WORKING on the Philips P2000C

Francis Massen (francis.massen@education.lu)

File: Working_on_the_Philips_P2000C.doc (or .docx and .pdf)

version: 1.0 date: 21 Jan 2026

INDEX:

1. Short description of the P2000C, type 2010-1

2. Creating system and other software disks

- **2.1. First steps with GreaseWeazle**
- **2.2. The structure of the disks**
- **2.3. Creating disks from software images**
  - **2.3.1. Greaseweazle**
  - **2.3.2. The CopyStation with ImageDisk and Sybex tools, and Greaseweazle compatibility**

3. The undispensable Philips utilities

4. CP/M emulators on the Windows PC

5. Transfering data into and out of the P2000C

- **5.1. Plain text transfer**
- **5.2. Transfer of binary data**

6. Overall conclusion

References

Addendum

1. Short description of the P2000C, type 2010-1

The Computarium acquired in November 2025 a vintage Philips P2000C (type 2010-1) computer, which was delivered in an excellent physical state, but without any software.
This computer from the 1983 is a CP/M machine with 2 Z80 microprocessors running at 4 MHz, 64kB RAM, a green CRT, two TEAC FD-55F 5.25" diskettes and several ports. For more information see [**ref.1**].

This computer was sold as luggable, and has a weight of 15 kp. For the transport, the keyboard (here a CH-DE layout) is put keys-down on the front-side, where it is maintained by 2 claws connected to a sturdy carry-ribbon, which practically must be put over a shoulder to make a non back hurting transport.

The computer is a CP/M machine running CP/M 2.2, and was known as one of the fastest in its time. I could be fitted with several options:

- a "Copower" board which was a board with an Intel 8080 uP, allowing to run standard DOS software.
- a memory expansion
- an IEEE-488 board for communication with IEEE devices
- a 10 MB hard disk (SASI standard)

None of these extensions are inside the Computarium's P2000C.

The picture on the title page shows the computer with its main cover removed.

The following picture shows the Mitsubishi 9" CRT and at the rear (here bottom side) the vertically mounted motherboard.

The 2 floppy drives are below the metal sheet with the circular openings.

Without any disk in one of the drives, the screen only shows the BIOS message **IPL 1.1**. and than the text **SYSTEM DISK?**

So the first problem after having received the computer, was to make a compatible CP/M 2.2 system boot disk and other disks with software and utilities..

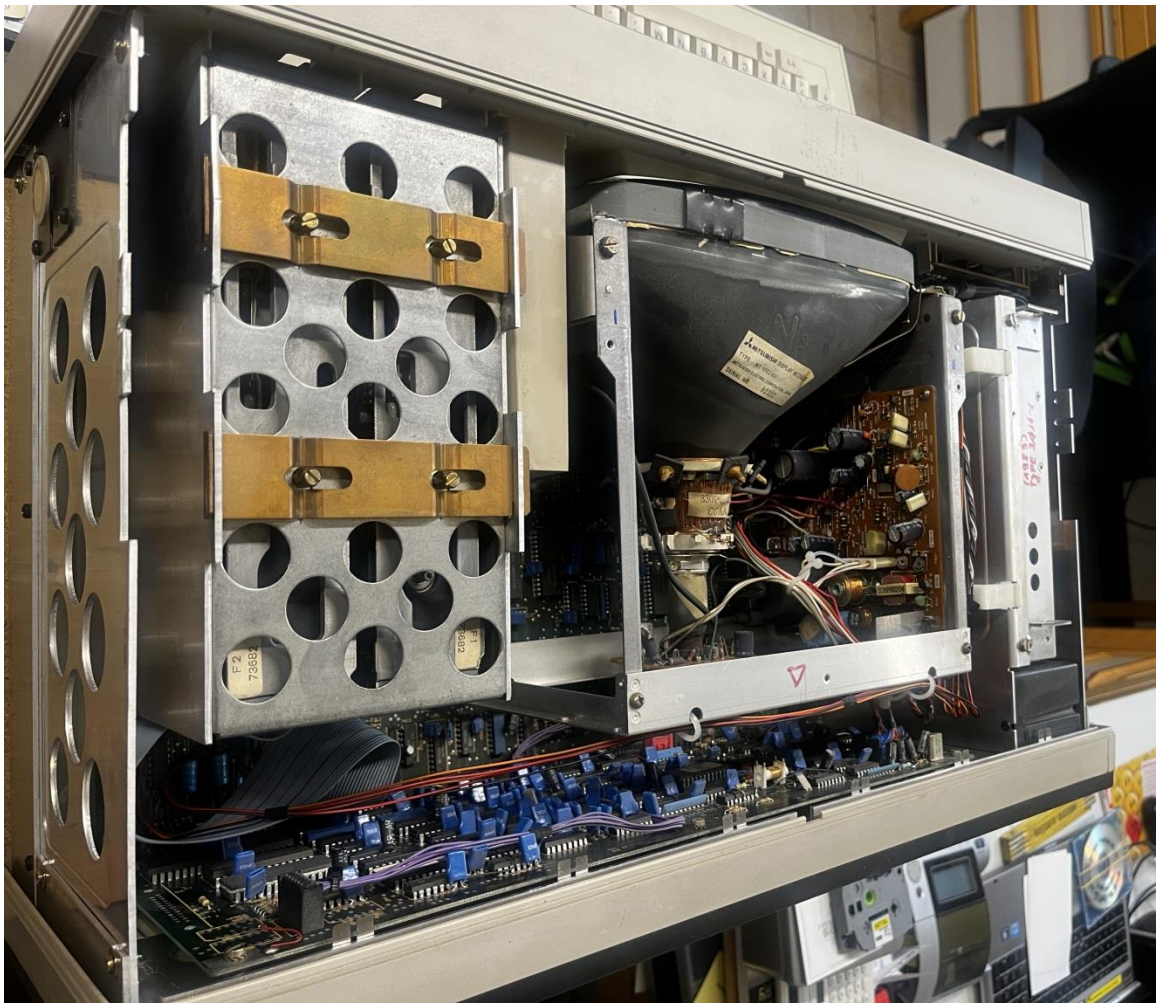

Fig. 1 View into the P2000C (type 2010-1)

2. Creating a system and other software disks

Before going into the details of the disk structure, I will just give a short recall of my first trials to make system disks..

