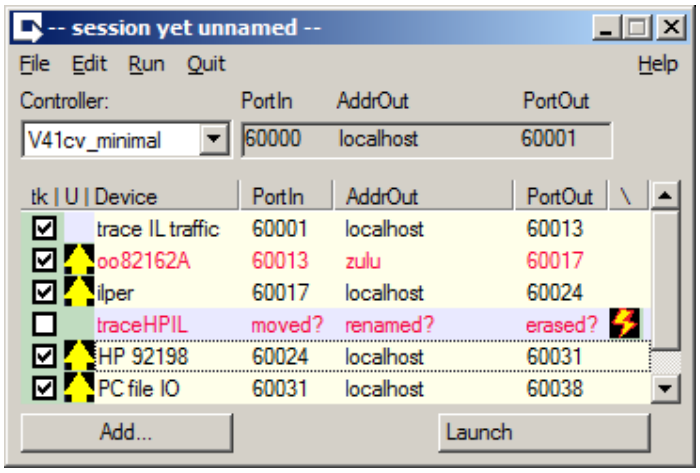
"And virtual is real."

George Clinton, 'Paradigm' (2001)
co-produced with Prince

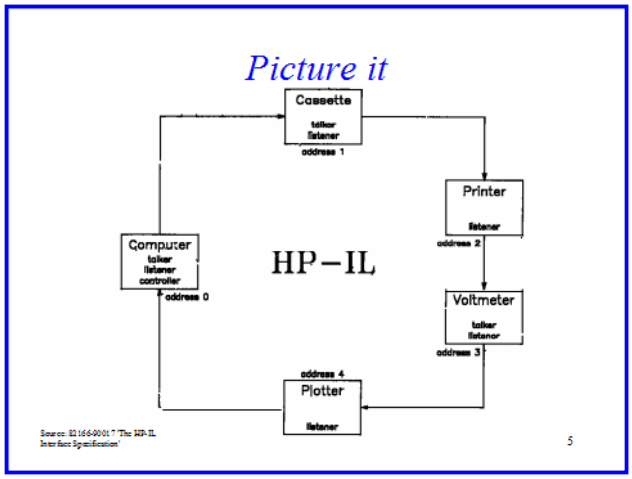
V*il*MA – Virtual-IL Manager
a REXX program for Windows

You do know HP-IL. You probably know also Virtual HP-IL. Have a look at this set-up depiction. You got it but you are still interested? — Then **V*il*MA** may be beneficial for you as for all virtual devices using VIL set-up files it self-acting assigns matching port numbers. In addition it may then start devices with the new port numbers and the controller (HP-41C/CV/CX under V41 or an HP-71B under Emu71).

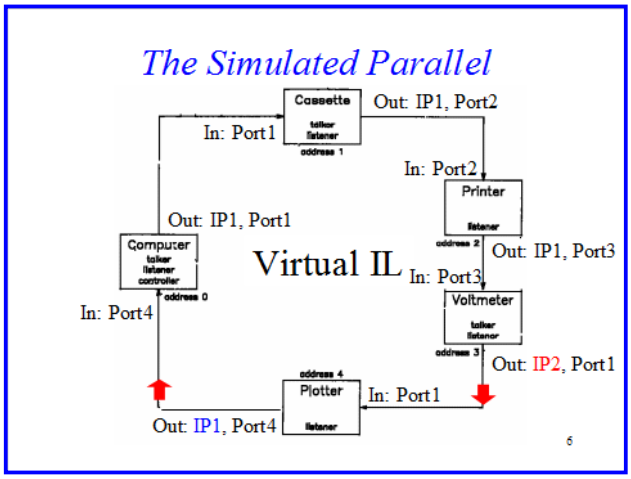


Purpose

To connect *real* HP-IL devices you use *real* cables. To connect *virtual* IL devices you need *matching IP port numbers*.



Every number you have to enter twice, first as a *PortOut* number in the HP-IL set-up of an antecedent device and again in the set-up of the following device as *PortIn* number. If possible free of typos, otherwise it will not work. For sure not. (Proven, many times.)



