

don't code today what you can't debug tomorrow

oo7470A

A HP-GL Plotter Linked to Virtual IL

Allschwil Meeting 2014

Executant:Mike

Today I present the outcome of several year's work with some interruptions. This length is the reason for the motto, because today not only some code, even the comments are incomprehensible or cryptic. I'd like to talk about the result, not the reasons for the long way from vision until today.

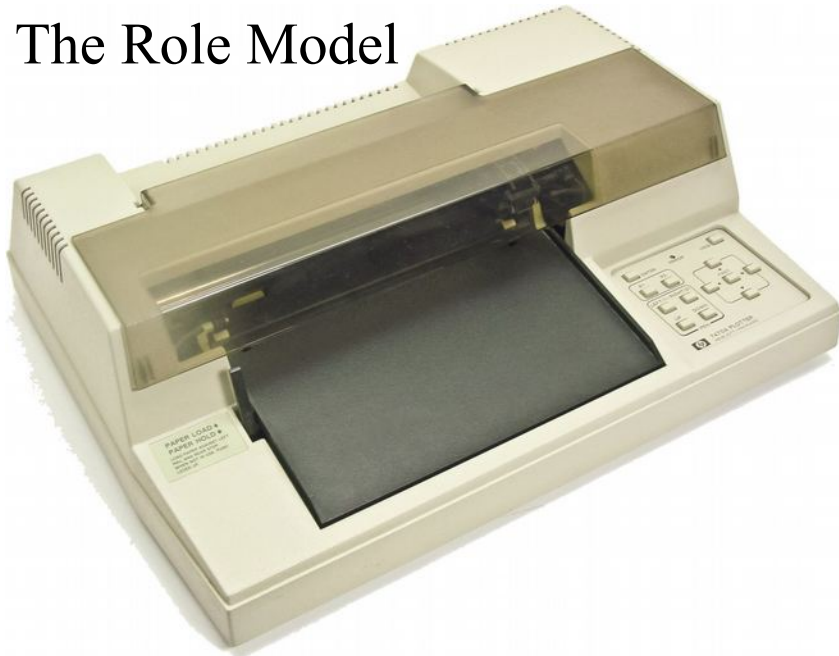
Agenda for the next 10 minutes:

- What is it all about? (The role model)
- What I had in mind as virtual replacement
- It is what it is (few aspects of the result)
- How it works (building blocks of **oo7470A**)
- The IL interface according to 82166-90017
- The HP-GL interpreter like 07470-90001
- A live presentation of **oo7470A**
- Conclusion, [Q&A](#), β -testers wanted, links

That is what I prepared for today.

To make it effective, please be so kind and ask after my presentation - if you still have some questions

The Role Model



The HP7470A, introduced 1982, was the first small format paper-moving plotter. Earlier pen plotters fixed the paper but moved the pen in x and y direction. The 7470A used two pens to draw on sheets up to DIN A4. It was available with an HP-IB, RS-232 or HP-IL interface. The later had the smallest set of HP-GL commands.

HP-GL?? Hewlett-Packard Graphics Language was the primary control for HP plotters which became with the time a standard for almost all plotters.

The Virtual Parallel

- Link to Virtual IL as specified by Christoph
- Simulate the IL interface described in *The HP-IL Interface Specification* (82166-90017)
- Act almost as described in *HP 7470A Graphics Plotter, Interfacing and Programming Manual* (07470-90001)
- Output: i) a HGL and ii) a PNG file, iii) display graph on screen

Caused
most
trouble

What I had in mind was a logical rebuild as close as possible to the original. For this I followed the two documents mentioned here.

A non-Windows-version worked quite sufficient for me since more than a year. But this was not good enough to publish. So I had to make it a Windows compliant application - what was not so easy for a mainframe dinosaur like me.