On the Internet several web-sites have images of different CP/M 2.2 disks sold for the P2000C, some of which also have specific software added by Philips.
One of the best known schemes for making software images of floppy disks is **Dave Dunfield's IMD** (= ImageDisk) software, that runs in plain DOS (see **[ref.2]**).
Among the large list of different systems there is a ZIP file for the Philips 2000, which when extracted yields six image files P2000_1.IMD to P2000_6.IMD. The first one is the main system disk.
Another web-site to be used is **electrickery.nl** from F.J. Kraan [**ref. 3**]

2.1. First steps with GreaseWeazle

By chance I had ordered and received the Greaseweazle board developed by Keir Fraser to make images (flux type and others) from vintage floppy disks.
(see **[ref.4]** for his GitHub web-site, and **[ref.5]** for a UK vendor of the board)..

This board can be compared to the Kryoflux, with the difference that this is an open source project, and that the Greaseweazle is much cheaper than the Kryoflux.

It is essentially an USB interface between a "modern" computer (running for instance a decently recent Windows version or Linux or MacOS) and an original floppy drive. In our case that means a 5.25" floppy drive, rotating at 300 RPM, with 80 tracks, two heads and MFM encoding at 250 kHz. From the logical structure, the Philips CP/M 2.2 asks for 16 sectors per track, each sector storing 256 bytes. The first 2 tracks hold the CP/M system, which can be installed even later on when a disk has its data and software programs written on it.

The 16 sectors by track is not a run-of-the-mill propriety, as most systems have 9 or 10, or even a lesser number of sectors per track.

By chance, the BBC Master (or ACORN) computer which we used in a great number in the 1980's, had 16 sectors/track with the ACORN ADFS (Advanced Disk Filing System). Many drives (sold for instance by Cumana) were 80 track DSDD drive, often with a switch to operate at 40 tracks for compatibility with the older DFS of the BBC model B microcomputer. The Greaseweazle list of disk definitions had such a drive definition called acorn.adfs.640.
So I used an old CUMANA drive, and without much reflecting about the many floppy parameters like the interleave, I wrote the P2000_1.IMD onto a 5.25" DSDD disk.
By sort of a miracle, this disk was bootable and started the CP/M system.
The following two pictures show the hardware at home (laptop, Greaseweazle board and Cumana drive) and the first screen of the properly booted system.

Keyboard ASCII means that the correct driver for the CH-D keyboard is not yet installed.

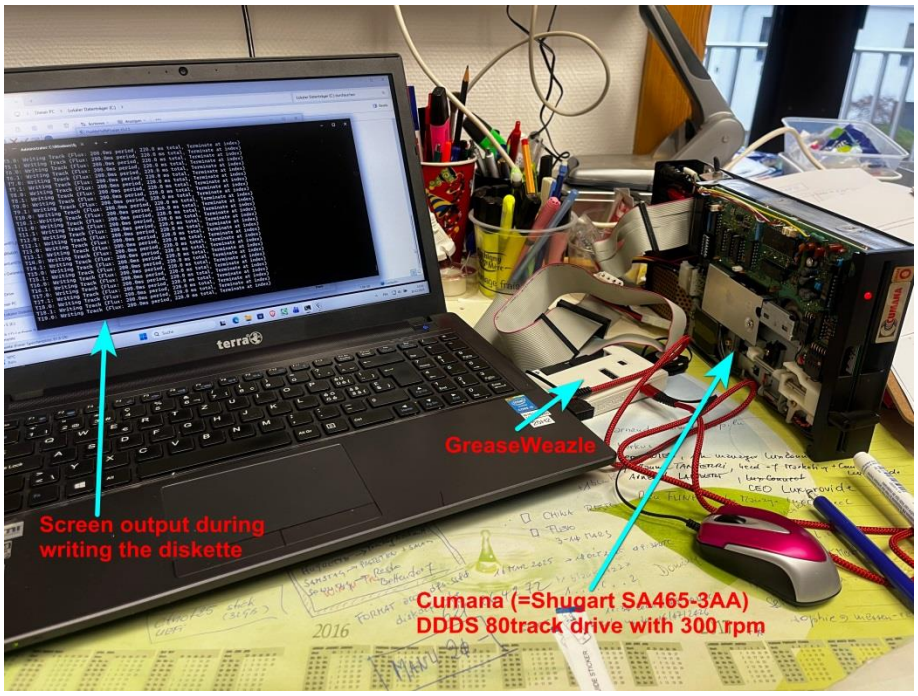

Fig.2. Laptop (with Win11), Greaseweazel on USB port and Cumana enclosure with the 5.25" floppy drive (actually a Shugart SA465-3AA)

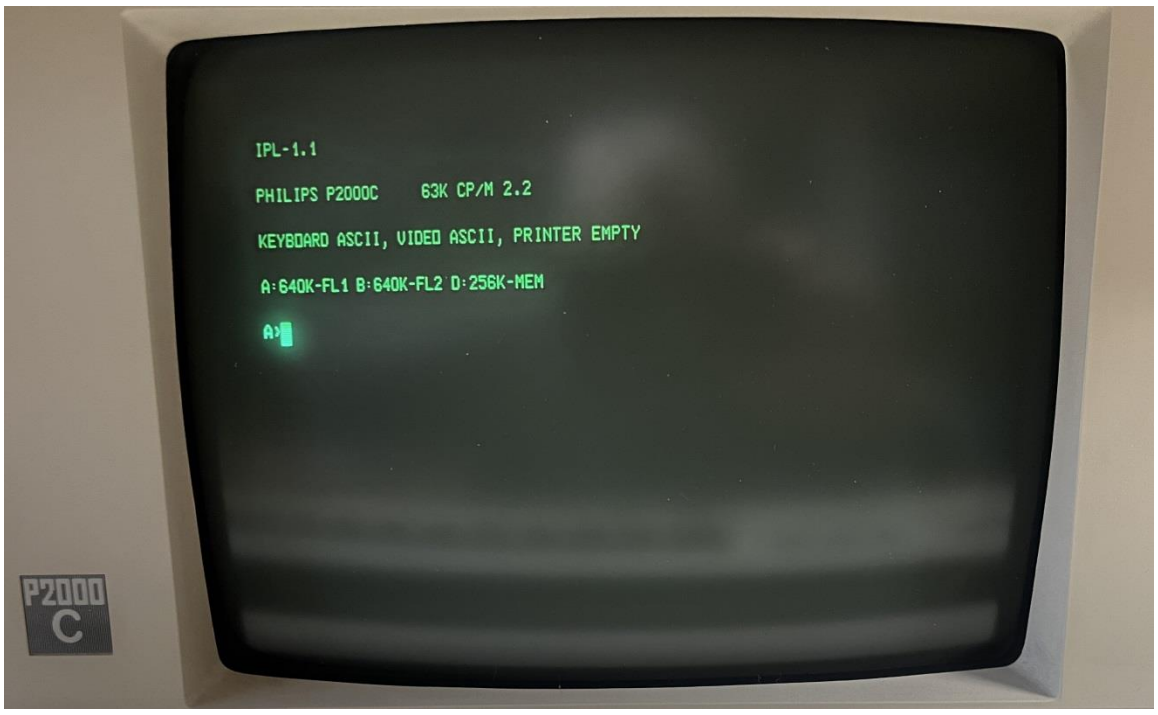

Fig 3. Boot screen with the system disk in first drive A:

