HRTmon

Alain Malek

COLLABORATORS			
	TITLE : HRTmon		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Alain Malek	April 12, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

HRTmon

1.1 HRTmon documentation

HRTmon v2.21 Documentation

look out for the latest version at:

http://dumbo.cryogen.ch/hrtmon/index.html

1.2 HRTmon/Introduction

HRTmon

Written by

Alain Malek

Have you ever wanted to get an Action Replay on your A1200, A3000 or A4000 ?

If the answer is yes, then you should have a look at HRTmon.

HRTmon is a monitor for your Amiga, which doesn't use the libraries.

You can invoke the monitor at any point. Even if a game or a demo switched

off the interrupts.

Once in HRTmon you can watch all the memory, disassemble, edit, save, etc...

(Have a look at the commands list).
and continue, as if nothing happend.
You don't have to care about any picture address. ALL the CHIP-memory can
be edited transparently.
To work perfectly HRTmon requires a 'magic' level7 button .
(This button is recommended but you can use HRTmon without it.)
HRTmon is now free, and released under the GNU General Public License (GPL)
(see: http://www.gnu.org)
visit the HRTmon web site at: http://dumbo.cryogen.ch/hrtmon/index.html

1.3 HRTmon/Requirements

REQUIREMENTS
Any Amiga with Kickstart 2.0 (or greater).
68000 - 68060 processor.
A level7 interrupt button . (Recommended)
Tested on:
- A1200
- MC68020,MC68060
- Kickstarts 39
- 2.0MB CHIPRAM, 0 - 16MB FASTRAM
Note:
- HRTmon won't work with tools like Enforcer.
To get a complete control over the system, HRTmon HAS to access the
exception vectors memory, which will generate lots of Enforcer-hits.
If you find any bugs under your configuration, please contact me
(address in the Disclaimer and Author Info section).

1.4 HRTmon/Installation

INSTALLATION
Simply copy these files in the same drawer.
hrtmon.data
HRTmonPrefs
HRTmonPrefs.info
HRTmon.guide
HRTmon.guide.info
HRTmon (this file can also be copied in c: for example)
Copy libs/reqtools.library to your libs: directory.
(If you don't have it already).

1.5 HRTmon/Usage

USAGE

HRTmon

HRTmonPrefs

Floppy disk launch

Enter HRTmon

The editor

The tracer

Commands list

Remove HRTmon

1.6 HRTmon/Usage/HRTmon

HRTmon

This program will launch HRTmon and use the preferences selected
in the HRTmonPrefs program.
It will try to load the HRTmon.data file from the current directory,
if it fails it will look into the HRTmonPrefs directory.
Usage: HRTmon -r
-r : only remove HRTmon if found in memory (no installation)

1.7 HRTmon/Usage/Tracer

Tracer

- You can enter the tracer by pressing the F7 key.
- To get out of the tracer just press F7 again
- To trace one single instruction press the right-arrow.
- To trace a BSR or a JSR instruction without tracing all the sub-routine
press the down-arrow. (The sub-routine will be executed but not traced.)

1.8 HRTmon/Usage/Editor

The Editor

General:

- When you enter HRTmon, you get directly in the editor.
- If the keyboard doesn't respond, just press your right-mouse button, to
reinit the CIA. (HRTmon doesn't do it automatically, to avoid modifying
some read-only registers.)
- The editor has 2 different pages. You can toggle between those two

pages by pressing F10
- To execute a command, you just have to type the command on any line
and press return to validate it.
- To repeat the same command without any parameter just press RETURN.
(Usefull when disassembling or dumping memory.)
- Use the arrows keys to move the cursor arround the screen and scroll up
and down into disassembled code, hex dump etc...
- The editor can work in 2 modes: insert-mode ON or OFF.
When the insert-mode is OFF, you can still insert some blank spaces, by
pressing SHIFT+Del.
- To clear a single line, press SHIFT+backspace.
- The mouse can move the 'snap cursor'. When you clic on a word
with the 'snap cursor', the word will be copied at the actual 'real cursor'
position.
History:
- If you need to enter a command you used before, just press CTRL+arrow-up
or CTRL+arrow-down to get the last commands you have executed.
Expression:
- Each time you have to enter a number in a command, you can also enter
an expression.
- You can use a register in an expression.
Examples: h a0+4
d pc
- Use:
% for binary
$ for hexadecimal
# for decimal (default)
[..] for indirect addressing
Example: h [4] will display the ExecBase structure.
- You can specify a size to each number you enter by using an extension.
.b for byte
.w for word (16 bits)
.l for longword (32 bits)
for example:
$134.l
This is usefull for functions like memory search or trainer.
- To enter a string with spaces use : '
Example: s 'mod.blue light' $40000 $50000

1.9 HRTmon/Usage/Floppy

Floppy

To launch HRTmon from a floppy disk you have to create a boot-disk

with HRTmonPrefs .