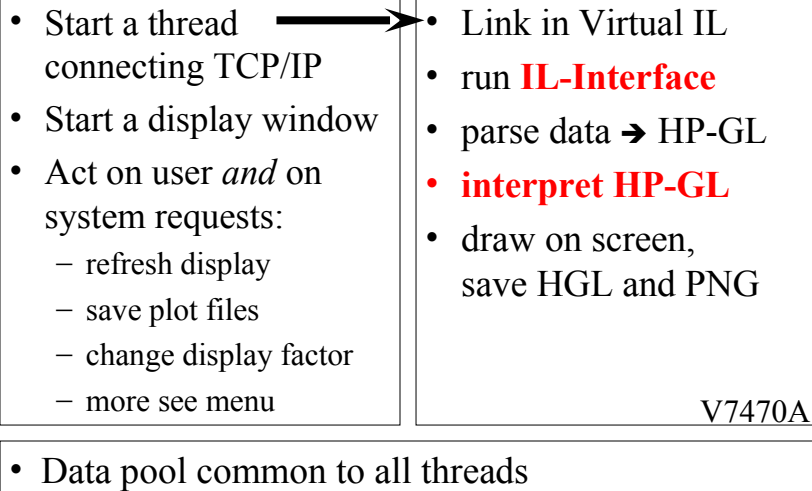
The Result

- It runs on PCs
- It's coded with ooREXX and ooDialog, for that not to compile to an EXE
- Why ooREXX?
 - I know REXX from the IBM host
 - It is simple and transparent for all → easy reuse
 - But: ooREXX is much more than REXX
- Today my **oo7470A** works, still improvable

With this slide about the result it is obvious, that I am not convinced and not too overwhelmed. I am proud that it is ok to present, alas it lacks the final touch. May be I am too demanding and ooREXX is not the tool for a shiny finish.

One reason to use ooREXX was the vision to get besides a nice application also a piece of code *everybody* may reuse for his own Virtual IL device.

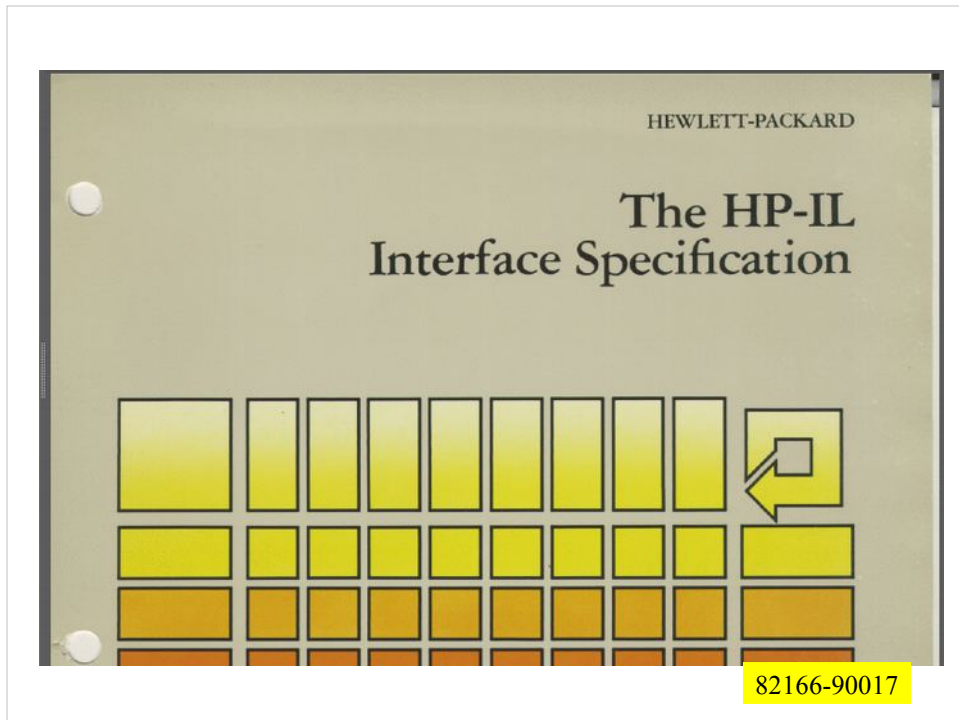
Main Parts of oo7470A



The building blocks of oo7470A worth to mention are:

- The virtual plotter with its sections as listed here (~2000 lines of code)
- The user interface, that respects also some needs of the underlying OS (~1000 lines of code)