2.2. The structure of the Philips P2010-1 (and P2012) floppy disks

The Computarium has a CopyStation for handling vintage disks and disk-images; it is a 486 ISA board with a CGA graphics card, a hard-disk and several drives (3.5", two 5.25") which can be activated or not by a toggle switch or simply by swapping a cable. Software is plain DOS.
The most important feature is the floppy controller: a vintage AT-Type Goldstar which can handle the SD (Single Density) format, often needed for very old CP/M (or 8") disks.
Many different software can be choosen, like Dave Dunfield's ImageDISK, SYBEX 22DISK and ANADISK, and some others.

When I tried to read with the Copystation one of the IMD type disks made by GreaseWeazle everything went well; reading back a disk made on the Philips gave a great number of errors. This means, one has to dive deeper into the structure of the CP/M 2.2 disk format of the P2000C.
CP/M has a Disk Parameter Block DPB (which is a structure hold in the BIOS). to allow the BDOS a correct access to the files. It has many parameters, of which the best known is the sector interleave. The interleave (that can exist in different versions for the physical and logical layout) is written as I =1:1 for an interleave of 1, which means that the different sectors of a track follow in regular sequential order 1,2,3,4…, or for example I = 2:1 if the sequence is 1,3,2,5,…and there is a jump over one sector to reach the next. This has to do with the speed that the CPU and disk controller could do their job: very fast chips did not need a higher interleave, slower ones relied on it to be able to write or read the data correctly on the disk surface.

Let us first show how the vintage 22DISK software from SYBEX represents the disk definition for the Philips computers (only the registered version 1.44 has entries for the P2000C model 2012 and the P3000). The relevant file is CPMDISKS.DEF; it is indexed but can be stripped of its index to allow editing with the STRIPIDX.EXE program.

The interleave is given in a line called SIDE1 0 (head 0) or SIDE2 1 (head 1). For the old 160KB original model 2010 (which has 40 tracks at SSDD with 16 sectors à 256 bytes per track)) this line reads:

SIDE1 0 1, 3, 5, 7, 9,11,15, 2, 4, 6, 8,10,12,14,16

To go from sector 1 to sector 2 one has to make **7** jumps from 1 to 3 to 5 to 7 to 9 to 11 to 15 to 2, so the interleave is **7** and will be given as **I = 7:1** (one also could count the sectors skipped, here 6, and add 1 to that number).
The registered version 1.44 of 22DISK (which has a larger list of disk definitions than the shareware version) has the following two lines for the Philips 2000C type 2012 (which is more or less identical to the Computarium's type 2010-1),

having TEAC FD55-F 80 track DSDD drives, with also 16 sectors of 256 bytes per track.

SIDE 1 0 1,5, 9,13, 2, 6,10,14, 3, 7,11,15, 4, 8,12,16
SIDE 2 1 1,5, 9,13, 2, 6,10,14, 3, 7,11,15, 4, 8,12,16

One can see that here the interleave for both identical sides layout is **4** (4 jumps to go from 1 to 2**, I = 4:1**), and that the first sector id is 1 on both sides.
The complete definition reads as follows:

BEGIN PHI2 Philips 2012   - DSDD 96 tpi 5.25"
DENSITY MFM, LOW
CYLINDERS 80
SIDES 2
SECTORS 16,256
SIDE 1 0 1, 5, 9,13, 2, 6,10,14, 3, 7,11,15, 4, 8,12,16
SIDE 2 1 1, 5, 9,13, 2, 6,10,14, 3, 7,11,15, 4, 8,12,16
BSH 5 BLM 31 EXM 3 DSM 157 DRM 127 AL0 80H AL1 0 OFS 2

Table 1: Disk definition in 22DISK version 1.44

The rather complicated last line has 8 parameters of which DRM is the maximum of possible directory entries (= 128, as count starts with 0). OFS 2 says that the OS takes the first 2 tracks (track 0 and track 1) at the outer rim of the disk.
Each one of these parameters is important, and they vary heavily from model to model; the world of CP/M is a far cry away from uniform standardisation!
Page 17 of the 22DISK manual from SYBEX has a short explanation of these 8 DPB parameters. A more complete discussion of the Disk Parameter Block (which is hold in the BIOS and not on the disk itself!) can be found in **[ref.7]**.

The SYBEX 22DISK manual also lists a small BASIC program to read out the DPB (**[ref.6**] gives various links). It runs under MBASIC (not ABASIC) on the Philips (see listing on Addendum 1) and says that the sectors are interleaved. The list is displayed one number per line but corresponds to this:

1, 2, 5, 6, 9,10,13,14,17,18,21,22,25,26,29,30,
3, 4, 7, 8,11,12,15,16,19,20,23,24,27,28,31,32

The following remarks are important:
- the sequence starts with 1, so what is often called SECTOR ID is here 1 (and not 0, as for instance in the Greaseweazle acorn.adfs.640 definition.)

- the numbers go up to 32, but there are only 16 sectors per track. This comes from the fact that traditional CP/M sectors have only 128 bytes, so this counting is often retained even if the number of bytes per sector is a multiple of 128.

