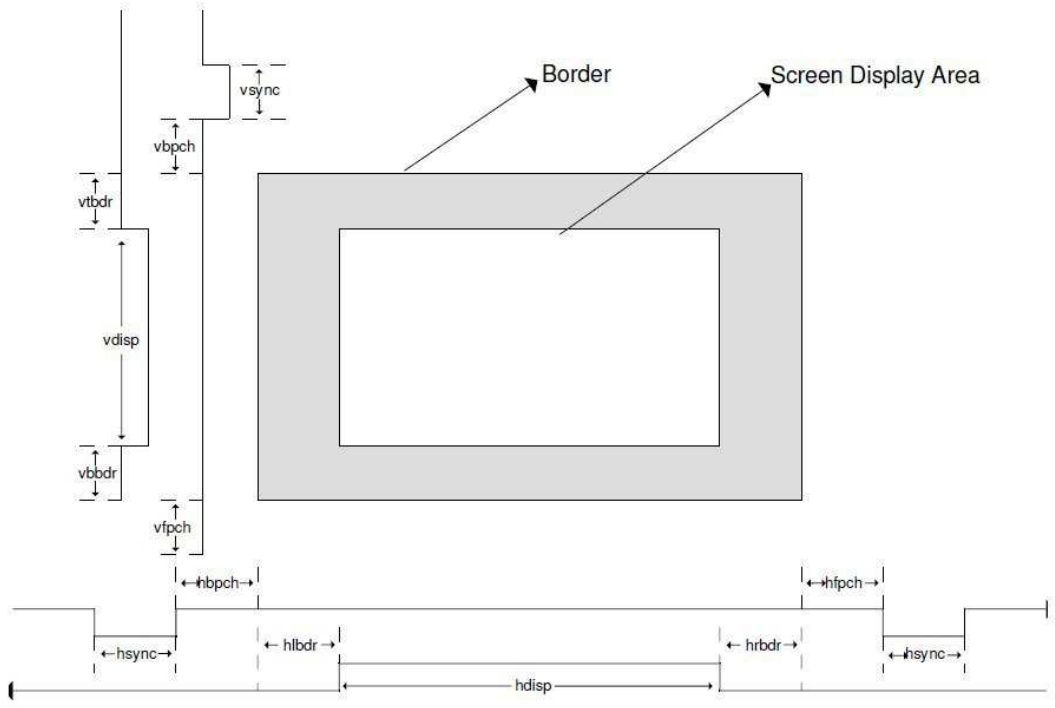
From CRT to LCD
by Gerald Fitton

"Out with the old and in with the new"
Traditional expression - origin unknown

Early Monitors used Cathode Ray Tubes. It was about 30 years ago that these weighty monitors began to be replaced by flat screen technology such as the LCD and LED displays. The Monitor Definition Files used with RISC OS hardware such as the RiscPC were designed for CRT monitors. Monitor Definition Files used with flat screens are much easier to create.

CRT Parameters

The diagram shows the 12 parameters which are required for a Monitor Definition File suitable for a CRT monitor. Six of these are required for the horizontal raster and 6 for the vertical raster.



Monitor Definition Files

The RiscPC and subsequent machines do not have screen modes built into the Operating System. Instead, screen modes are defined in an MDF (Monitor Definition File) which is loaded when the machine boots up. The concept of an MDF was introduced so that the RiscPC user choose from a much wider range of CRT monitors with screen modes individually tailored to to suit each monitor. Because of this history of use with CRT monitors the current MDFs retain the 12 parameters even though they are unnecessary.

Where are the MDFs stored?

Inside the directory !Boot.Resources.Configure Monitors you will find sub-directories such as Acorn and LG. Each of these sub-directories will contain one or more Mode Files with titles such as AKF92 (an Acorn monitor).

What does an MDF look like?

All the Mode Files are plain text files which can be edited with a text editor. The Mode File for the AKF92 (CRT) monitor contains about three dozen MDFs. Each MDF defines a Mode which can be displayed with that particular monitor. As examples, here are two MDFs taken from the AKF92 Mode File.

```
# 240 x 352 (60Hz)
startmode
mode_name:
x_res:240
y_res:352
pixel_rate:8400
h_timings:20,16,20,240,8,8
v_timings:2,58,0,352,0,37
sync_pol:2
endmode
```

... and ...

```
# 1280 x 1024 (54Hz)
startmode
mode_name:1280 x 1024
x_res:1280
y_res:1024
pixel_rate:110000
h_timings:166,90,140,1280,30,30
v_timings:3,32,50,1024,50,4
sync_pol:0
endmode
```