V*il*MA will relieve you from this awful drudgery. Just put the set-up files of the devices you like to link in a list, set them in your preferred sequence, choose a pre-defined controller and that's it. **V*il*MA** reads in- and out-port from the controller's state file, use these as start and end of the virtual loop and computes for all devices in between suitable port numbers. Those are written to the config file of each device (using VIL as suffix) before starting them and also the controller. Simple, efficient, convenient. *Awesome*. At least worth this memo.

Prerequisites

To run **V*il*MA** you need (in brackets what I used for the development):
Don't worry, even this list looks large, most of it you probably have already, otherwise you would not be interested in **V*il*MA**.

- Windows (W7 Home Premium SP1),
- Open Object REXX (V4.2.0), see oorexx of "Open Object Rexx",
- ooDialog (V4.2.3), see ooDialog of "Open Object Rexx",
- OS/2 Pipelines (version 1.00.56, a *copy* of the VM **PIPE** command — unlikely you find it outside IBM, just to make you envious see the manual of the role model),
- a virtual HP-IL controller, **V*il*MA** is prepared to work with V41 (V9D beta 3) – which comes with all firmware you need
  – or Emu71 (V1.11) **and** the ROMs to run it, *tnx JF*.
- at least one of the virtual IL devices you find here to download (latest version each), to start I suggest this set
  – *and* to have at least one description a single devices, e. g. video display,
- a text-only editor (Win32Pad V1.5.10.4) to modify VILMA.INI or else,
- ~~all files from the ZIP where you also found this **V*il*MA** description~~.

Installation

Have a plan before you start. There are three components to consider, the controllers, the devices, and **V*il*MA**.

Controllers
: will write state files, to which directory? Note it in VILMA.INI as value of CTLDR.

Devices
: will use ini files (file type VIL), where? Note the directory in VILMA.INI as value of VILDR.

V*il*MA
: will use VILMA files (to keep buildups). Note this in VILMA.INI as value of VIMDR.

Needless to say that you do need R/W access for those directories. Now install the components according to your plan. As they are somehow related I suggest to put OS/2 Pipelines files into the directory of your REXX installation. So you are sure to have it in the path.

To **prepare** controller state files useable for **V*il*MA** run your preferred one, either V41 or Emu71, equipped with an HP-IL module (virtually). Set in-port, out-port and out-address (localhost in most cases). **V*il*MA** will use the settings found in state files for the automatism of port number assignment.
Notes:
– Only the port numbers of devices will be changed, those of controllers remain unaffected.
– Under Emu71 the HPIL module may be "plugged" to other ports than on a real machine. **V*il*MA** expects it in **first place of Port0**.

Next **prepare** VIL files for each device you want to use with **V*il*MA**. To do so open a command window in the devices' directory. (This should still be the same as in your plan.) Now run the corresponding device simulator with the VIL file-ID as argument. Example for the small video display:

```
ILVideo smlVideo.VIL
```

Place the shown application window to a spot on the screen where you desire it to pop up in future too. Quit the simulator. This will produce the smlVideo.VIL file to be used by **V*il*MA**.

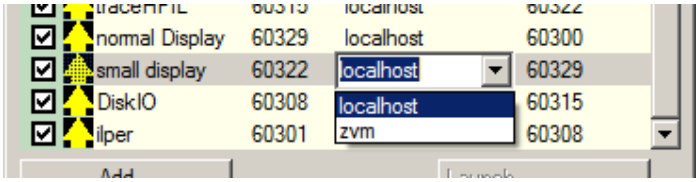
Connectivity

V*il*MA does not communicate with other systems, it only sets the proper settings in set-up files of programs that

will communicate. I recommend to read paragraph *TCP/IP Setup* in any of the device descriptions. **Note**: This proposed paragraph is not part of the documentation comprised in the IPv4 package of devices I suggested above.