The following table shows how to compute the "real" sector-numbers per side for our P2010-1 (or a P2012):

Sectors in sequential order	DPB as shown by BASIC	SIDE1 0	SIDE2 1
1	1, 2	1	1
2	5,6	5	5
3	9,10	9	9
4	13,14	13	13
5	17,18	2 =17-16+1	2
6	21,22	6 = 21-16+1	6
7	25,26	10 = 25-16+1	10
8	29,30	14 = 29-16+1	14
9	3,4	3	3
10	7,8	7	7
11	11,12	11	11
12	15,16	15	15
13	19,20	4 = 19-16+1	4
14	23,24	8 = 23-16+1	8
15	27,28	12 = 27-16+1	12
16	31,32	16 = 31-16+1	16

Table 2: Sectors sequence on the P2012 disk

So we find the same interleave **I = 4:1** as that given in the two lines of the disk definition entry of 22DISK.

A second clue can be found in **Ivo FILOT's** Github project on the transfer of binary programs between a PC and the P2000C **[ref. 7].** In the last part of his project I.Filot writes about imaging the disks, and gives the following disk definition table for the CPMTOOLS software he uses:

```
# P2000C P2012 disk format
# 5.25" disk DS/DD
diskdef p2000c
  seclen 128
  tracks 80
  sectrk 32
  blocksize 4096
  maxdir 128
  boottrk 2
  skewtab 0 1 4 5 8 9 12 13 16 17 20 21 24 25 28 29 2 3 6 7 10 11 14 15 18 19 22 23 26 27 30 31
  os 2.2
end
```

Table 3: Sectors sequence as given by cpmtools

One can see that the sequence given in the skewtab line is the same as that displayed by the DPB Basic program, with the difference that counting starts at 0 instead of 1. The *maxdir* also shows 128 whereas the DPB program says DRM 127. The *sectr 32* number corresponds to the traditional CP/M counting with 128 bytes/sector, even if in reality there are 256 bytes/sector on the disk.
The blocksize 4096 derives from the BSH=5 and BLM=31 values (see **[ref.7]**)

To make matters worse is that usually CP/M uses an interleave of 1 (**I = 1:1**) in the first two tracks at the outer rim holding the system (written for instance by the SYSGEN command), and starts the higher interleaving only from the third track on up to the last one near the central hole.

All that information can be rather confusing and allows for much potential problems in the down to earth imaging jobs!

2.3. The results of practical tests

2.3.1. Greaseweazle

For the **Greaseweazle** disk definition, the following **diskdefs_philips.cfg** entry makes disks from .IMD images that are bootable and readable on the P2010-1 Philips computer, and also readable on the CopyStation using ImageDisk IMD118:

```
# prefix philips
disk 2012.640
   cyls = 80
    heads = 2
    tracks * ibm.mfm
          secs = 16
          bps = 256
          gap3 = 57
          id = 1
          rate = 250
          interleave = 1
     end
end
```

The gap3 = 57 and interleave=1 do not seem to be mandatory (as I = 1:1 is the default). What is important is the (sector) id =1 line; using id =0 makes for a disk that is not readable on the Copystation with IMD.

2.3.2. The CopyStation with ImageDisk and Sybex tools, and Greaseweazle compatibility

The Copystation has been fitted for the Philips tests with a drive from an ACORN BBC Master; sold by Technomatic (which made the enclosure with power-supply for a dual disk-combo, the 5.25" DSDD 80 track drive is a Mitsubishi MF530A).

Fig.4. The Mitsubishi drive MF530A-312M

Here follow the steps of the different disk tests:

1. Erase 5 disks labeled A, B, C … on a SHUGART SA465-3AA drive with the Greaseweazle to a virgin state, using ½ step size (work done at home).

2. On the Philips 2000C, with a system disk in A: format the first disk A (in the 2nd drive) and then copy A: to this disk, using the UTIL.COM software. Check that this copy is bootable, shows a correct directory, and that the programs on it execute without any fault.

Run the DPB.BAS program with MBASIC; here a screenshot of the results:

3. Now the disk A (made in the Philips B: drive) is taken to the Copystation to be read with ImageDISK version 1.18 into the file TESTA.IMD. The settings of IMD are such that nearly all parameters are left in the default "Calculated" for the gaps and "as read" for the interleave (see fig.6). Fig 7 shows a screenshot of that reading procedure (that went smoothly without any errors)