This boot-disk makes possible to debug games or demos using 'NON-DOS'

disks. (have also a look at the 'reboot' command)

You have to do the following :

- Boot the HRTmon disk.

- When the screen flashs insert the disk you want to debug.

- Press the left-mouse button to enter HRTmon.

You are now debuging the boot-block of the disk you inserted.

1.10 HRTmon/Usage/HRTmonPrefs

HRTmonPrefs

Starting HRTmonPrefs:

Type HRTmonPrefs from the CLI or simply click on the HRTmonPrefs icon.

HRTmonPrefs options:

Address

Enter the address where you want HRTmon to be located.

This address must be entered in hexadecimal.

If you don't have any idea where to locate HRTmon then type in this

address: $1d0000

Repeat key delay

Change the repeat key delay of HRTmon. (Default = 15)

AllocABS

When HRTmon is installed the memory used will automatically be

allocated through the AllocABS function of exec.library.

This behaviour can be turned off by this switch.

AutoAlloc

If you don't want to specify yourself the address field just

activate this option. HRTmon will try to find out by itself the

best address. (This option works at best with Kickstart 3.0 or higher,

but you can also use it with Kickstart lower than 3.0)

Insert mode as default

Editor will use 'insert' mode as default. (You can still toggle

the mode with the F2 key)

Right-Mouse Enter

Select this option if you want to enter HRTmon by pressing on
your right mouse button. If you don't select this option or the
the Key Enter option, then the only way to enter HRTmon will be
the level7 button .
Key Enter
Select this option if you want to enter HRTmon by pressing on
the backslash key (on US keyboard). This key is located next to
the backspace key just over the return.
IDE hardisk
Select this option if you have an A1200 or A4000 with an IDE
harddisk and you want to access this disk from HRTmon.
Have also a look at the HD WARNINGS .
LoadView Task
If selected a special task will be added to restore special
screens correctly (gfx boards, or non 15khz display)
The LoadView call can't be done from an interrupt routine
(here HRTmon), that's why it uses an additional system task.
Keyboard
Select the keyboard you want to use in HRTmon.
No VBR move
Do not relocate the vbr if it was previously set to zero.
(if VBR isn't zero, HRTmon never relocates the VBR)
Make Boot-Disk
Click this button to create a disk which will load HRTmon at
'address', and allows you to boot from another disk.
(See Floppy disk launch)
Install
After having checked the Address, and AllocABS options click on
this button to install HRTmon in memory.
Saving HRTmonPrefs options:
Select 'Save' in the preferences menu.

1.11 HRTmon/HD warnings

Hard Disk Warnings !
When accessing a partition from HRTmon you have to follow the following
rule:
- After having saved a file to your harddisk using HRTmon. You should
perform a RESET of your computer before accessing this partition again

from the AmigaDOS or Workbench side. But you can perform as many access
you want from HRTmon.
If you don't follow this rule you might corrupt the files saved from
HRTmon and you will need to perform a repair of your partition using
Disksalv of QuarterBackTools or any similar program, but you won't loose
any file already on your HD !
Instead of reseting your computer you can also perform a 'diskchange' of
the partition. (This is an AmigaDOS command)
example:
Get into a Shell or the CLI and type:
DiskChange hd0:
So the AmigaDOS will see the modifications of the disk made by HRTmon.

1.12 HRTmon/Usage/Enter

ENTER
To enter HRTmon, simply press your level7 button or
the right-mouse button if you have selected the Right-Mouse Enter in
HRTmonPrefs or the BackSlash key also depending
of your settings in HRTmonPrefs .
If the keyboard doesn't work in HRTmon press the right
mouse button. (It shouldn't happen very often.)
HINTS:
HRTmon uses the VBR register to get transparent.
If you can't enter a demo or game, than try to trace the startup-code
and modify every instruction concerning the VBR instruction.