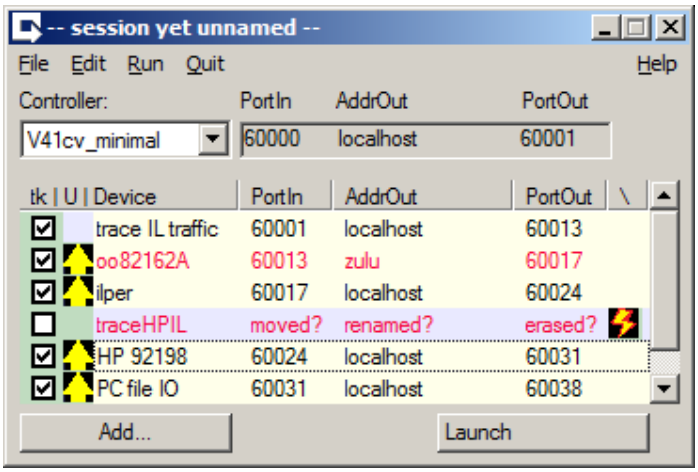
One word about **localhost** that **Vil MA** shows in column *AddrOut*. This host name is equal to dot-IP (or dotted IP) **127.0.0.1** - what is one of the widely known internal addresses of the computer you just use. To define additional mappings Windows 7 uses file C:\Windows\System32\drivers\etc\hosts. **Vil MA** respects it. I assume all devices in your list show 'localhost' as *AddrOut*.

But (!) you may change that. Selecting a device and click on its AddrOut will open this field for modification. In addition to pick one of the host names shown in the drop down list you may enter any address you like, as host name or dot-IP. (I do use this feature to link my own "devices" running on a mainframe (running under Hercules).



Vil MA increments the port number of such "outsiders" as if it would be one on localhost. On the other machine ensure to route virtual IL back to the shown address on your computer.

Controls of this Program




```
Keys: F1  - shows this help file (see note 1),
      F2  - displays the About information,
      Alt - or F10 activates menu,
      Esc - ends the program,
      Enter - when controller entry is selected and still shows 'VOID'
                a 'state file select window' opens.
Menu: If you wonder about the underlines see note 2.
      File              - Opens a sub-menu with following items:
         New            - starts a new, unnamed session (see note 3),
         Read...        - lets you choose a saved session (note 4),
         Write          - saves the current session (s. note 5),
         Save as...     - lets you choose a new name for the session to save,
         Exit           - ends the program immediately.
      Edit              - Opens a sub-menu with following items:
         Controller...  - to modify nickname, comment and file-ID,
         Add...         - to add one or more devices (VIL files),
         Get...         - merge devices from a saved session,
         Options...     - to modify VILMA.INI (see note 6).
      Run               - Opens a sub-menu to pick:
         Alter VILs     - save all VILs if port numbers changed,
         Selected Devices   - runs all check-marked devices,
         Controller     - runs the chosen controller.
         Shut Down Devices  - ends all devices VilMA started (s. note 10),
```

```
      Quit              - Ends all devices and VilMA, also saves
                             current session if named and altered,
      Help              - Opens a sub-menu with following items:
         About...       - displays information about VilMA,
         In detail...   - almost same as F1 pushed.
System Menu: as last items it offers an About...
                which does the same as Menu/Help/About... described above.
Controller Drop Down List:  - Choose a known controller
Buttons: Add... - Shows a file select dialog to "pick devices"
                   (same as Menu/Edid/Add...).
         Launch - if enabled combines all options of Menu/Run/
                    saves (if modified) all VIL files,
                    starts all checked devices,
                    starts (.6 seconds later) the controller.
```

The *device list* offers following capabilities:

- List devices you may pick up by `Menu/File/Read...`, `Menu/Edit/Add...`, `Menu/Edit/Get...` or the `Add...` button.
- Tick (check) or tick off a device with a click on the check-box in question (column `tk`), if not ticked the device is not included in the virtual loop to run.
- To move a device one slot up in the sequence click the up-icon (if shown, column `U` , the icon is not shown if it makes no sense to change the sequence).
- In case the naming of a device does not fit entirely in the column `Device` you may either hover the mouse cursor over the incomplete name and hold it for a second or so until it shows up fully in a *tool tip message or* — adjust the column width either by drag'n'drop vertical bars in the column header or a double click on them to align the width according the longest content.
- Switch columns by drag'n'drop column heading horizontally.
- Indicate by a text in red that this device is either *read-only* (its VIL file is write-protected), *a dummy* or *not found*, see notes 7 and 8,
  Note: files not found or corrupted will be indicated by questionable content in columns `PortIN, AddrOut, PortOut` .
- Choose another host for `AddrOut` : click an already selected device on its `AddrOut` box will open it to edit showing a drop down list to pick an already known address – or type a new one.
  Note: in contrast to other applications the only way to avoid an unwanted change is to push ESC.
- Remove a not ticked device from the list with a click on the lightning icon (column marked with *backslash*) by keeping the Ctrl key pressed the same time. The *remove column* is intentionally placed far off the check-box to avoid deletion by chance. Further you must press Ctrl.