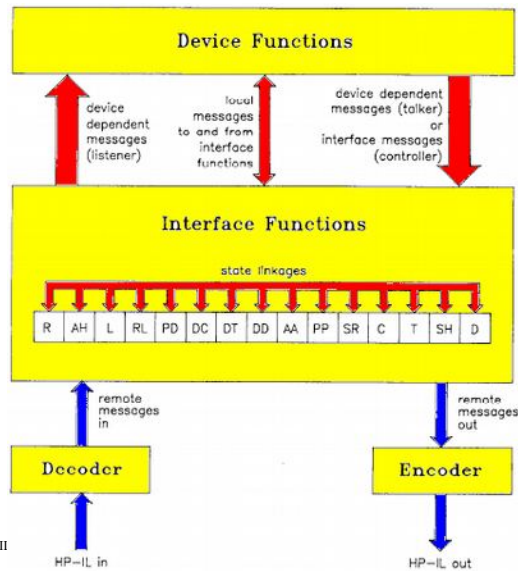
Of cause this diagram does not show all details. What is marked in red I will explain a bit more closely now.



This specification describes the internal logic of the IL interface 'on its own', without relation to a device

((scroll through some pages))

The HP-IL Interface

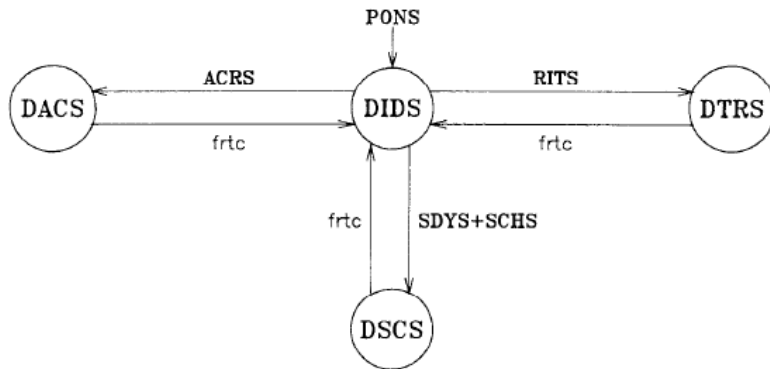


Source: 82166-90017 'The HP-II Interface Specification'

This is the fundamental block diagram of the IL interface. The lines in blue are no problem, this is Christoph's Virtual IL. But the rest is completely new ground - at least for me.

- R - Receiver
- AH - Acceptor Handshake
- L - Listener
- RL - Remote Local
- PD - Power Down
- DC - Device Clear
- DT - Device Trigger
- DD - Device Dependent Command
- AA - Automatic Address
- PP - Parallel Poll
- SR - Service Request
- C - Controller
- T - Talker
- SH - Source Handshake
- D - Driver

D (Driver) Interface Function



Source: 82166-90017 'The HP-IL Interface Specification'

Here one of the IF functions as an example.

