I)

```

1 19171 6F4A          _sto_C_D0_n:      exec_mem store C D0 n
2 19172 6F4A BB 217Cr          mov  bx, offset _regD0
2 19173 6F4D BF 2116r          mov  di, offset _regC
2 19174 6F50 E9 FD1D          jmp  exec_store_n
1 19175 6F53          _lod_A_D1_n:      exec_mem load A D1 n
2 19176 6F53 BB 2180r          mov  bx, offset _regD1
2 19177 6F56 BF 2102r          mov  di, offset _regA
2 19178 6F59 E9 F6F6          jmp  exec_load_n
1 19179 6F5C          _lod_C_D1_n:      exec_mem load C D1 n
2 19180 6F5C BB 2180r          mov  bx, offset _regD1

```

The Emu71/DOS page: <http://www.jeffcalc.hp41.eu/emu71/index.html>

```

2 19181
2 19182
1 19183 6F65          _sto_A_D1_n:      exec_mem store A D1 n
2 19184 6F65 BB 2180r          mov  bx, offset _regD1
2 19185 6F68 BF 2102r          mov  di, offset _regA
2 19186 6F6B E9 FD02          jmp  exec_store_n
1 19187 6F6E          _sto_C_D1_n:      exec_mem store C D1 n
2 19188 6F6E BB 2180r          mov  bx, offset _regD1
2 19189 6F71 BF 2116r          mov  di, offset _regC
2 19190 6F74 E9 FCF9          jmp  exec_store_n
19191
19192
19193
end

```

▲ Turbo Assembler
Symbol Table

Version 4.1

23/07/07 16:19:23

Page 338

The end...