Notes:

1. On your machine suffix `HTM` should be linked with a browser.
2. Underscored letters indicate which key will launch the option (when menu is activated by Alt or F10).
3. For accommodativeness the chosen controller remains as is. Only the list of devices is cleared.
4. The data saved for a *session* is the controller and the devices with each check-mark.
5. If last VILMA file read is read only, `Write` is disabled. If the session to save is yet unnamed `Write` acts like `Save as...`
6. The `INI` file is renamed to `TXT` and then the editor used which on your machine is linked to this suffix. It must save the file in a *text only* format. Before use of your modifications the original suffix is reconstituted.
7. The purpose of dummy or R/O devices is to have the chance to include devices not using VIL files as their input of settings, e. g. Dual-IL of HP-71B or similar. **V*il*MA** will respect the data but will not change it. It is users' task to put it right, **V*il*MA** may only control VIL based devices.
8. Dummy devices are defined by an entry beginning with `Placeholder` in the `[Run]` section. The search

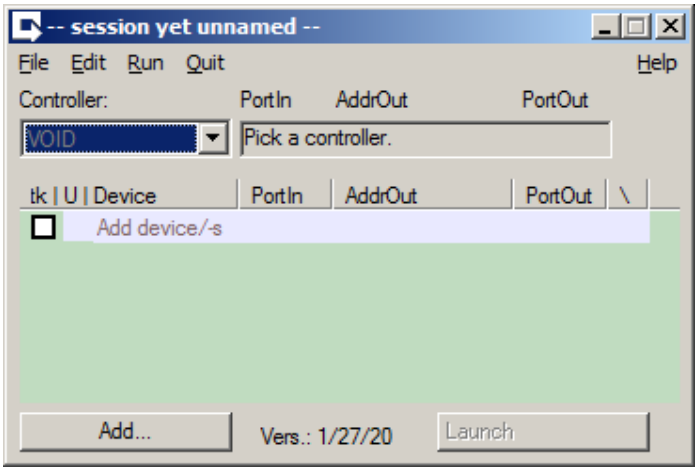
is case insensitive.

9. **Vil**MA is prepared to use V41 and Emu71 state files. The later are restricted to have the HPIL module set at first place of Port0 as the real machine.
10. **Vil**MA keeps a list of probable captions and tries to shut down those listed. If an application changed its caption (while running or with an update) this approach will fail.