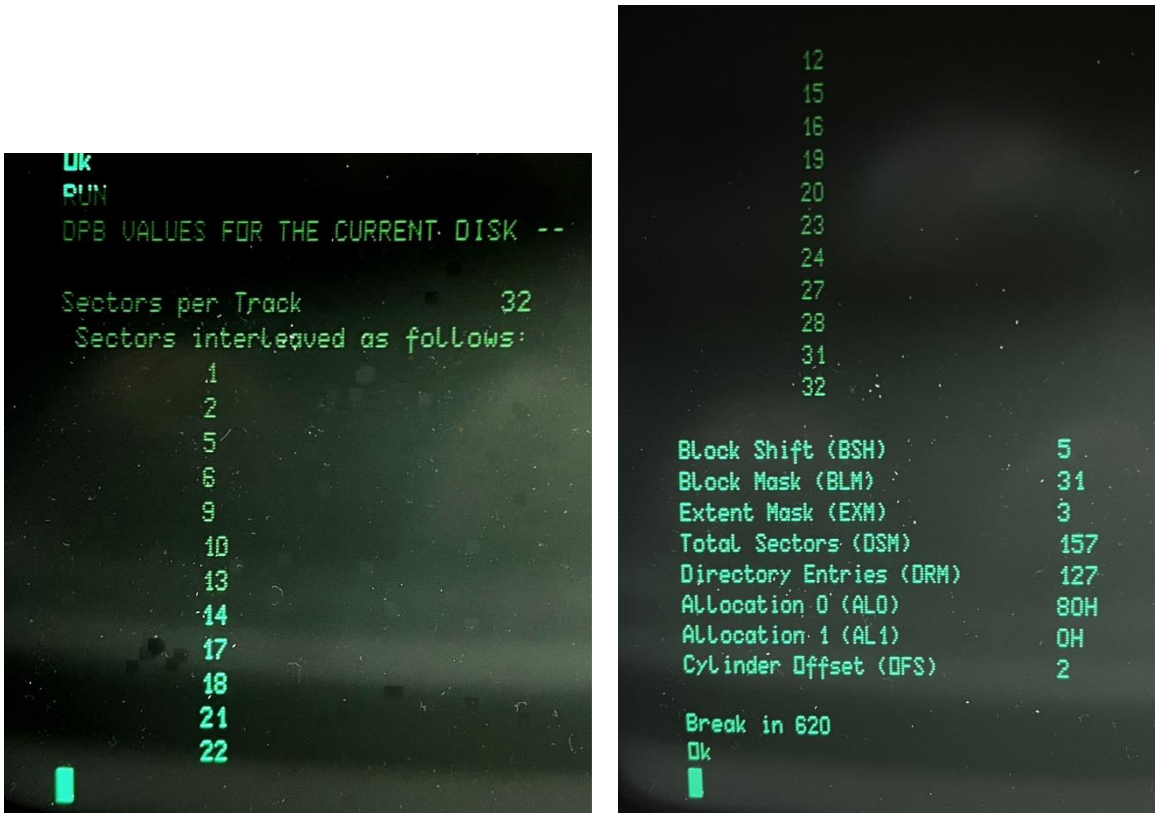

Fig. 5. Screenshot of the DPB.BAS results. The number of 32 means 32 sectors of 128 bytes (=standard CP/M counting), which here correspond to 16 sectors of 256 bytes per track on each side. See table 2. ***The important result is that the sectors are interleaved (with I = 4:1) as shown in table 2.***

.

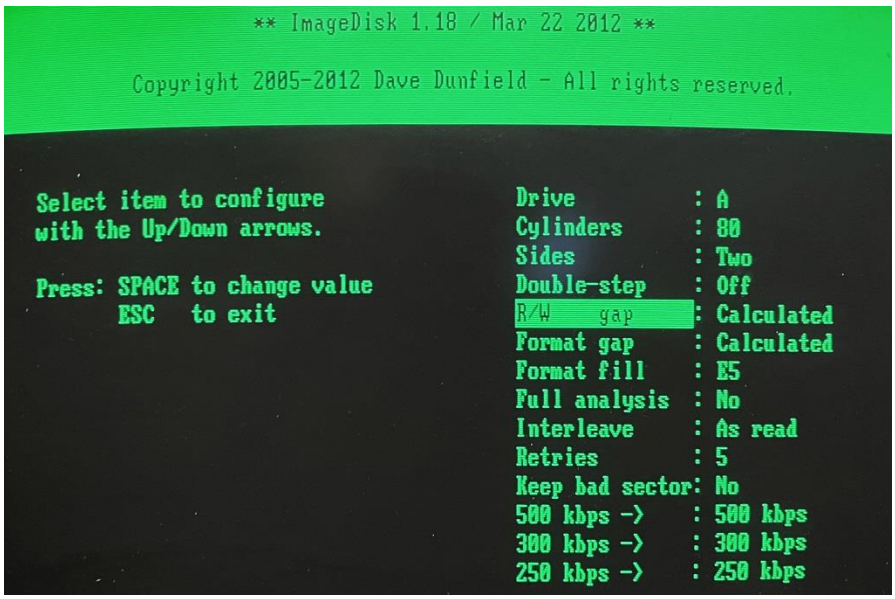

Fig 6. Settings of IMD for reading the Philips-made disk A

Fig. 7. IMD reads disk A (which according to the DPB should have an interleave 4:1) in the normal sequence 1,2,3,… corresponding to an interleave 1:1 (= no interleave).

5. The next step is to write the TESTA.IMD file back on the virgin disk C, without changing the IMD settings. Figure 8 shows this write procedure going on:

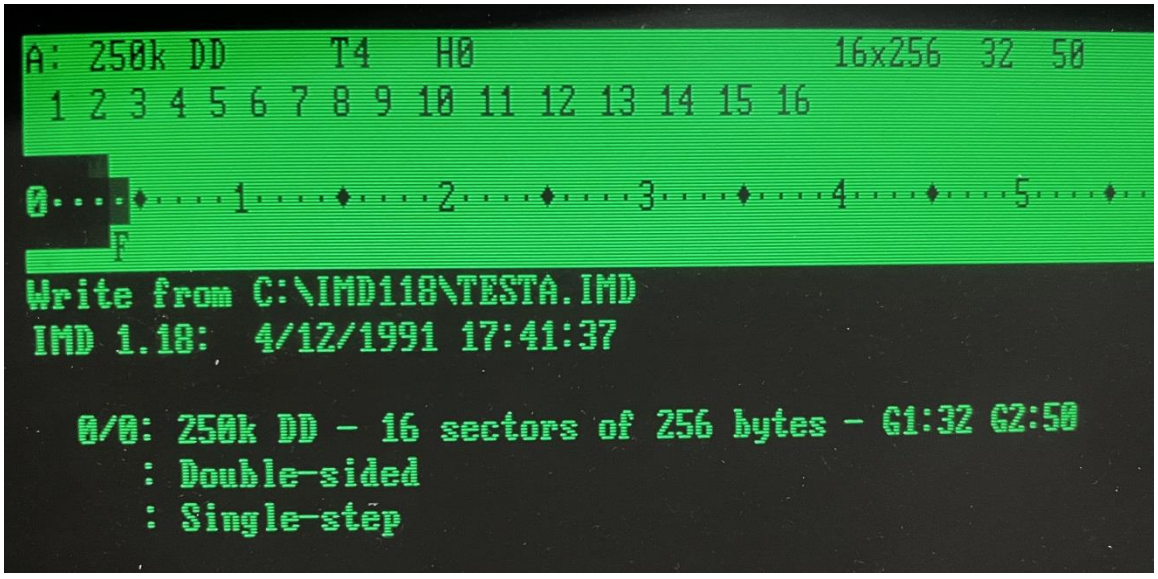

Fig. 8. Test-file TESTA.IMD is written to a new virgin disk C; note that IMD does this in the sequential sector order, i.e. with I = 1:1.
The gaps G1 and G2 are the same as those given in the read screen (fig. 7)

6. Disk C is put into the Philips P2000C first drive: it boots without any problems, DIR is ok as is the running of different software on it.

Conclusion from steps 1 to 6:

A disk made in the Philips P2000C (type 2010-1 and probably also valid for type 2012) with an interleave I = 4:1 can be read by ImageDISK with an interleave setting "as read"; a virgin disk can be written with the same settings and is bootable and fully functional on the Philips.
A second test sequence using "no interleave" (i.e. setting: interleave =1 which is equivalent to I 1:1) gives the same results.

Now follows a sequence of tests with the ImageDISK Interleave set to 4.

7. Read the first Philips made disk A with I =4 into TESTD.IMD, all other settings remaining the same. The procedure terminates without any errors, but is a bit slower than that made in step 2.
Fig.8 shows this reading. The right upper corner displays the interleave 4:1, and the reading sequence is the same as that given in table2 (and by the DPB.BAS program).