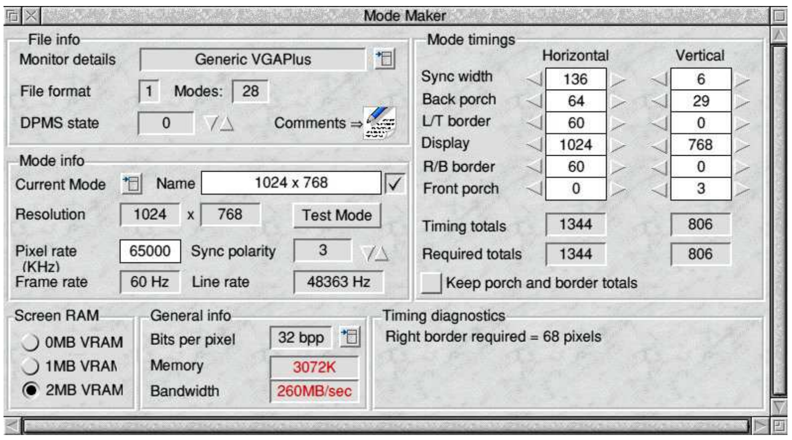
The format of an MDF is:

```
startmode
mode_name: mode_name
x_res:      x-resolution
y_res:      y-resolution
pixel_rate:  pixel_rate
h_timings:   hsync, hbpch, hlbdr,hdisp,hrbdr,hfpch
v_timings:   vsync, vbpch, vtbdr, vdisp. vbbdr, vfpch
sync_pol:    sync_polarities
endmode
```

where:
mode_name: is a textual name for the mode that will be used in the display manager's mode menu. It is possible to prevent defined screen modes from appearing in the modes menu by simply leaving this field blank. Mode names are limited to 19 characters in length and may contain space characters.
x-resolution: is the number of pixels displayed across the screen
y-resolution: is the number of rasters displayed vertically (pixels)
hsync: is the width of the horizontal sync pulse
hbpch: is the width of the horizontal back porch
hlbdr: is the width of the left border
hdisp: is the number of pixels displayed horizontally (usually the same as the x-resolution)
hrbdr: is the width of the right border
hfpch: is the width of the horizontal front porch
vsync: is the width of the vertical sync pulse
vbpch: is the width of the vertical back porch
vtbdr: is the width of the top border
vdisp: is the number of rasters displayed vertically (pixels)
vbbdr: is the width of the bottom border
vfpch: is the width of the vertical front porch
pixel_rate: is the pixel rate in kHz
sync_polarities: is a number indicating the type of sync signals are required.
The types of sync signal are as follows:
0 hsync normal, vsync normal
1 hsync inverted, vsync normal
2 hsync normal, vsync inverted
3 hsync inverted, vsync inverted
All values on the h_timings line are in units of pixels, and all values on the v_timings line are in units of raster lines.

MDFs for CRT Monitors

If you have a CRT monitor then it is probable that your computer is a RiscPC or some other Acorn hardware machine. Making MDFs for a CRT monitor is a tricky business. The 12 parameters of the MDF (e.g. the synch width) depend on the CRT used. A utility, MakeModes version 0.27a (8th November 2006), which simplifies this process can be found at: http://www.cjemicros.co.uk/micros/resources/index.shtml. It will run in RISC OS 4.02 but not in RISC OS 5 or 6. Here is a screenshot of the utility in action:



File info
Monitor details: Generic VGAPlus
File format: 1 Modes: 28
DPMS state: 0 Comments ⇒

Mode info
Current Mode Name: 1024 x 768 ✓
Resolution: 1024 x 768 Test Mode
Pixel rate (KHz): 65000 Sync polarity: 3
Frame rate: 60 Hz Line rate: 48363 Hz

Mode timings

	Horizontal	Vertical
Sync width	136	6
Back porch	64	29
L/T border	60	0
Display	1024	768
R/B border	60	0
Front porch	0	3
Timing totals	1344	806
Required totals	1344	806