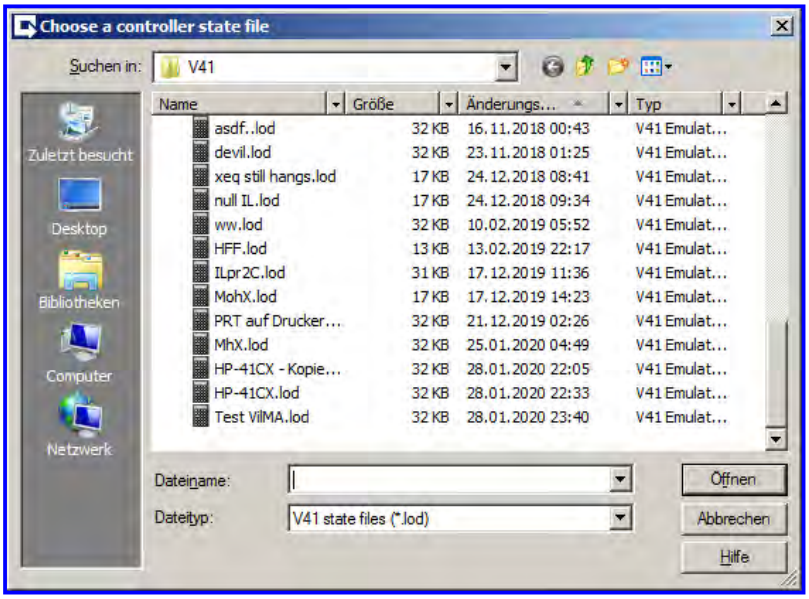
Examples

First sample run

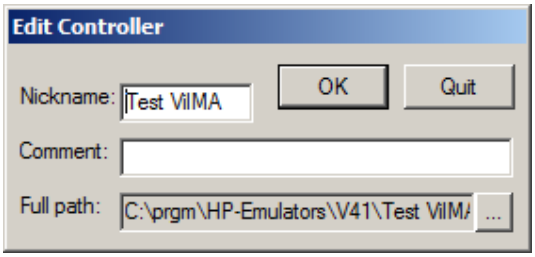
Start **Vil**MA as you would start every REXX program.



No controller and no device is defined. You may like to choose a controller first, so see if it already presented in the drop-down list (which is filled from the file defined in INI with variable CTRLS), if not and the field still shows `VOID` press *Enter* or select `Menu/Edit/Controller` . This will show controllers found in the directory set in INI with variable CTLDR.

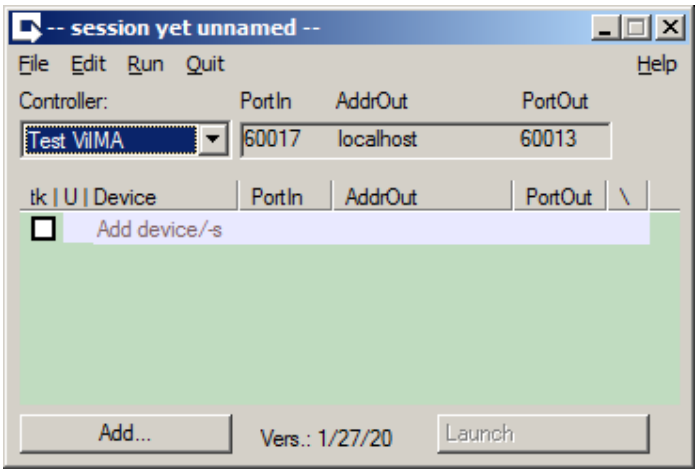


(I know, there are few too many, most are beta test relics.)
Picking one will show an "Edit controller" window:



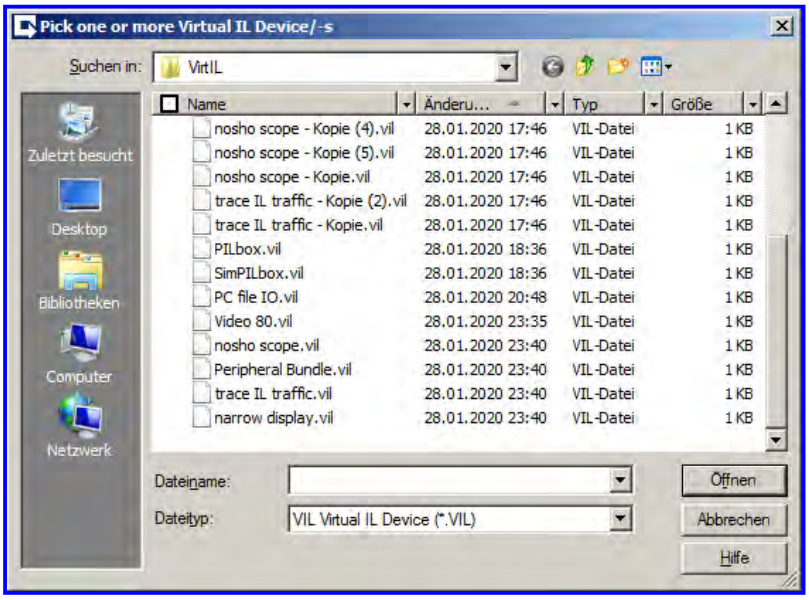
Here you have the chance to add a comment (could be used as tool tip in a future version) and you may change the nick name which is predefined by the state file's designation.

Push button `OK` and you have set the controller to use. As confirmation this worked you see now the controller's *PortIN*, *AddrOut* and *PortOut* — or if it not worked a hint about the reason (no IL module, file corrupted, moved, erased, renamed).