1.13 HRTmon/Usage/Commands

COMMANDS
List of commands :
r view or edit CPU registers
a assemble 680x0
d disassemble/assemble 680x0
h hex dump/modify memory
n ascII dump of memory
e dump/modify custom registers
type type from memory
c copy memory

q compare memory

o fill memory

ce exchange memory

f find

fi find instruction

fs find string

cop find copperlist

p enter the graphic searcher

b set or remove breakpoint (view all)

bj set or remove JSR breakpoint

bd remove all breakpoints

mw set/remove/view memory watch

mwd delete all memory watch

t trace

ta trace till address reached

debug switch debug mode on/off

drive change current drive

motor switch drive motor on

format format FFS disk

qformat quick format FFS disk

diskchk check a floppy disk

ide get information from IDE drives

part list all partitions

dir dir

cd change current directory

l load file

s save file

sp save the actual picture

makedir makedir

del delete a file

copy copy a file

rs read sectors from disk

ws write sectors to disk

bb calculate boot-block checksum

la load all

sa save all

sac save all compressed

d2f read a floppy and save it as a file

f2d read a file and save it as a floppy

ts trainer start

tsd deep trainer start

tf trainer find

output redirect output to memory

af assemble 65816

df disassemble 65816

fif find instruction 65816

clear clear all memory and reboot

reboot reboot and keep HRTmon resident

pal set PAL display

ntsc set NTSC display

31k set 31Khz display (Productivity,...)

led switch power led on/off

? evaluate expression

excep show exception vectors

x exit HRTmon

kill kill HRTmon

List of special keys :

F1 clear screen

F2 toggle insert mode on/off

F6 toggle 65802/65816 CPU mode (for disassemble)

F7 enter and exit tracer

F10 toggle betwen page 1 and 2

ESC break the actual command

HELP show commands list

See also : The Editor

1.14 HRTmon/Usage/Commands/CMD_31K

CMD_31K : Set 31khz display

Syntax:

31K

Notes:

- If you have stopped a program which runs on a Productivity, Euro72,

DblPal, DblNtsc screen you can restore the display correctly by

using this command.

1.15 HRTmon/Usage/Commands/CMD_FS

CMD_FS : Find String

Syntax:

fs start end 'string'

Parameters:

start = start address of search

end = end address of search

string = string to search

Notes:

- The search isn't case sensitive.

1.16 HRTmon/Usage/Commands/CMD_E

CMD_E : dump/edit custom registers

Syntax:

e offset [newval]

Parameters:

offset = offset of custom register ($100 <=> $dff100)

newval = new value for the custom register

Notes:

- This command replaces the old cmd_esc (for HRTmon 1.x users)

1.17 HRTmon/Usage/Commands/CMD_DISKCHK

CMD_DISKCHK : check disk

Description:

- Check a floppy disk for errors

Syntax:

diskchk

Notes:

- Use the DRIVE command to check a disk in df1:

1.18 HRTmon/Usage/Commands/CMD_R

CMD_R : registers

Description:

- View the processor registers, or modify a register.

Syntax:

r

or r reg val

Parameters:

reg = name of register to modify.

val = new value for the register.

Notes:

- register RTS shows you the return address if a RTS is executed.

if the PC points on a RTE instruction, you will see the 'RTE address'.

1.19 HRTmon/Usage/Commands/CMD_A

CMD_A : assemble

Description:

- Assemble MC680X0 code into memory.

Syntax:

a address instruction

Parameters:

address = address where the instruction must be assembled.

instruction = MC68000-MC68040/FPU/MMU instruction.

Notes:

- You have to enter the instruction with the new syntax.

Example: move.l d0,(4,a0) correct

move.l d0,4(a0) incorrect

move.l d0,($40000) correct

move.l d0,$40000 incorrect

- You have to enter the instructions in their original forme.

Example: cmpi.w #4,d0 correct

cmp.w #4,d0 incorrect

- If the assemble command succeeds it will automatically print 'a nextaddress'

on the next line.

If you want to stop assembling just press return.

Otherwise type in the next instruction.

- You can also use the D command.

1.20 HRTmon/Usage/Commands/CMD_D

CMD_D : disassemble / assemble

Description:

- Disassemble (or assemble) MC680X0 code from memory.

Syntax:

d address [instr]

Parameters:

address = address of the 680x0 code to disassemble.

Notes:

- 8 instructions from 'address' will be disassembled and printed.

1.21 HRTmon/Usage/Commands/CMD_H

CMD_H : hex dump

Description:

- Dump memory in hex and ascII, or edit memory.

Syntax:

h address ;hex dump

or h address val.x val.x val.x ... ;modify memory

or h address 'string' ;modify memory

Parameters:

address = address of the memory to dump or to modify

val = new values to write in memory.

x = size of val = b,w,l (default: w)

Examples:

h $40000 10.b $140.w $12345678.l

h $50000 'HRTmon'

1.22 HRTmon/Usage/Commands/CMD_N

CMD_N : ascII dump

Description:

- Dump memory in ascII only.