PONS - power on state

DIDS - driver idle state

DACS - driver transmit from acceptor state

DSCS - driver transmit from source state

DTRS - driver transfer state

ARCS - acceptor ready state

SDYS - source delay state

SCHS - source command handshake state

RITS - receiver immediate transfer state

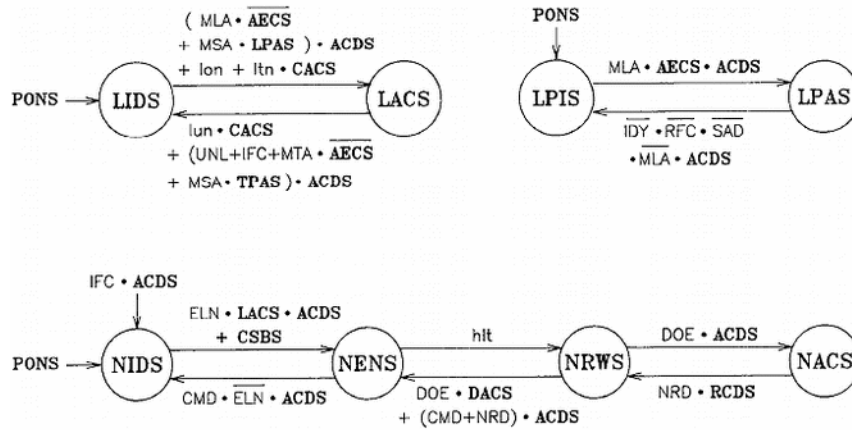
frtc - frame transmission complete

D (Driver) Interface Function

```
/*
** D (Driver) Interface Function
** -----
** All devices must have the complete D function implemented.
*/
IFFD:
  prs = lif.D
  if sv.DIDS then select
    when sv.ACRS then lif.D = 'DACS'
    when sv.RITS then lif.D = 'DTRS'
    when sv.SDYS | sv.SCHS then lif.D = 'DSCS'
    otherwise nop
  end
  else select
    when actmsg = 'frtc' then lif.D = 'DIDS'
    when sv.PONS then lif.D = 'DIDS'
    when prs = "dpsl" then say 'Driver IF not initialised yet.'
    otherwise nop
  end
  if prs \= lif.D then do
    call stacha('D')
    if ¬sv.DIDS & wordpos('frtc', msgstk)=0 then,
      call senDat          /* DTRS is not really device specific */
    end
  end
  return
```

... and how I coded it.

L (Listener) Interface Function



Source: 82166-90017 'The HP-IL Interface Specification'

One more example

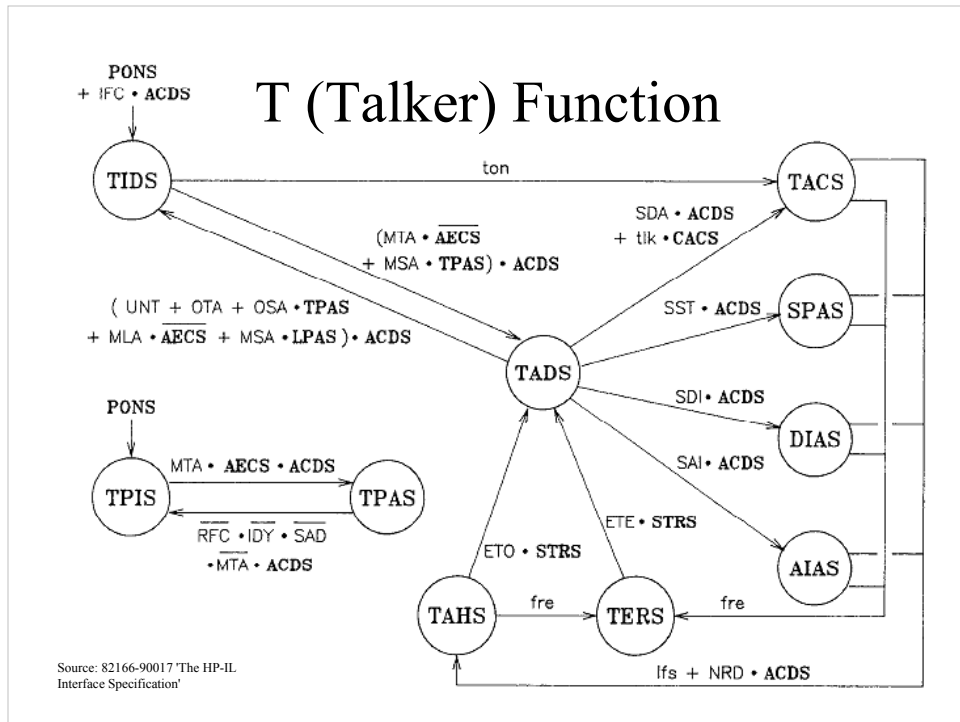
L (Listener) Interface Function

```
/*
** L (Listener) Interface Function
** -----
** L1 Basic capability      Implement LIDS, LACS
** not yet:
** L3 Extended listener    Additional to L1; implement LPIS
**                          and LPAS (requires AA2 or AA3)
**
*/
IFFL:
  prs = lif.L
  select
    when sv.LIDS then if rm.MLA & ¬sv.AECS | rm.MSA & sv.LPAS & sv.ACDS,
      | actmsg = 'lon' | actmsg = 'ltn' & sv.CACS then lif.L = 'LACS'
    when sv.LACS then if actmsg = 'lun' & sv.CACS | rm.UNL | rm.IFC,
      | rm.MTA & ¬sv.AECS | rm.MSA & sv.TPAS & sv.ACDS then lif.L = 'LIDS'
    otherwise
      if sv.PONS then lif.L = 'LIDS'
      else say 'Listener IF not yet initialised.'
  end
  if prs \= lif.L then call stacha('L')
return
```