Keep porch and border totals

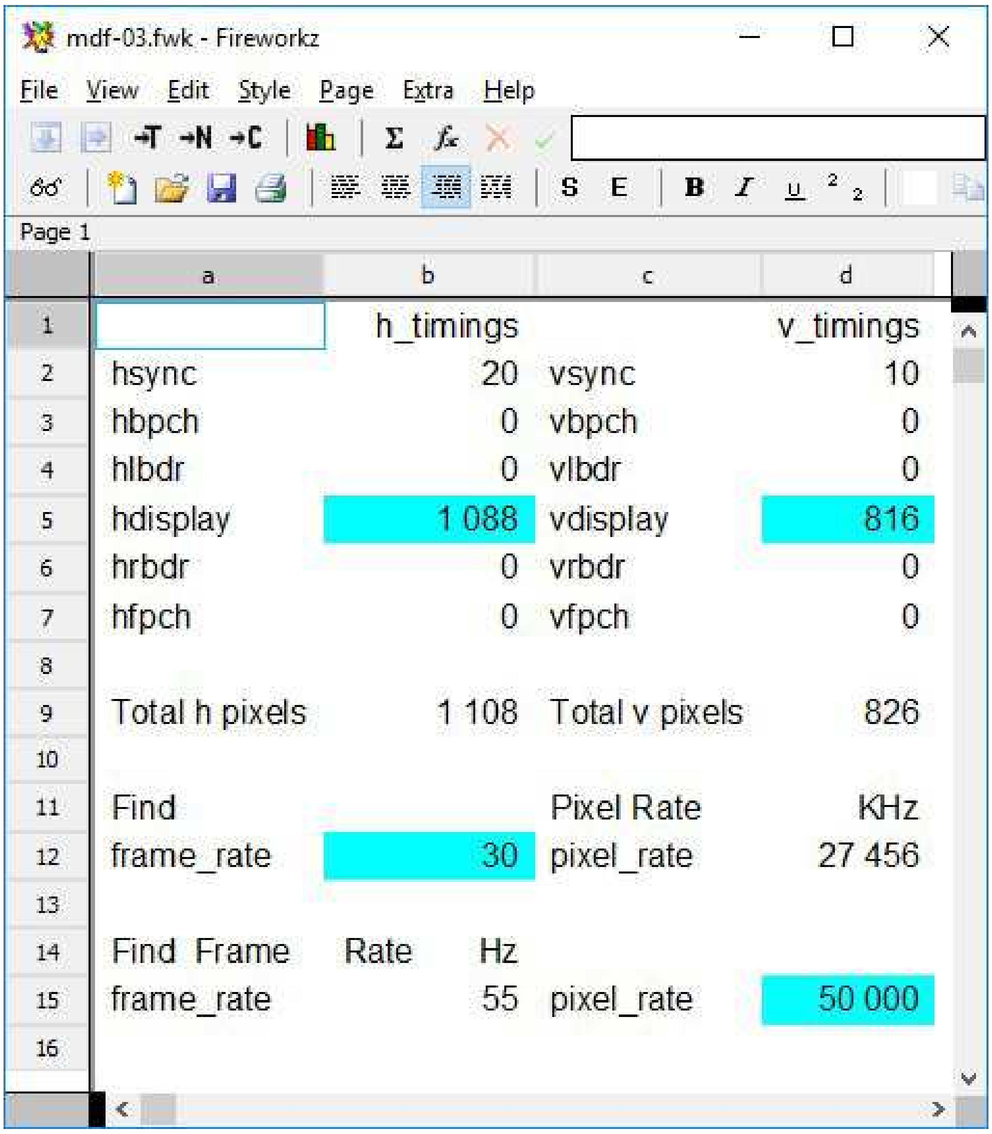
Screen RAM
0MB VRAM
1MB VRAM
● 2MB VRAM

General info
Bits per pixel: 32 bpp
Memory: 3072K
Bandwidth: 260MB/sec

Timing diagnostics
Right border required = 68 pixels

MDFs for Emulators

If you are using an emulator such as Red Squirrel, VRPC or RPCEmu then the calculations necessary in order to create a working MDF can be simplified considerably because the synchronisation parameters are much more flexible. All the synchronisation parameters but the horizontal and vertical resolutions can be the same for a wide range of resolutions. I have experimented with these timings and I have found that with hsync = 20 and vsynch = 10 the MDFs I create always work. Have a look at the Fireworkz spreadsheet (mdf-03.fwk) shown in the screenshot. In use, enter your desired horizontal and vertical resolutions into b5 and d5 respectively then, when you enter your desired frame rate into b12 the pixel rate is returned in d12. Alternatively, you can enter a pixel rate into d15 and a frame rate will be returned in b15. These parameters can be used to create an MDF and, provided certain conditions are met, the MDF will work. The principal conditions are that the horizontal resolution must be divisible by 4 and the vertical resolution divisible by 2.