Syntax:

n address

Parameters:

address = address of the memory to dump.

1.23 HRTmon/Usage/Commands/CMD_TYPE

CMD_TYPE

Description:

- Type a text in memory in the editor.

Syntax:

type address

Parameters:

address = address of the memory to type.

Notes:

- the memory will be printed in the editor.

A zero will stop the command.

1.24 HRTmon/Usage/Commands/CMD_C

CMD_C

Description:

- Copy a block of memory.

Syntax:

c start end dest

Parameters:

start = start address of the block to copy.

end = end address of the block to copy.

dest = destination address for the block.

1.25 HRTmon/Usage/Commands/CMD_Q

CMD_Q

Description:

- Compare two blocks of memory to see if they are equals.

Syntax:

q start end start2

Parameters:

start = start address of the block to compare.

end = end address of the block to compare.

start2 = start address of the 2nd block to compare with the first one.

Notes:

- If the two blocks are different, the address of the first byte different

from the 2nd block will be printed.

- If the two areas are equal, the following message will be printed:

'Equal areas.'

1.26 HRTmon/Usage/Commands/CMD_O

CMD_O
Description:
- Fill the memory with the same byte.
Syntax:
o start end val
Parameters:
start = start address of the block to fill.
end = end address of the block to fill.
val = value used to fill the block.
Notes:
- 'val' can only be a byte. (If you enter a word or a long it will be
truncated to a byte.)

1.27 HRTmon/Usage/Commands/CMD_CE

CMD_CE
Description:
- Exchange two blocks of memory.
Syntax:
ce start end dest
Parameters:
start = start address of the block to exchange.
end = end address of the block to exchange.
dest = destination address for the exchange.
Notes:
- The two areas will be exchanged.

1.28 HRTmon/Usage/Commands/CMD_F

CMD_F
Description:
- Find a sequence of bytes,words,longs in memory.
Syntax:
f start end val.x val.x ...
Parameters:
start = start address for the search.
end = end address for the search.
val = value to find.
x = size (b,w,l)
Examples:
f $f80000 $f90000 $4ef9.w $f800d2.l

1.29 HRTmon/Usage/Commands/CMD_FI

CMD_FI

Description:

- Search for a string in the disassembled (MC680X0) code.

Syntax:

fi start end 'string'

Parameters:

start = start address for the search.

end = end address for the search.

string = string to find.

Notes:

- don't forget to use the new Motorola syntax.

- you can use the * character to replace any character.

- you can also find relative address as the fa command of the

ActionReplay

Examples:

fi $f80000 $f90000 '9a,a*)'

will find for example:

MOVE.W #$7FFF,($9A,A4)

1.30 HRTmon/Usage/Commands/CMD_P

CMD_P

Description:

- Enter the graphic searcher

Syntax:

p picno

Parameters:

picno (optional parameter)

Notes:

- First use the COP command to find

the right copper-list, then use this command

- F10 or ESC : exit this mode.

- arrows keys : to move unlocked plans

- Shift : Fast move

- 1-8 : Lock, Unlock plans

- M : Increase modulo

- N : Decrease modulo

- , : Clear modulo
- Q : Decrease picture width
- W : Increase picture width
- A : Decrease picture height
- S : Increase picture height
- R : Set all bitplans to the same address
- H : Toggle HAM mode on/off
- Help : Switch info panel on/off
- Del : Change colors of info panel
- F1 : Start / stop set height mode

1.31 HRTmon/Usage/Commands/CMD_B

CMD_B

Description:

- Set or remove an Illegal breakpoint.

Syntax:

b address ;set,remove breakpoint

or b ;view all breakpoints

Parameters:

address = address of the breakpoint to set or remove.

1.32 HRTmon/Usage/Commands/CMD_BJ

CMD_BJ

Description:

- Set or remove a JSR breakpoint.

Syntax:

bj address ;set,remove JSR breakpoint

or bj ;view all breakpoints

Parameters:

address = address of the breakpoint to set or remove.

Notes:

- Will enter the monitor with a JSR instruction.

Don’t forget that it will modify 6 bytes in memory to put a JSR.

- This feature was implemented for Amigas without a VBR register. (68000)

It doesn’t need to modify the illegal instruction exception vector.

Use b command instead.

1.33 HRTmon/Usage/Commands/CMD_BD

CMD_BD

Description:

- Remove all breakpoints set by {"b " Link cmd_b} or {"bj " Link cmd_bj}.

Syntax:

bd

Notes:

- All breakpoints will be removed.

