

HowToCode7

| |
|----------------------|
| COLLABORATORS |
|----------------------|

| | | | |
|---------------|------------------------------|----------------|------------------|
| | <i>TITLE :</i> HowToCode7 | | |
| <i>ACTION</i> | <i>NAME</i> | <i>DATE</i> | <i>SIGNATURE</i> |
| WRITTEN BY | | March 28, 2025 | |

| |
|-------------------------|
| REVISION HISTORY |
|-------------------------|

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------|-------------|------|
| | | | |

Contents

| | | |
|----------|--------------------------------------|----------|
| 1 | HowToCode7 | 1 |
| 1.1 | HowToCode: Video Standards | 1 |

Chapter 1

HowToCode7

1.1 HowToCode: Video Standards

Coping with different Video Standards

As an European myself, I'm naturally biased against the inferior NTSC system, but even though the US & Canada have a relatively minor Amiga community compared with Europe (Sorry, it's true :-)) we should still help them out, even though they've never done a PAL Video Toaster for us (sob!).

You have two options.

Firstly, you could write your code only to use the first 200 display lines, and leave a black border at the bottom. This annoys PAL owners, who rightly expect things to have a full display. It took long enough for European games writers to work out that PAL displays were better.

You could write code that automatically checked which system it is running on and ran the correct code accordingly:

Two pieces of check code are needed. One handles the simple NTSC or PAL differences under Kickstart 1.2/1.3, but under Kickstart 2.0 or higher, everything is complicated by new monitor types...

1.3 Check for NTSC/PAL

```
move.l 4.w,a6          ; exechbase
cmp.b #50,VBlankFrequency(a6)
beq.s .pal

        jmp I'm NTSC
.pal    jmp I'm PAL
```

2.x/3.x Check for NTSC/PAL

```
move.l  GfxBase,a6
        btst  #2,gb_DisplayFlags(a6)  ; Check for PAL
bne.s   .pal

        jmp  I'm NTSC
.pal    jmp  I'm PAL
```

This test *may* work under 1.3, but the code in Kickstart 1.2/1.3 rom is totally broken, so it can guess wrong about NTSC/PAL quite often!

Check startup.asm for a way to combine the two tests together...

This is fine *EXCEPT* for one thing... It only tells you what video system the system was booted under. If you have a PAL machine and you run a 60hz interlaced workbench (for less flicker) it's fine because the demo still runs in 50hz (as long as your system runs from 50hz power).

However, NTSC owners can lose out, because if their display is capable of PAL (by running a PAL fixer or running a PAL display mode) this code completely ignores them and runs NTSC anyway, however, if NTSC users select PAL from their boot menu (2.x and 3.0 only) then it will work.

For demos and games you'd probably only want to run 50Hz anyway..

Now, if you want to force a machine into the other display system you need some magic pokes: Here you go (beware other bits in \$dffldc can do nasty things. One bit can reverse the polarity on the video sync, not to healthy for some monitors I've heard...)

To turn a NTSC system into PAL (50Hz)

```
move.w  #32,$dffldc    ; Magically PAL
```

To turn a PAL system into NTSC (60Hz)

```
move.w  #0,$dffldc     ; Magically NTSC
```

Remember: Not all displays can handle both display systems! Commodore 1084/1084S, Philips 8833/8852 and multisync monitors will, but very few US TV's will handle PAL signals.

It might be polite for PAL demos to ask NTSC users if they wish to switch to PAL (by the magic poke) or quit.