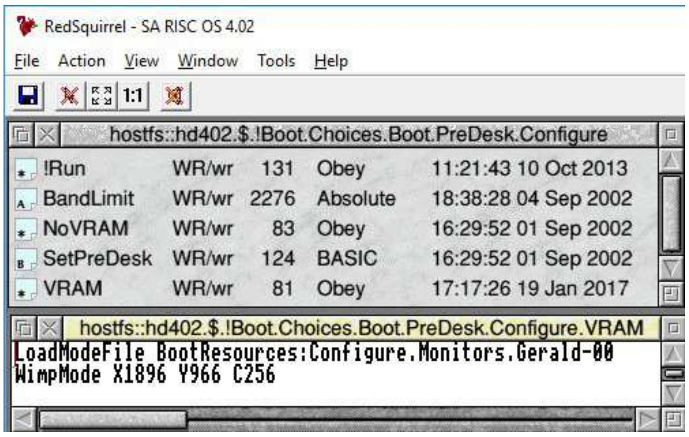
	a	b	c	d
1		h_timings		v_timings
2	hsync	20	vsync	10
3	hbpch	0	vbpch	0
4	hlbdr	0	vlbdr	0
5	hdisplay	1 088	vdisplay	816
6	hrbdr	0	vrbdr	0
7	hfpch	0	vfpch	0
8				
9	Total h pixels	1 108	Total v pixels	826
10				
11	Find		Pixel Rate	KHz
12	frame_rate	30	pixel_rate	27 456
13				
14	Find Frame	Rate	Hz	
15	frame_rate	55	pixel_rate	50 000
16				

There are only two formulae in the spreadsheet. The formula in cell d12 is b9*d9*b12/1000; the formula in b15 is d15/b9/d9*1000. If your favourite spreadsheet is something other than Fireworkz then you can create your own spreadsheet (or do the sums manually) using the parameters and formulae used in this spreadsheet.

Startup Mode

First of all in Red Squirrel.

There is a file at: hostfs::hd402.$.!Boot.Choices.Boot.PreDesk.Configure. The file is called VRAM. The screenshot below will help you locate this file and its contents.



RedSquirrel - SA RISC OS 4.02
File Action View Window Tools Help
hostfs::hd402.$.!Boot.Choices.Boot.PreDesk.Configure

```
!Run        WR/wr   131  Obey      11:21:43 10 Oct 2013
BandLimit   WR/wr  2276  Absolute  18:38:28 04 Sep 2002
NoVRAM      WR/wr    83  Obey      16:29:52 01 Sep 2002
SetPreDesk  WR/wr   124  BASIC     16:29:52 01 Sep 2002
VRAM        WR/wr    81  Obey      17:17:26 19 Jan 2017
```

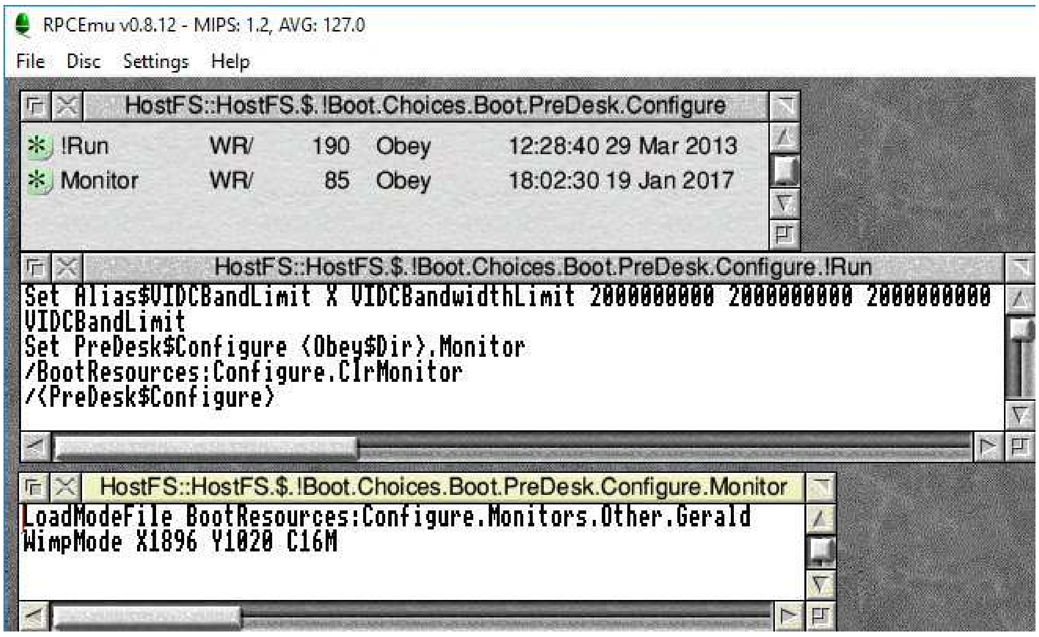
hostfs::hd402.$.!Boot.Choices.Boot.PreDesk.Configure.VRAM

```
LoadModeFile BootResources:Configure.Monitors.Gerald-00
WimpMode X1896 Y966 C256
```

In Red Squirrel I use as my (Monitor) Modefile a file called ‘Gerald-00’ It is located at: $.!Boot.Configure.Resources. It contains my monitor definitions. To boot up with the resolution 1896x966 the VRAM file must contain a path to the (Monitor) Modefile (e.g. Gerald-00) and it must contain at least one MDF for that resolution. You will see that the second line of VRAM contains “WimpMode X1896 Y966 C256”. It is this line which sets the resolution which is used by the Boot sequence.