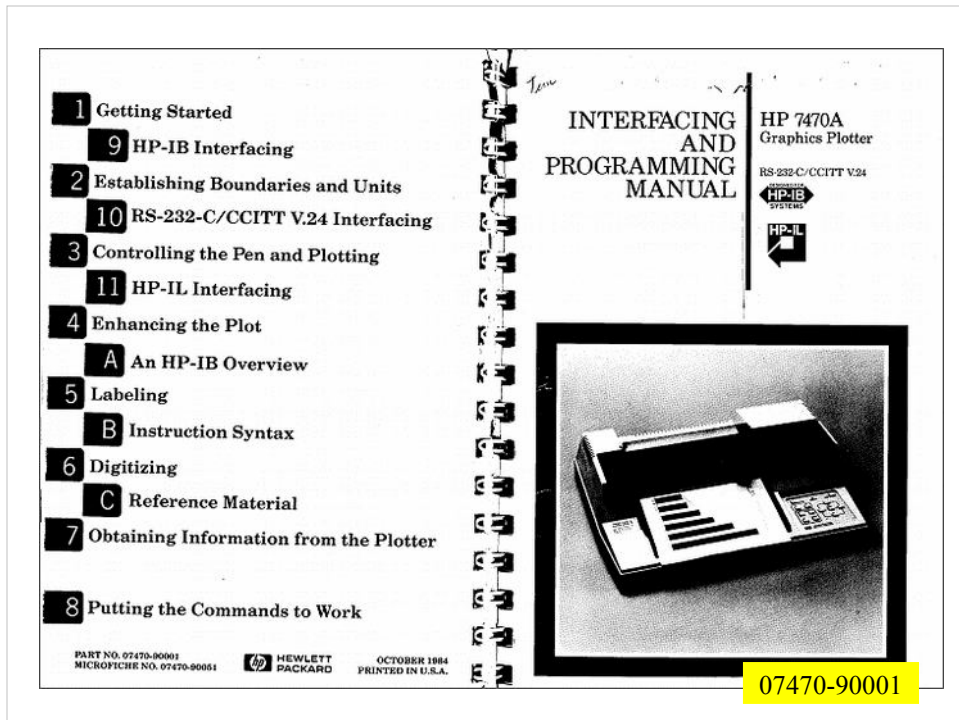
See : 07470-90001,
p. 11-2 (96)

And that's all. Why? There are subsets of the IF functions and here only L1 is implemented.

Note: the only “output” of this IF function is to change its state. In contrast to the driver function before (which, BTW, is the only path to the device, at least in my implementation)



Someone likes to get an explanation of this?



This is the device specific documentation of HP-GL.
((Scroll through some pages))

Example

Interpret HP-GL

The Absolute Direction Instruction, DI

DESCRIPTION The absolute direction instruction, DI, specifies the direction in which characters are lettered.

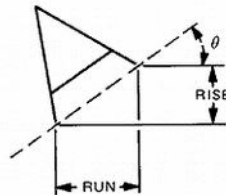
USES The instruction can be used to change the direction of labeling to a new absolute direction; by absolute we mean independent of P1,P2 settings. It is especially useful for labeling a Y-axis or labeling a vertical graph.