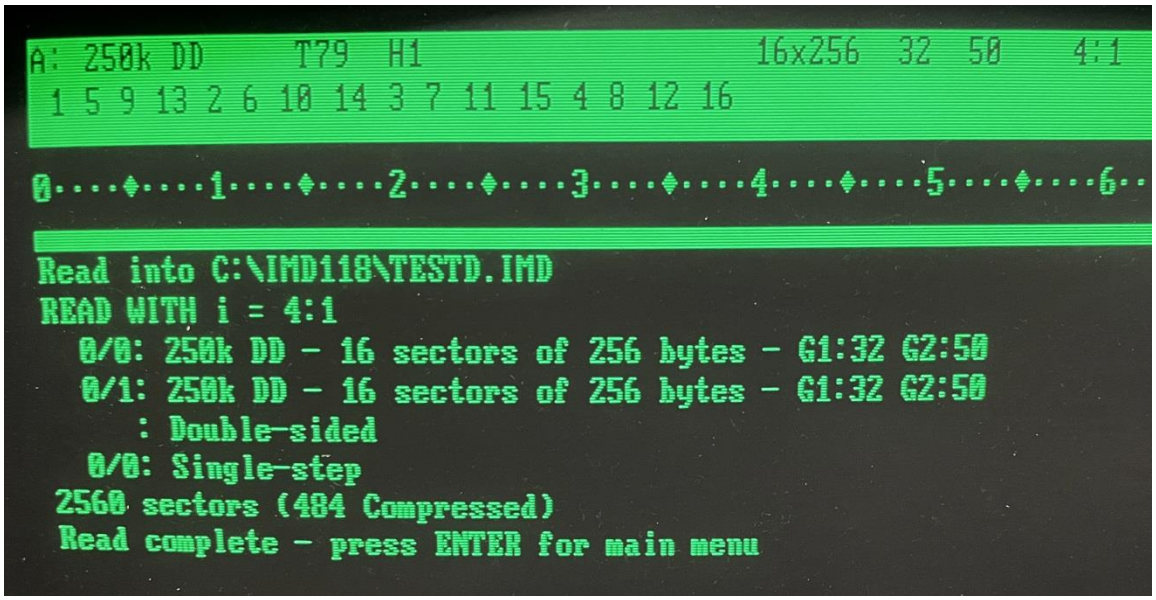

Fig.8. A disk made on the Philips is read into file TESTD.IMD with an interleave of 4. Note the shown sequence of sectors and compare to table 2.

8. Write the file TESTD.IMD image back to a virgin disk D, all settings remaining the same.

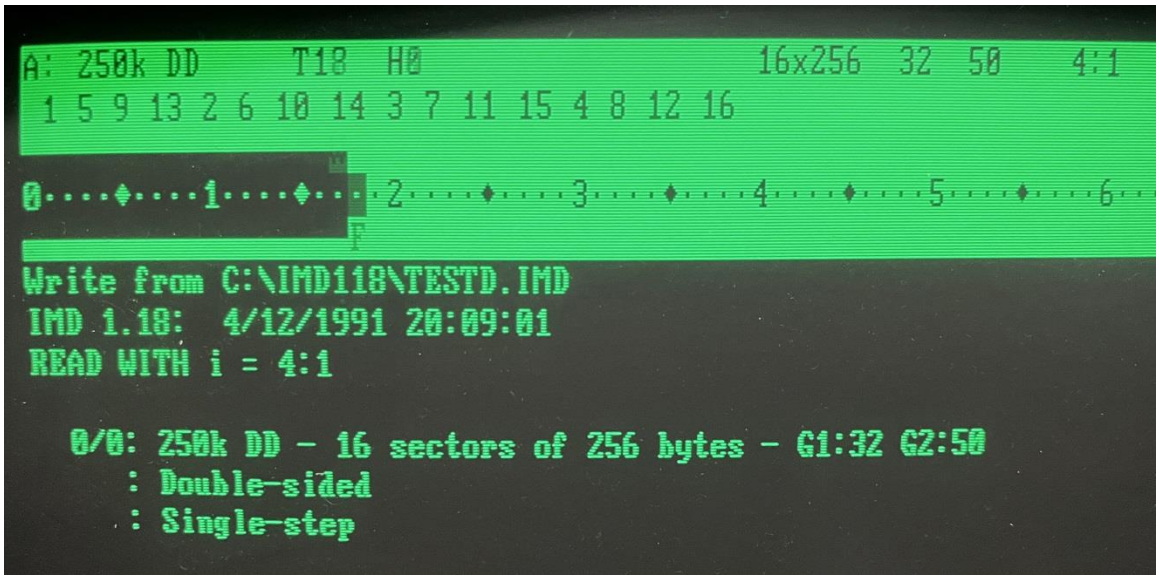

Fig.9. The file TESTB.IMD is written to a virgin disk D with the same settings (interleave 4:1); the displayed writing sequence of the sectors is the same as that used for reading.

9. This written disk D is put into the Philips second drive: the directory is correctly shown, programs on it can be executed**, but it is not bootable** on either drive. Trying to make a SYSGEN terminates without any error messages but the disk remains unbootable.

Conclusion from steps 7 to 9:

A disk made in the Philips P2000C (type 2010-1 and probably also valid for type 2012) with an interleave I = 4:1 corresponding to the DPB can be read by ImageDISK with an interleave set at 4 (I = 4:1); a virgin disk can be written with the same settings **but it is not bootable** (nor SYSGEN'able) . Except this it is fully functional on the Philips. This problem probably comes from the fact hat the first 2 system tracks have a interleave of 4 instead of 1.

10. The Sybex software **22DISK** can not make images, but it can format a disk as CP/M (CFORMAT.COM) Doing this on the Copystation with the disk definition given in table 1 makes a blank disk that can be written on the P2010-1 and also made bootable with SYSGEN. So probably 22DISK automatically writes the first 2 blank sectors with an interleave of 1.

Using the ANADISK program (also from SYBEX) to SCAN a disk made on the Philips starts with many read errors and wildly varying interleaves, but after having read a lot of tracks it settles down showing an interleave of 1:1.

.

The final test sequence checks if the disks made on the Philips and the Copystation are readable on the Greaseweazle.

The disk definition shown on page 10 is used (it is not part of the default GW list). The disks will be read into an .IMD image.

11.