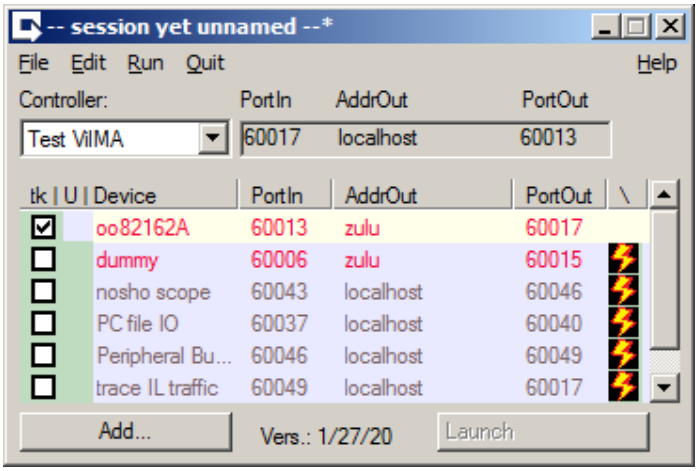


BTW, the `Menu/Run/Controller` is now enabled and you may run it for testing or changing some settings. If you changed the controller's *PortIN*, *AddrOut* and *PortOut* you should, after closing it, pick the current controller again from the drop down list. This rereads its state file to update the changed settings in **V*il*MA** too.

Now press the `Add...` button on the lower left to pick some devices. **Note**: you may choose several devices at once.

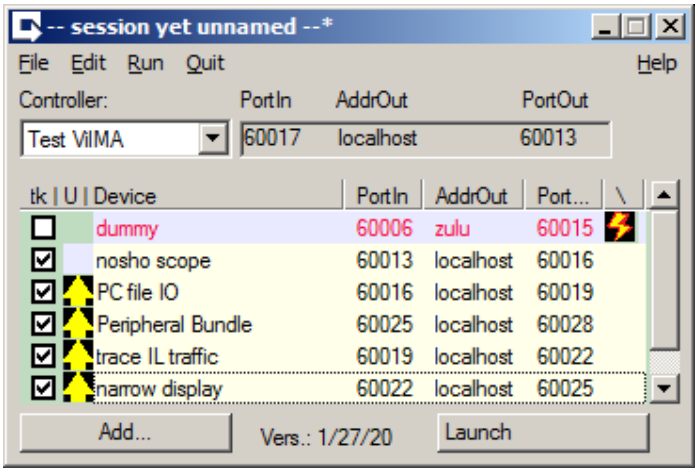


Where to find VIL files is set in VILMA.INI, variable VILDR. If there are no VIL files you should prepare some as mentioned in *Prerequisites*. You may select just all with a click on the check-box in the column head of file names. For this first try I picked only a few.



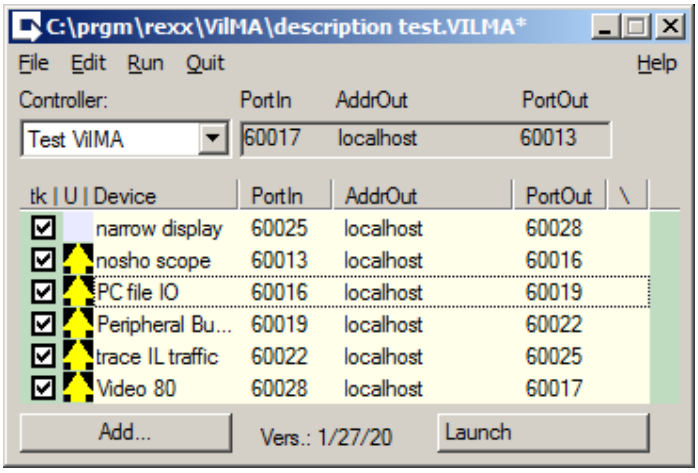
Oops, that is not what I wanted to show now. Only one device is executable and all the rest is deselected — why?
The lines marked in red are write-protected files or placeholders **not to be changed** by **Vil*MA***. Apparently the first outlier matches the in- and out-ports of the controller and for this no other device may be forced in between.
The motive for placeholders are device simulators not using VIL files to set ports and out-address.