SYNTAX *DI* run, rise terminator
or
DI terminator

EXPLANATION Run and rise are in decimal format, 0 to ± 127.9999 , and specify the direction according to the relationship:

$$\theta = \tan^{-1}\left(\frac{\text{rise}}{\text{run}}\right)$$

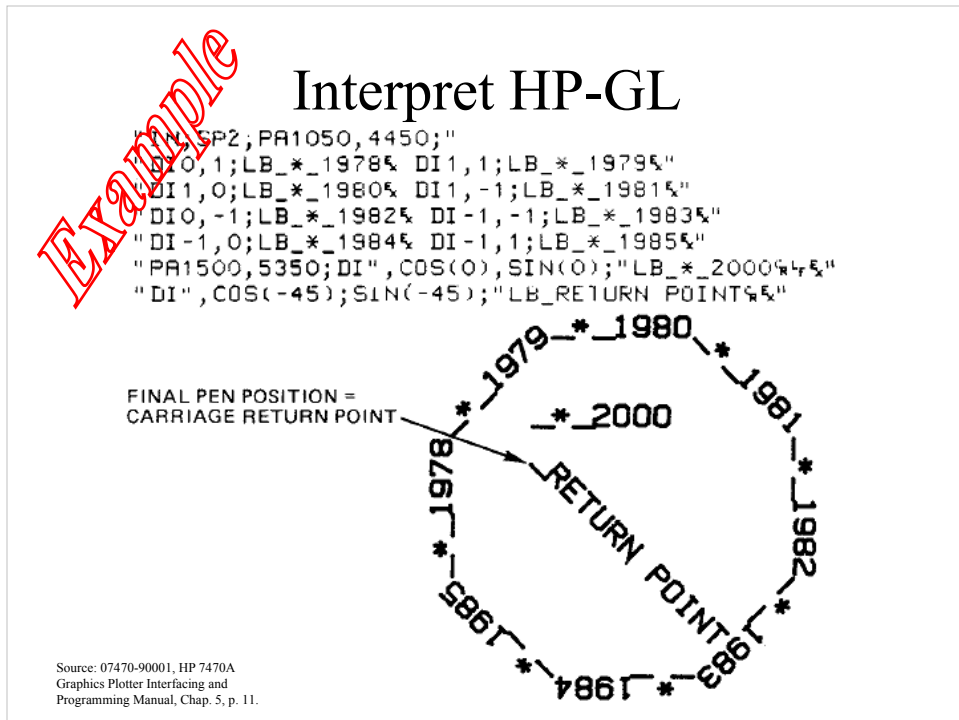
where:



$$\begin{aligned}\text{rise} &= \text{SIN}(\theta) \\ \text{run} &= \text{COS}(\theta)\end{aligned}$$

Source: 07470-90001, HP 7470A
Graphics Plotter Interfacing and
Programming Manual

Just an example



Mostly there is a piece of code and the result of it, which shows for what and how the GL command in question may be used.

((Chapter 5, p. 11 -- show rendering of oo7470A, 7470.exe, and hpglview.exe))

This leads to the used letters or font...