Read disk A (= original Philips-made):	no errors
Read disk C (= written by IMD with interleave as read or I = 1:1 settings):	no errors
Read disk D (= written by IMD with interleave = 4):	no errors

Overall conclusion for handling Philips 2012 (or 2010-1) disks and .IMD disk-images on the Greaseweazle and on a DOS computer running ImageDISK:

For clarity use an explicitely given interleave setting of 1 (I = 1:1), even if that is the implied Greaseweazle default and the IMD default of Interleave "as read" is also ok.

3. The undispensable Philips utilities.

Philips has added to its CP/M 2.2. two very important programs.

CONFIG.COM is a configuration program to define the keyboard layout and the bit-rates of the two serial female DB25 COM sockets (only the bit rates, no other parameters like parity, stop bits etc…). The right-most socket (when seen from behind) is the connector for a serial printer (there is no provision for a parallel printer), and the one with a double-flash symbol is the classical COM connector, but in a female configuration. The TX and RX signals are by default on pins 2 and 3, but an internal jumper board can flip this order. On the Computarium's machine, we have the default. The SIO (channel A) driving this socket can be configured for asynchronous and synchronous operation, so the pin-out is different from a "standard" COM port. Pins 15, 17 and 24 are related to clock outputs and inputs and must be tied together for asynchronous operation to work. For a "normal" operation, a 3-wire cable is all what's needed (assuming 8 bit, no parity, 1 stop bit no flow control). As I connected a standard PC running Teraterm, the cable from the male DB25 plug at the Philips to the female DB9 plug at the PC has the following wiring:

pin 2 to ----------	pin 2
pin 3 to ----------	pin 3
pin 7 to ----------	pin 5 (GND)
pins 15-17-24 tied together	pins 7-8 (RTS-CTS) tied together

The configuration program also allows to define a program executed at startup, and to display a one-line welcome message.

UTIL.COM is a program to format floppy disks in 3 formats: 160kB for the P2010, 640kB for the 2012 (and 2010-1) and another for the 2000T. We will use exclusively the 640 kB format. After formatting, the disk is not yet bootable, so a SYSGEN must be made to transfer the system. UTIL.COM also has an option to copy disk A: to B: (including the system).

On one of the disk images of the P2000-x.IMD collection of Dave Dunfield there was a program called **COMINP.COM**. It is a utility to accept text files coming from the outside through the COM port into the P2000C. It works at the speed set in the configuration, without any flow-control nor parity. When a text file is sent from the PC by Teraterm, it is accepted by COMINP; when everything is sent typing CTRL-C on the Philips closes the transmission and puts the file with its EOF (=CTRL-Z) on the floppy disk (there is no need to type a CTRL-Z code on the PC side).

4. CP/M emulators running on the PC

A lot of preliminary work or testing can be done in a CP/M emulator on the PC. I use two of these open source projects (on a WIN11 Pro machine):

iz-cmp is a very short, portable "CP/M environment" developed by Ivan Izag; see [**ref.8]** for a link to his GitHub project page.
All files put into the folder containing iz-cpm.exe appear as files in disk drive A:
The usual CP/M utilities and tools are present: ASM, LOAD, DDT, ED ….or can easily be included by simply copying rhe .COM file into that folder. Ivan Izag warns that his project (developed with RUST) is an educational project, so not all CP/M bells and whistles are there.

RunCPM is a larger multi-platform system by MockbatheBorg (real name unknown); see his GitHub page here **[ref.9**]. A very nice feature is that you can add drives by simply including a folder with sub-folder 0 in the main directory. By default a folder A with subfolder 0 is there: that is the drive A0: for User 0.
Just by creating a folder B with sub-folder 0 adds a second drive B0:
If a utility does not run in iz-cpm. one should try it on RunCPM.
The Windows binaries can be found in Sourceforge at
https://sourceforge.net/projects/runcpm/files/

Both programs run in a command window; there is no need for a GUI.

I used these with the UNLOAD.COM utility to translate binary .COM files into Intel-Hex files (.HEX), that could be sent into the Philips by a plain text transfer.

5. Transfering data into and out of the P2000C

The work related in this chapter has for a good part done together with Computarium Crew member (and friend) **Claude Baumann**.

5.1. Plain text transfer

The Philips uitility COMINP.COM mentioned above is all what's needed for sending a text file into the Philips (but not the other way round!). I found that speeds up to 9600 bps are possible, under the condition that one includes a delay after each character and end of line sent. In **Teraterm**, values that work good are 10ms delay for the characters, and 100-200ms for each line. If the incoming text displayed on the Philips screen becomes corrupted, the transmission speed should be lowered to 2400 or even 1200 bps

Using this type of transfer allows sending decently long .ASM or .HEX files to the Philips, where thay can be assembled and loaded into an executable .COM file.

The Philips CP/M disks did not include the MLOAD.COM (a more powerful variant of LOAD.COM), so I used an MLOAD.HEX made by UNLOAD in an emulator to create the MLOAD.COM on the Philips with LOAD MLOAD.HEX.

As told above, COMINP is a simple Text-Input transfer program. Philips developed also a set of two utilities called TTYCONF.COM and TTY.COM.

I entered into contact with the very knowledgeable people of the German **Verein zum Erhalt klassischer Computer e.V**. (website: www.classic-computing.de), and member Mathias sent me the TTY.COM and TTYCONF.COM files (the TTY User Guide can easily found on the Internet, but not the .COM or .ASM files). After the usual UNLOAD – text transfer – MLOAD procedure, both utilities were running on the P2000C.
The configuration program TTYCONF allows a more diverse setting as CONFIG; all the usual serial parameters like number of bits and stop bits, parity, flow-control etc. can be defined. I found that in reality TTY only works at the "standard" 9600 bps..
TTY.COM allows to receive or to send text files, so we have here a tool for a complete control over exchanging text files. As usually files should not be too long, and one has to experiment with the delay settings in Teraterm to get a reasonably correct transmission.

5.2. Transfer of binary data

Binary data must be exchanged by a transmission protocol; XMODEM was one of the earliest used. Mike Douglas, the owner of the website deramp.com, has a couple of variants called PCGET and PCPUT for different CP/M systems, among them Kaypro 4 which as the Philips uses a Z80 uP and its serial chip SIO. The adresses of the control and data channels vary with company, and we were unable to get one running (with Teraterm on the PC using XMODEM for sending or receiving). Neither could we put the .HEX files of two similar programs written by Mathias and called P2000GET and P2000PUT into working .COM files.