Now to RPCEmu.

The corresponding file (to the VRAM file) is called ‘Monitor’ and is an Obey file. Monitor is called up by a !Run file within the Configure directory. The !Run file overcomes the 2MB VRAM limitation by forcing the VIDC bandwidth to a much higher value than is possible with 2MB VRAM. You’ll see that I use 1896x1020 with 16 million colours as the mode into which I boot up.



RPCEmu v0.8.12 - MIPS: 1.2, AVG: 127.0
File Disc Settings Help
HostFS::HostFS.$.!Boot.Choices.Boot.PreDesk.Configure

```
!Run      WR/   190  Obey   12:28:40 29 Mar 2013
Monitor   WR/    85  Obey   18:02:30 19 Jan 2017
```

HostFS::HostFS.$.!Boot.Choices.Boot.PreDesk.Configure.!Run

```
Set Alias$VIDCBandLimit X VIDCBandwidthLimit 2000000000 2000000000 2000000000
VIDCBandLimit
Set PreDesk$Configure <Obey$Dir>.Monitor
/BootResources:Configure.ClrMonitor
/<PreDesk$Configure>
```

HostFS::HostFS.$.!Boot.Choices.Boot.PreDesk.Configure.Monitor

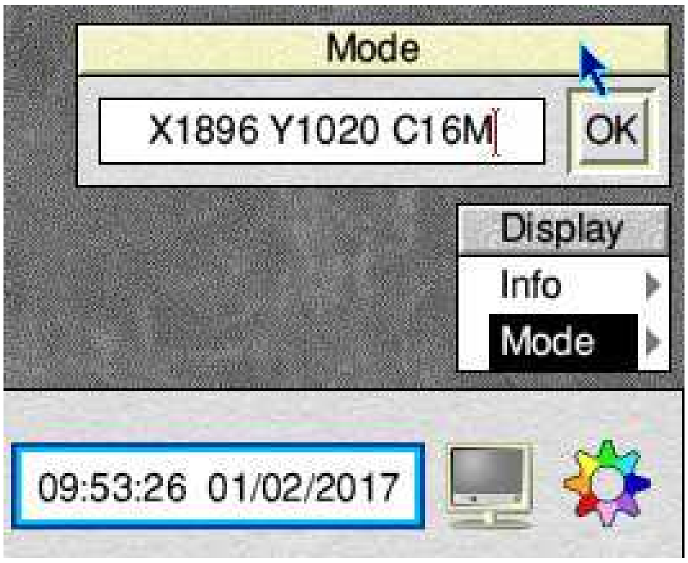
```
LoadModeFile BootResources:Configure.Monitors.Other.Gerald
WimpMode X1896 Y1020 C16M
```

Once again the (Monitor) Modefile must exist at the location specified as: $.!Boot.Configure.Monitors.Other.Gerald – and it must contain a definition for 1896x1020 with 16 million colours.

AnyMode Module

S Harrison has created a module which can be used with RPCEmu (and the RaspberryPi) to allow any screen mode to run by entering only the X and Y resolutions into the Display Manager submenu. It has the limitation that the horizontal resolution must be divisible by 32 and the vertical resolution by 2.

In use place the AnyMode module in the $.!Boot.Choices.Boot.PreDesk directory and then click <menu> over the monitor icon to reveal the Display menu. Select Mode and the Mode menu will open. Type into the window of the Mode menu the resolution (and number of colours) that you desire. If it can be achieved then the mode will change; if it cannot be achieved then the mode will not change.



The website on which I found this module is www.pi-star.co.uk/riscos/anymode; however, when I tried this website recently it is no longer available.

Once you have found a suitable resolution then this can be used in Configure.Monitor file as described above.

I am unsure of the copyright situation with regard to the AnyMode module. I have included the ReadMe file for AnyMode in the files available on the Archive website.

Bio-Bit

Gerald has discovered that often his ‘Old Eyes’ can see things that ‘Young Eyes’ don’t. Out with the old and in with the new is not always a good thing.