What to do now to run the other devices? Two options, either start the controller to change its settings, or remove the first dummy. I did the later, `Menu/File/New` to empty the list, once more `Add...` and select all but the unwanted device.
Looks different now:

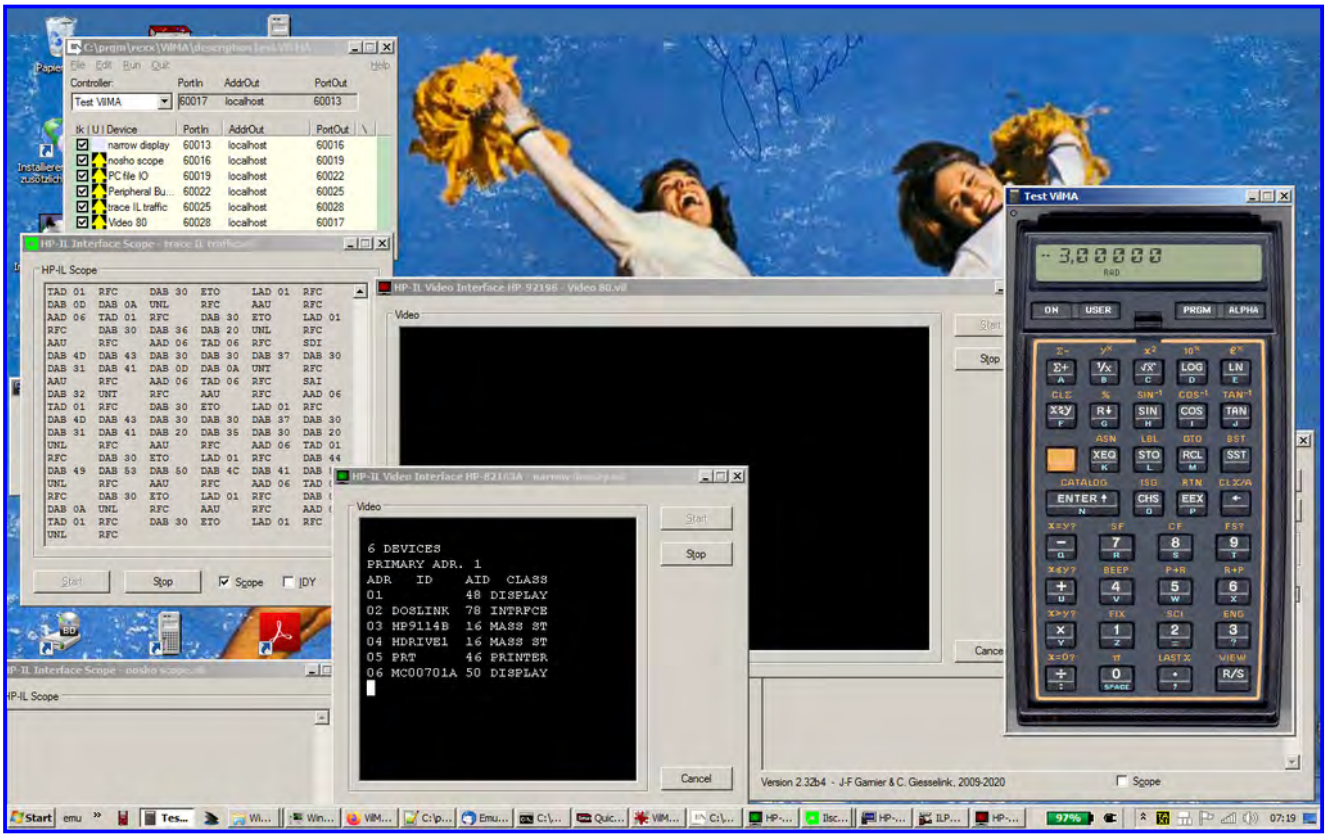


May be, you like to keep this bundle for later. So do `Menu/File/Write` and give it a name. The directory where VILMA files are saved is defined in the INI, variable VIMDR.

Is this the loop you'd like to run? I removed the red line - `Ctrl` and a click on the lightning icon to the right - and moved up the *narrow display* to the top - click on the up-arrow to the left of it.



Now — push `Launch` and see what happens.