So we looked for another solution. As told above, Ivo FILOT (an assistant professor of the Dutch Technical University of Eindhoven) has several projects on GitHub, among them one called **TRANSFER.ASM** for receiving binary data on the Philips, and a Python program **sendcom.py** running on the PC for sending the data. The protocol is a proprietary version of his own (not pure XMODEM). We transfered the text file TRANSFER.ASM using TTY, and assembled it with ASM without any problem on the Philips. MLOAD then produced the executable TRANSFER.COM, and we were able to send P2000GET.COM from the PC to the Philips. This P2000GET.COM worked fine, and we could send via Teraterm/Xmodem the missing P2000PUT.COM.

So we now have two solutions for handling **binary** transfers:

- the TRANSFER/sendcom.py combo to send data into the P2000C

- the P2000GET & PC2000PUT duo for transfers in both directions.

What's missing is a **KERMIT** running on the P2000C.

Kermit was the solution of choice in the old days, and I used it on many machines. I could not find a KERMIT.COM specially configured for the Philips, except a version given by I Filot in his Github page that aborts with a LOAD error.

It is possible to build a generic CP/M Kermit from two packages found on the website https://www.kermitproject.org/archive.html under CP/M-80.

Unzipping the cpm80.zip file yields a lot of .asm and .hex files, among them 2 for creating a generic Kermit**: cpsker.hex** and **cpvgen.hex**. Both can be assembled into a .COM file by LOAD CPSKER, CPVGEN, but the resulting KERMIT.COM does not work in a correct manner (only Kermit Connect works partially).
So this is a problem still waiting to be solved.

6. Overal conclusion

This work on the Philips P2000C (type 2010-1) was very exciting, even if it took many hours. We now have a more transparent view of how to handle the Philips disk images via Greaseweazle or ImageDISK.

The problem of the transfer of binary files is solved with the XMODEM tools P2000GET/PUT.

The vintage CP/M 2.2 is a quite respectable beast, and many aspects of its working remain in the dark for me.

If a reader finds errors in this paper, I am grateful for all remarks sent fo mailto:francis.massen@education.lu.

References:

1. Centre for Computing History
https://www.computinghistory.org.uk/det/7130/Philips-P2000C/

2. Dave's Old Computers – System and Install Disks
http://dunfield.classiccmp.org/img42841/system.htm

3. Electrickery.NL
https://electrickery.nl/comp/p2000c/floppyImages/

4. Keir Fraser Greaseweazle GitHub project
https://github.com/keirf/greaseweazle

5. A vendor of the Greaseweazle board
https://www.retropassion.co.uk/product/greaseweazle/

6. DPB.program from the 22DISK SYBEX manual, page 17
DPB.BAS (MBASIC); see listing in Addendum
DPB.ASM
DPB.COM.

7. The Disk Parameter Block (in CP/M)
https://www.idealine.info/sharpmz/dpb.htm

8. Ivo Filot: P2000C CPM Transfer project on GitHub
https://github.com/ifilot/p2000c-cpm-transfer

9. Ivan Izag iz-cpm CP/M Environment
https://github.com/ivanizag/iz-cpm

10. RunCPM
https://github.com/MockbaTheBorg/RunCPM

Addendum : Listing of the DPB BASIC program (22DISK manual)

```
100 REM PROGRAM TO DISPLAY A DEFAULT DPB
110 REM
120 REM FOR CP/M 2.2 ONLY
130 REM
140 DEF FNP(X) = PEEK(X) + 256*PEEK(X+1)
150 IF PEEK(0) <> 195 THEN PRINT "Requires CP/M" : STOP
160 PRINT "DPB VALUES FOR THE CURRENT DISK -- " : PRINT
170 REM
180 REM STUFF S$ WITH THE FOLLOWING PROGRAM -
190 REM
200 REM PUSH H           ; Save address of FAC-3
210 REM LHLD 1           ; Get BIOS Cold boot entry
220 REM MVI L,27         ; Change to SELDISK vector
230 REM LDA 4            ; Get current disk
240 REM MOV C,A
250 REM MVI A,PCHL       ; for indirect call
260 REM STA 38H          ; set in RST 7 vector
270 REM RST 7            ; call SELDISK
280 REM POP D            ; FAC-3
290 REM XCHG
300 REM MOV M,E          ; Store pointer to DPH
310 REM INX H
320 REM MOV M,D
330 REM RET              ; exit...
340 REM
350 S$ = ""
359 REM Read the USR code
360 READ SD : IF SD < 0 THEN 380
370 S$ = S$ + CHR$(SD) : GOTO 360
380 SX = VARPTR(S$) +1
390 SR = FNP(SX)
299 REM  User program set up
400 DEF USR = SR
409 REM Get DPH Pointer
410 DPH = USR(0)
420 IF DPH = 0 THEN PRINT "Select error" : STOP
429 REM Get Sectors/Track
430 SPT = FNP( FNP( DPH +10))
440 PRINT "Sectors per Track";TAB(30);SPT
449 REM  if no interleave
450 IF FNP(DPH) = 0 THEN 510
460 PRINT " Sectors interleaved as follows:"
470 FOR I = 1 TO SPT
480  PRINT TAB(10); PEEK( FNP(DPH) + I-1)
490 NEXT I
500 PRINT : GOTO 520
510 PRINT " Sectors are not interleaved"
520 DPB = FNP(DPH+10)
530 PRINT "Block Shift (BSH)";TAB(30);PEEK(DPB+2)
540 PRINT "Block Mask (BLM)";TAB(30);PEEK(DPB+3)
550 PRINT "Extent Mask (EXM)";TAB(30);PEEK(DPB+4)
560 PRINT "Total Sectors (DSM)";TAB(30);FNP(DPB+5)
570 PRINT "Directory Entries (DRM)";TAB(30);FNP(DPB+7)
580 PRINT "Allocation 0 (AL0)";TAB(31);HEX$(PEEK(DPB+9));"H"
590 PRINT "Allocation 1 (AL1)";TAB(31);HEX$(PEEK(DPB+10));"H"
600 PRINT "Cylinder Offset (OFS)";TAB(30);PEEK(DPB+13)
610 PRINT
620 STOP
630 DATA  229,42,1,0,46,27,58,4,0,79,62,233,50,56,0
640 DATA  255,209,235,115,35,114,201,-1
650 END
```