1.34 HRTmon/Usage/Commands/CMD_MW

CMD_MW

Description:

- set/remove/view memory watch

Will switch in trace mode until the watched memory is modified.

Syntax:

mw (view all memory watch)

mw address (set a ’byte’ memory watch)

mw.b address (set a ’byte’ memory watch)

mw.w address (set a ’word’ memory watch)

mw.l address (set a ’long’ memory watch)

mw address (remove a single memory watch)

Notes:

- Programs will slowdown a lot.

- It won’t trace interrupt routines.

1.35 HRTmon/Usage/Commands/CMD_MWD

CMD_MWD

Description:

- Remove all memory watch.

Syntax:

mwd

1.36 HRTmon/Usage/Commands/CMD_T

CMD_T
Description:
- Trace one or more instructions.
Syntax:
t ;trace 1 instruction
or t nbsteps ;trace nbsteps instructions
Parameters:
nbsteps = nb of instructions to trace without entering in the monitor.

1.37 HRTmon/Usage/Commands/CMD_TA

CMD_TA
Description:
- Trace till a specific address is reached.
Syntax:
ta address ;trace 1 instruction
Parameters:
address = when the processor reaches this address, it will activate the
monitor.
Notes:
- This function is usefull when you want to put a breakpoint in a ROM.
- Warning ! This function will slow-down your computer, as each instruction
will be traced until the address is reached.
- Be carfull when the processor switchs from User mode, to Supervisor !

1.38 HRTmon/Usage/Commands/CMD_DEBUG

CMD_DEBUG
Description:
- Intercept some exceptions.
Syntax:
debug ;switch on/off debug mode
Notes:
- When in debug mode. The monitor will be invoked when one of the following
exceptions is raised.
Bus Error
Address Error
Illegal Instruction
Divide by Zero
LINE-A
LINE-F

1.39 HRTmon/Usage/Commands/CMD_DRIVE

CMD_DRIVE

Description:

- View or set the actual drive number.

Syntax:

drive ;view actual drive no

or drive no ;change drive no

Parameters:

no = no of the drive to use for disk operations. (0-3)

Notes:

- All disk operations will take place on that drive.

1.40 HRTmon/Usage/Commands/CMD_MOTOR

CMD_MOTOR

Description:

- Switch the drive motor on.

Syntax:

motor

1.41 HRTmon/Usage/Commands/CMD_FORMAT

CMD_FORMAT

Description:

- Format a floppy disk in FFS.

Syntax:

format diskname

Parameters:

diskname = name for the disk. Default = HRTmon

Notes:

- The disk will be formatted in FFS.

- Use the DRIVE command to format a disk

in df1:

1.42 HRTmon/Usage/Commands/CMD_QFORMAT

CMD_QFORMAT
Description:
- Quick format a floppy disk.
Syntax:
qformat diskname
Parameters:
diskname = name for the disk. Default = HRTmon
Notes:
- The disk will be quick formatted in FFS.
- This command won't work properly if the disk as never been formatted
in any Amiga format. (First use format)
- Use the DRIVE command to format a disk
in df1:

1.43 HRTmon/Usage/Commands/CMD_DIR

CMD_DIR
Description:
- View the contents of the actual directory.
Syntax:
dir
or dir name
Parameters:
name = name of the directory to list.
Notes:
- use cd command to change the current directory.

1.44 HRTmon/Usage/Commands/CMD_CD

CMD_CD
Description:
- Change current directory.
Syntax:
cd path
Parameters:
path = path of the future current directory
Examples:
cd hd0:c
cd df0:
Notes:
- To go to the parent directory type : cd /

1.45 HRTmon/Usage/Commands/CMD_L

CMD_L

Description:

- Load a binary file into memory.

Syntax:

l name address

Parameters:

name = name of the file to load from disk.

address = destination address for the file.

1.46 HRTmon/Usage/Commands/CMD_S

CMD_S

Description:

- Save a block of memory to disk.

Syntax:

s name start end

Parameters:

name = name of the file to save to disk.

start = start address of the block to save.

end = end address of the block to save.

Have also a look at the HD WARNINGS .

1.47 HRTmon/Usage/Commands/CMD_MAKEDIR

CMD_MAKEDIR

Description:

- Create a new directory.

Syntax:

makedir name

Parameters:

name = name of the new directory to create.

1.48 HRTmon/Usage/Commands/CMD_RS

CMD_RS

Description:

- Read some sectors from disk.

Syntax:

rs address startsec nbsec

Parameters:

address = address where you want to load the sectors.

startsec = first sector you want to load.

nbsec = nb of sectors from startsec you want to load.