To see if it works I ran the program `LCAT` (you know it, from the Extended I/O module). The output is shown on the narrow display in the middle. In addition a trace of the IL traffic is shown in the *HP-IL Scope* window.

And how to stop it all? Two ways: `Menu/Quit` or `Menu/Run/Shut Down Devices`. Both commands will not end the controller (it may still be useful without all those peripheral devices), the later will keep **Vil*MA*** running like here:

More samples?

The detailed example above gives enough insight, I suppose. So you may imagine the remainig functions of **V*il*MA** work as well.

Final Remark

Frankly, I assume no one but me will ever run this program. The main hurdle is to get OS/2 Pipelines outside IBM. Nonetheless I described **V*il*MA** to show a working example how an IL Device Manager could look like. With it **connecting virtual devices is now as simple as plugging real cables**. That was my goal.

Last but not least I like to thank Christoph Gießelink for i) enhancing V41 to be HPILable, ii) for the development of several IL devices, some in close collaboration with JF Garnier, and iii) for the documentation – or should I say description? – how to find AddrOut, PortOut and PortIn within an Emu71 state file.

M., in January 2020.

The fine print

(i) *Do not use this program*.
(ii) Any damage you cause with this program, intentional or by chance, is a violation of clause (i). Thus *all consequences are completely at your own risk*.
(iii) If you manage to improve this program you have to inform me as stipulated in following THE Q PUBLIC LICENSE (QPL).
(iv) In addition applies the QPL below:

THE Q PUBLIC LICENSE version 1.0

Copyright (C) 1999 Trolltech AS, Norway.

Everyone is permitted to copy and distribute this license document. The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model.

This license applies to any software containing a notice placed by the copyright holder saying that it may be distributed under the terms of the Q Public License version 1.0. Such software is herein referred to as the Software. This license covers modification and distribution of the Software, use of third-party application programs based on the Software, and development of free software which uses the Software.

Granted Rights

1. You are granted the non-exclusive rights set forth in this license provided you agree to and comply with any and all conditions in this license. Whole or partial distribution of the Software, or software items that link with the Software, in any form signifies acceptance of this license.
2. You may copy and distribute the Software in unmodified form provided that the entire package, including - but not restricted to - copyright, trademark notices and disclaimers, as released by the initial developer of the Software, is distributed.
3. You may make modifications to the Software and distribute your modifications, in a form that is separate from the Software, such as patches. The following restrictions apply to modifications:
   a. Modifications must not alter or remove any copyright notices in the Software.
   b. When modifications to the Software are released under this license, a non-exclusive royalty-free right is granted to the initial developer of the Software to distribute your modification in future versions of the Software provided such versions remain available under these terms in addition to any other license(s) of the initial developer.
4. You may distribute machine-executable forms of the Software or machine-executable forms of modified versions of the Software, provided that you meet these restrictions:
   a. You must include this license document in the distribution.
   b. You must ensure that all recipients of the machine-executable forms are also able to receive the complete machine-readable source code to the distributed Software, including all modifications, without any charge beyond the costs of data transfer, and place prominent notices in the distribution explaining this.
   c. You must ensure that all modifications included in the machine-executable forms are available under the terms of this license.
5. You may use the original or modified versions of the Software to compile, link and run application programs legally developed by you or by others.
6. You may develop application programs, reusable components and other software items that link with the original or modified versions of the Software. These items, when distributed, are subject to the following requirements:
   a. You must ensure that all recipients of machine-executable forms of these items are also able to receive and use the complete machine-readable source code to the items without any charge beyond the costs of data transfer.
   b. You must explicitly license all recipients of your items to use and re-distribute original and modified versions of the items in both machine-executable and source code forms. The recipients must be able to do so without any charges whatsoever, and they must be able to re-distribute to anyone they choose.
   c. If the items are not available to the general public, and the initial developer of the Software requests a copy of the items, then you must supply one.

Limitations of Liability

In no event shall the initial developers or copyright holders be liable for any damages whatsoever, including - but not restricted to - lost revenue or profits or other direct, indirect, special, incidental or consequential damages, even if they have been advised of the possibility of such damages, except to the extent invariable law, if any, provides otherwise.

No Warranty

The Software and this license document are provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Choice of Law

This license is governed by the Laws of England if still member of the EU.