Fonts from Olivier De Smet

```
/* "Fonts" of the HP7470A plotter */
/* Undef */ chagra. = '' /* NOP */
/* ETX is special, CR, SO and SI also */
/* BS */ chagra.8 = '0,-48, 0'
/* LF */ chagra.10 = '0, 0,-64'
/* VT */ chagra.11 = '0, 0, 64'
/* Blank */ chagra.32 = '0, 48, 0'
/* Olivier De Smet was so kind and sent me his solution: */
/*
 * [taken from] Hp-7470.h
 * Copyright 2006 Olivier De Smet
 */
/* ! */ chagra.33 = '0, 0, 0,1, 2, 0,1, 2, 2,1, 0, 2,1, 0, 0,0, 1, 11,1, 1, 32'
/* " */ chagra.34 = '0, 10, 26,1, 10, 36,0, 22, 26,1, 22, 36'
/* # */ chagra.35 = '0, 4, 0,1, 16, 32,0, 16, 0,1, 28, 32,0, 0, 12,1, 32, 12,0, 32, 20,1, 0, 20'
/* $ */ chagra.36 = '0, 0, 6,1, 2, 3,1, 7, 1,1, 25, 1,1, 30, 3,1, 32, 6,1, 32, 11,1, 30, 14,1,
25, 16,1, 7, 17,1, 2, 19,1, 0, 22,1, 0, 26,1, 2, 29,1, 7, 31,1, 24, 31,1, 29, 29,1, 31, 26,0, 16, -
4,1, 16, 36'
/* % */ chagra.37 = '0, 0, 24,1, 1, 22,1, 4, 21,1, 10, 21,1, 13, 22,1, 14, 24,1, 14, 29,1, 13, 31,1,
10, 32,1, 4, 32,1, 1, 31,1, 0, 29,1, 0, 24,0, 18, 3,1, 19, 1,1, 22, 0,1, 28, 0,1, 31, 1,1, 32,
2,1, 32, 8,1, 31, 10,1, 28, 11,1, 22, 11,1, 19, 10,1, 18, 8,1, 18, 3,0, 0, 0,1, 32, 32'
/* & */ chagra.38 = '0, 32, 13,1, 32, 7,1, 30, 3,1, 26, 1,1, 21, 0,1, 11, 0,1, 6, 1,1, 2, 3,1,
0, 7,1, 0, 12,1, 2, 16,1, 8, 19,0, 29, 25,1, 29, 28,1, 27, 31,1, 22, 32,1, 11, 32,1, 6, 31,1, 4,
28,1, 4, 24,1, 6, 21,1, 32, 0'
/* ' */ chagra.39 = '0, 15, 27,1, 17, 28,1, 18, 29,1, 18, 34,1, 15, 34,1, 15, 32,1, 18, 32'
/* ( */ chagra.40 = '0, 32, -4,1, 28, 0,1, 25, 5,1, 24, 11,1, 24, 23,1, 25, 27,1, 28, 32,1, 32, 36'
/* ) */ chagra.41 = '0, 0, -4,1, 4, 0,1, 7, 5,1, 8, 11,1, 8, 23,1, 7, 27,1, 4, 32,1, 0, 36'
/* * */ chagra.42 = '0, 4, 4,1, 28, 28,0, 4, 28,1, 28, 4,0, 0, 16,1, 32, 16'
/* + */ chagra.43 = '0, 0, 16,1, 32, 16,0, 16, 4,1, 16, 28'
/* , */ chagra.44 = '0, 0, -5,1, 2, -4,1, 3, -3,1, 3, 2,1, 0, 2,1, 0, 0,1, 3, 0'
/* - */ chagra.45 = '0, 0, 16,1, 32, 16'
...
```

1981 SALES BY REGION
1981 SALES BY REGION

Olivier De Smet once was so kind and sent me his vector representation of the HP7470A's letters. That way I could take many hurdles in one jump.

Well, it turns out that there are some minimal deviations, but for now I leave it as is.

An unparalleled vector joke

$$\lambda = \sqrt{\frac{(\cancel{A_x} + \lambda(B_x - A_x) - \cancel{A_x})^2 + (\cancel{A_y} + \lambda(B_y - A_y) - \cancel{A_y})^2}{\lambda^2(B_x - A_x)^2 + \lambda^2(B_y - A_y)^2}} = \lambda(|AB|)$$

$$\lambda = \frac{\lambda}{|AB|}$$

$$T_x = A_x + \lambda(B_x - A_x)$$

$$T_y = A_y + \lambda(B_y - A_y)$$

pattern 0...21
LT → alpha

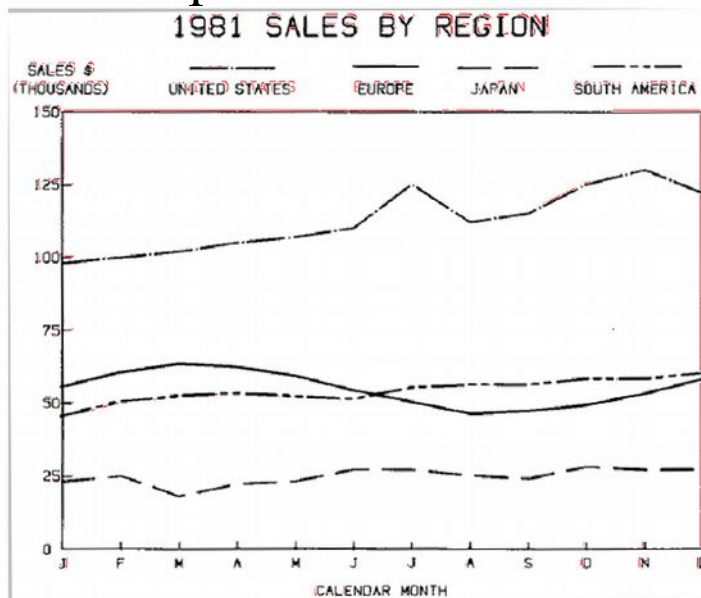
21 - alpha
Whip - ?