1.49 HRTmon/Usage/Commands/CMD_WS

CMD_WS

Description:

- Write some sectors to disk.

Syntax:

ws address startsec nbsec

Parameters:

address = address from where you want to save.

startsec = first sector of the disk to save.

nbsec = nb of sectors from startsec you want to save.

1.50 HRTmon/Usage/Commands/CMD_LA

CMD_LA

Description:

- Load all. (See sa)

Syntax:

LA

Notes:

- You will be asked to enter the disks you used for sa .

- As all registers can't be read. This function won't work 100%.

- To get the best results do the following.

1st load the programm you want to restore.

2nd when you are in the programm use the LA command.

By this way all custom registers will be already initialized properly.

1.51 HRTmon/Usage/Commands/CMD_SA

CMD_SA

Description:

- Save all. (Save memory, registers, etc... to disks.)

It is usefull if you want to save a game which doesn't have a 'save game'
option.
Syntax:
SA
Notes:
- You will be asked to enter multiple disks if not saving onto a HD.
- As all registers can't be read. This function won't work 100%.
- See la command to see how to restore the programm.
- Warning ! Only the CHIP memory from $0 to $200000 will be saved.
So, if you want to use this command you will have to switch off your
FAST-RAM.
- Have also a look at the HD WARNINGS .

1.52 HRTmon/Usage/Commands/CMD_SAC

CMD_SAC
Description:
- Crunch and save all. (Save memory, registers, etc... to disks.)
It is usefull if you want to save a game which doesn't have a 'save game'
option.
Syntax:
SAC
Notes:
- You will be asked to enter multiple disks if not saving onto a HD.
- As all registers can't be read. This function won't work 100%.
- See la command to see how to restore the programm.
- Warning ! Only the CHIP memory from $0 to $200000 will be saved.
So, if you want to use this command you will have to switch off your
FAST-RAM.
- Have also a look at the HD WARNINGS .

1.53 HRTmon/Usage/Commands/CMD_TS

CMD_TS
Description:
- Start the trainer process.
It will save all address containg a specific value (the actual number of
lives you have) in a buffer.
Syntax:

ts start end val.x

Parameters:

start = start of the memory block you want to search.

end = end of the memory block.

val = the value you want to search. (number of lives)

x = b or w or l

Notes:

- After having used this command, continue your game and loose one life.

Then enter HRTmon and use the tf command.

1.54 HRTmon/Usage/Commands/CMD_TS

CMD_TS

Description:

- Start the trainer process in deep mode

It will save all address containg a specific value (the actual number of

lives you have +- 1) in a buffer.

Syntax:

tsd start end val.x

Parameters:

start = start of the memory block you want to search.

end = end of the memory block.

val = the value you want to search. (number of lives)

x = b or w or l

Notes:

- After having used this command, continue your game and loose one life.

Then enter HRTmon and use the tf command.

1.55 HRTmon/Usage/Commands/CMD_TF

CMD_TF

Description:

- Contine the trainer process.

It will check if the address contained in the trainer buffer

have the value given in parameter.

If it's the case the address will be printed.

Only the printed address will remain in the trainer buffer, so you

can repeat the same procedure, to track step by step the exact address.

Syntax:

tf start end val

Parameters:

start = start of the memory block you want to search.

end = end of the memory block.

val = the value you want to search. (new number of lives)

Notes:

- If there are too many address, continue your game and loose one life.

Then enter HRTmon and use the tf command again.

1.56 HRTmon/Usage/Commands/CMD_OUTPUT

CMD_OUTPUT

Description:

- Makes a copy of all outputs of HRTmon in memory.

Syntax:

output address ;start output

or output ;stop output

Parameters:

address = address where you want to redirect the output of HRTmon.

1.57 HRTmon/Usage/Commands/CMD_AF

CMD_AF

Description:

- assemble 65816 code in memory.

Syntax:

af address string

Parameters:

address = address where the code must be assembled.

string = code to assemble.

Parameters:

- I don’t remember if this function was 100% finished.

As I never use it, I don’t care... (Ask me if you really need it)

1.58 HRTmon/Usage/Commands/CMD_DF

CMD_DF

Description:

- disassemble 65816 code from memory.

Syntax:

df address

Parameters:

address = address to disassemble

Notes:

- Use the F6 key to toggle the 8/16 bit mode.

1.59 HRTmon/Usage/Commands/CMD_FIF

CMD_FIF

Description:

- Search for a string in the disassembled (65816) code.

Syntax:

fif start end 'string'

Parameters:

start = start address for the search.

end = end address for the search.

string = string to find.

Notes:

- you can use the * character to replace any character.

1.60 HRTmon/Usage/Commands/CMD_COP

CMD_COP

Description:

- Find the actual copper-list.

Syntax:

cop start end

Parameters:

start = start of the range to search.

end = end of the range to search.

Notes:

- This command can take some time if the copper-list is very long...

1.61 HRTmon/Usage/Commands/CMD_?

CMD_?

Description:

- Evaluate an expression.

Syntax:

? expression

Parameters:

expression = expression to evaluate.

Notes:

- use the '$' character to enter an hexadecimal value.

- use the '%' character to enter a binary value.

- use the '#' character to enter a decimal value.

- you can use any register in the expression.

Example:

? a0+d0*4

- use the '[' and ']' characters to enter an indirect address.

1.62 HRTmon/Usage/Commands/CMD_X

CMD_X

Description:

- Exit HRTmon

Syntax:

x

1.63 HRTmon/Usage/Commands/CMD_KILL

CMD_KILL

Description:

- Kill and exit HRTmon

Syntax:

kill

1.64 HRTmon/Usage/Commands/CMD_IDE

CMD_IDE

Description:

- Display IDE HardDisk information

Syntax:

ide

Notes:

- You need an A1200 or an A4000 to access a HardDisk.

- You need to activate the IDE option in HRTmonPrefs .

- Read HD WARNINGS before accessing your HardDisk.

1.65 HRTmon/Usage/Commands/CMD_PART

CMD_PART

Description:

- Display a list of accessible disks and partitions.

Syntax:

part

Notes:

- You need an A1200 or an A4000 to access a HardDisk.

- You need to activate the IDE option in HRTmonPrefs .

- Read HD WARNINGS before accessing your HardDisk.

1.66 HRTmon/Usage/Commands/CMD_SP

CMD_SP

Description:

- Save the actual picture of the graphic searcher (P command)

in IFF format.

Syntax:

sp filename

1.67 HRTmon/Usage/Commands/

This function doesn't need any comments !

1.68 HRTmon/Usage/Remove

REMOVE

- To remove HRTmon just run HRTmonPrefs .

HRTmon will be automatically located and removed.

- You can also use the KILL command.

1.69 HRTmon/Level7

LEVEL7 BUTTON

HRTmon needs a special hardware to work properly.

(If you don't have it you can still use HRTmon with the 'right-mouse enter'

or the 'key-enter' options of HRTmonPrefs)

You will have to add a level7 button to your Amiga (if you don't have it

already). This button will also work with other software as ASMOne.

This button will generate a level7 (Non maskable interrupt) on you Amiga.
When you press this button, you will immediatly enter HRTmon.
Don't be afraid if you are not an electronics specialist ! It's very easy
if you know soldering.
(If you can't do it, just ask some friends. I'm sure you know someone
who will be able to do it.)
Material needed :
- 1 button.
- some wires.
- 3 diodes 1N4148 or any other equivalent.
Here is the schematics of this button:
IPL0 -------- >| ------\
|
IPL1 -------- >| ---------- button ------- GND
|
IPL2 -------- >| ------/
>| = diode.
IPL0 IPL1 IPL2 signals can be found on pin 83-82-81 of the expansion
port of the A1200.

1.70 HRTmon/Disclaimer and Author Info

HRTmon Information
HRTmon Amiga system monitor

Please send your comments, wishes and bug reports for HRTmon to:
Alain Malek
Bonne-Esperance 1
1006 Lausanne
SWITZERLAND
Or by email:
Alain.Malek@cryogen.com
Please remember to state the version number you are using, in all
correspondance.

1.71 HRTmon/Thanks to...

Thanks to ...
- Nico Francois for his great reqtools library.
- Carnivore/BeerMacht for the assembler and disassembler functions.
First I did my own routines, but only for 68000, and I was too lazy to
add the 68020-68040,FPU,MMU instructions. So I took his routines from
BeerMon and adapted them to HRTmon.
- Roberto Marra, sCREEM/cRUX, SHANE, Jean-Francois Fabre, Henry Sopko
for beta-testing and new ideas.
- Pablo d'Angelo for his help in finding the cpu detection bug.
- And all the people who registred HRTmon !