An other hurdle was:

line patterns are only started at PU. All other pen moving commands continue the pattern. Here my notes on how to accomplish this. The funny thing, by adding vectors I came to the **theorem on intersecting lines** (Strahlensatz in German). *What a finding!*

So no vectors needed to draw patterns, **rule of three** (Dreisatz) is enough.

Output almost identical



Here a visual compare of an example from the manual, here in black,

and now in red the output of my **oo7470A**. When I first saw this I was amazed that even the position of the line patterns sections are identical.

You may applaud now.

To be honest, there is one more deviation than in the letters: currently I do not render round line ends.

The free REXX/gd I use does not what I expect.

Now to a live example...

Summary

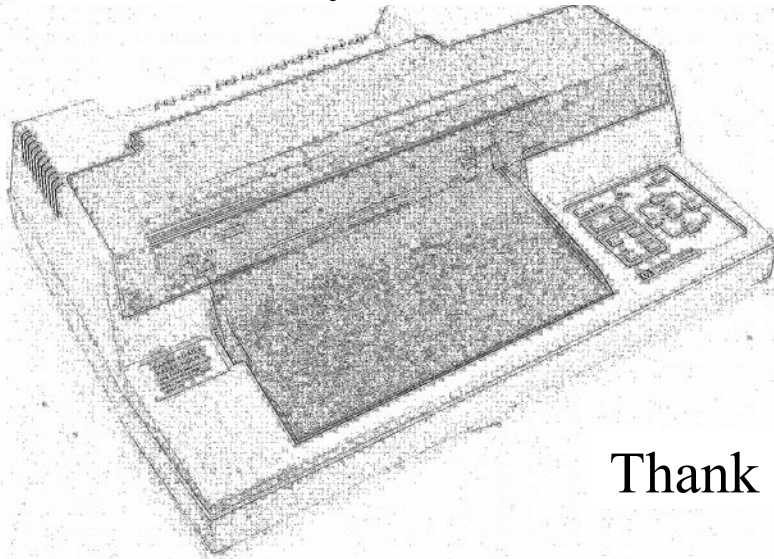
- Was it worth while? Yes and no:
 - **No!** of cause not, it is just a hobby.
 - Yes, may be the IL interface could help when coding next virtual device
- **oo7470A** still requires completion
- Nevertheless β -testers welcome
- ooREXX is not REXX, for a nice UI on Windows I was *much* faster using Delphi



My experience from this project: it is fundamental what tools you use. But after 2000 lines of code with the essential core of this project, the Virtual 7470A working ok, I feared the task to re-code it in Delphi. So I continued to embed it in Windows with ooREXX. Hard cheese.

This 4-banger - sorry, non-RPN, not HP, but the first of a kind assembled in Germany - was the first Windows compliant program I ever made and finished it in a few days only. Of cause it's at most an UI only, behind the surface it is not comparable to **oo7470A**.

Any Questions?



Thank you

Q from my side: who wants to be β -tester?

Links

- ooREXX
<http://sourceforge.net/projects/ooress/files/ooress>
- ooDialog
<http://sourceforge.net/projects/ooress/files/ooDialog>
- REXX/gd <http://rexxgd.sourceforge.net>
- IrfanView <http://www.irfanview.com>
- CERN HP-GL viewer <http://cern.ch/hpglview>
- John Miles' Intelligent Plotter Emulator
<http://www.ke5fx.com/gpib/7470.htm>
- My 4-banger <http://www.stehlin.net/hpclub/Aristo/4-banger.zip>

Here some links,

the first three are needed to run **oo7470A**,

IrfanView is needed only for printing a plot (may easily be replaced with your choice),

the rest was just mentioned in my slides or used in the presentation.