1.72 HRTmon/Future

The Future
- Add play-sound command (no one seems to need it, till now...)
- Add breakpoint option in the tracer.
- Add saving of FASTRAM to SA/LA commands.
- Address access breakpoint by using the MMU
- Trap write-only registers with MMU
- PPC assembling/disassembling
- Any good idea from you :)

1.73 HRTmon/FAQ

FAQ (Frequently Asked Questions)
1 - I can't find a FA command ?
Use the FI command instead. Finding relative
addresses is just one aspect of this powerfull command.
2 - HRTmon doesn't restore the display of my game correctly ?
I can't do anything against this. The problem is due to some
write-only registers which can't be restored.
To get the correct display mode you can try to change the default
values of some custom registers as BPLCON0 ($100),
BPLMOD0 ($108), FMODE ($1fc) by using the E command.
3 - HRTmon doesn't restore my Productivity, Euro72, DblPal, DblNtsc, ...
(or gfx board) screen ?
Again this is due to some write-only registers which can't be
restored.
You can solve this problem by using the 31K command,
or use the LoadView option of HRTmonPrefs.

1.74 HRTmon/History

History

version 2.21 Jul 18 1998

- memory watch command added

- level2 interrupt not used anymore, to avoid freeze with extension cards

(pcmcia, ethernet, scsi, etc...)

- option -r added to the HRTmon command (for removing HRTmon)

- HRTmonPrefs fixed and Kickstart 2.0 compatible

- no more crash when unknown command entered

version 2.20 Jul 16 1998

- German keymap fix (Bert Jahn)

- copper search fix

- vbr relocation changed

- support for 68000 and 68010

- HRTmon command doesn't use reqtools anymore

- config file format changed

- new 'no vbr move' option in HRTmonPrefs

- command crash detection and recover system added

version 2.19 Jun 29 1998

- some WHDLoad adaptation (Bert Jahn)

- better excep command (Bert Jahn)

- display HRTmon entry reason (Bert Jahn)

- file Lock fix in HRTmon command (Bert Jahn)

- gfx-ripper info pannel fix

- copper search routine fix

version 2.18 Jun 23 1998

- hd partition detection fixed

- floppy disk access fixed when in multiscan mode

- mmu related problem fixed

version 2.17 Jun 12 1998

- source code clean-up

- PhxAss support (v4.39alpha or better)

- BASM support (Bert Jahn)

- disable PCMCIA interrupts when in HRTmon. (avoid freeze)

version 2.16 Jun 4 1998 (9th public release)

- first source code release under GPL

version 2.15 25 Apr 1998

- EHB (halfbright) support in gfx-ripper

version 2.14 9 Jan 1998 (8th public release)

- support for CD32 ProModule IDE interface
- faster copper list search (cmd COP)
- support for NTSC and Multiscan screen modes
- configuration fileformat changed, users of v2.13 or prior should save a new config from HRTmonPrefs

version 2.13 6 Nov 1997 (7th public release)

- level 7 a little bit more powerfull
- better IDE HD support
- trainer functions fixed

version 2.12 16 Jun 1997 (6th public release)

- 68060 instructions added
- 68060 register display

version 2.1 22 May 1997 (5th public release)

- 'Snap cursor' added in editor window
- new status bar
- no output on screen when switching insert mode on/off
- '#' key was missing on French keyboard
- command history invoked through CTRL-up/down keys
- use SHIFT-up/down to scroll faster in memory dump
- Garbage on screen with some 040 machines fixed
- help page doesn't scroll further when HELP key released
- 31K command added
- Bug in disk bitmap allocation removed. (got a disk full msg too early)
- Bug in CPU detection routine removed. (failed on 68020 boards with FPU)
- LoadView task option added in HRTmonPrefs (gfx-board compatible)

version 2.00 24 Nov 1996 (4th public release)

- IDE hard-disk access (A1200 and A4000 only)
- Support for International FileSystem
- Works on 68020-68060
- A4000 bug fixed
- Scroll up and down for disassembly, mem dump, etc...
- New powerfull gfx searcher and ripper
- Save picture command
- Improved floppy disk access
- HRTmon can be entered by pressing a key
- Auto memory location
- Some more preferences
- Completly rewritten file access routines

- New HRTmon program launcher
- Lots of small usefull commands added...

version 1.20 2 May 1995 (3rd public release)

- SETMAP command removed.
- Now you select your keyboard from HRTmonPrefs
- Added MMU option in HRTmonPrefs
- Added Clock option in HRTmonPrefs

(The auto-detection of the clock caused a crash on A3000 and A4000/030 and maybe some other configurations too.)

version 1.10 22 April 1995

- Added KILL command.

version 1.01 10 April 1995 (2nd public release)

- Removed a bug when entering HRTmon. (Sometimes it crashed...)
- No more crash when exiting HRTmon, after having pressed the right-mouse button in HRTmon (only in 'right-mouse enter' mode.)
- Now the debug command should work correctly.

version 1.00 2 April 1995

- First public release