# 0dd33f78-4

"

**COLLABORATORS**

| | *TITLE* :<br><br>0dd33f78-4 | |
|---|---|---|

| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
|---|---|---|---|
| WRITTEN BY | " | August 22, 2024 | |

**REVISION HISTORY**

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

# Contents

# Chapter 1

# 0dd33f78-4

## 1.1 v0.92 Read this v0.92.

```
                *** AGABoost 0.94. 1996-07-15. ***

DESCRIPTION
-----------

High speed video driver for Shapeshifter. Doesn't need mmu.



DISCLAMER
---------

 The author assumes no responsibility or liability whatsoever for any damage
 or dataloss caused by using any version of the AGABoost driver or any file in  ↩
    this
 package.


 See the appropriate "please read" file.


Here is the Index.
```

## 1.2 Index

```
AGABoost0.9 Index
-----------------

Distribution.

Start~text.

Installation.
```

## 1.3 Installation

```
 Put the driver files that you are interested of using
 in any dir you like (eg. your shapeshifter dir/Video Drivers).
```

## 1.4 Usage

```
USAGE
-----

 Go to the "Graphics" settings menu.

 A) Select "Screen type" as "External".

 B) Select the AGABoost file that suits best for you
    in the "External driver" string gadget.

 C) Set "Refresh rate" to an apropriate value and use the version of the driver
    with the SAME FILENAME EXTENSION AS THE SETTING you are using.


    Recommendations when using a 70ns display mode and a 640*480 screen
    (like (non mode promoted) Pal Hires, Ntsc Hires, Euro36 Hires) EC :

      Amiga4000/030     : refresh3.
      Turbo030 25mhz+     : refresh3 or maybe 2.
      Turbo030 40mhz+ & Turbo040  : refresh2.
      Amiga4000/040     : refresh2.

    Recommendations when using a 70ns display mode and a 640*480 screen
    (like (non mode promoted) Pal Hires, Ntsc Hires, Euro36 Hires) MMU :

      Turbo040        : refresh2.
      Amiga4000/040     : refresh2.


    When using a 35ns display mode and a 640*480 screen:
```

(like dblpal hires, dblntsc hires, super72 superhires) :

Using the 256->64 mode, the above ones should work fine.

Using the standard version will be slow as as almost all chipmem bandwith is needed for the video dma.

## 1.5 Memory usage

MEMORY USAGE (bytes)
---

| AGABoost driver v. | Mac Screen Fast | Other Fast | Chip |
|---|---|---|---|
| AGABoost (B1230 dir) | screenw*screenh | 160k | screenw*screenh*2 |
| AGABoost (A4K040 dir) | screenw*screenh | 40k | screenw*screenh*2 |
| AGABoost256>64 | screenw*screenh | 565k | screenw*screenh*1.5 |
| AGABoost256>ecs64 | screenw*screenh | 565k | screenw*screenh*1.5 |

"Other fast" will vary with a few k depending on screen size.

Add some 20k for the mmu drivers.
Divide the amount of chipram with 2 for the skipy driver.
Divide the amount of chipram with 4 for the skipxy driver.

NOTE: Shapeshifter's memory setting DOES NOT SUBTRACT the fastram needed for the graphics.
(selecting 320*200 1bit will show as much mem available as 4096*4096 24bit)

## 1.6 256>64 & 256>halfbrite64

About 256>64 dithered modes
---

The 256>64 mode provides high performance using 35nS modes like dblpal hires with aga chipset.

The kind of chunky_to_planar code used performs best on 030:s.

The 256>hb64 mode will make it possible for you to run 256color applications with an ecs amiga in lores.

Note that the color reduction isn't particularly optimized for the halfbrite concept. This combined with only a 12bit palette will make the quality significantly lower than with the aga 256>64 mode.

```
 Drawback:

 Palette setting time is extremely long with these modes invoking the init
  of a 512k chunky2planar table.
 None of this code is particilarly optimized.

 Therefore color fading and color cycling situations won't work so well.
 To avoid system stall the drivers will, after setting the palette,
 wait for 2 seconds before allowing the next setting.

 A deliberate green/blue color flash is provided at every done palette setting.

 If you experience a situation where the palette get's set continously,
 "that you can stand", you can toggle the flash off by
 HOLDING RIGHT MOUSE BUTTON DOWN during two changes.

The color cycling problem could without too much effort be solved, for
 cyclings with a limited amount of colors.
```

## 1.7 Technical stuff.

```
TECHNICAL STUFF (very rough text)
---------------------------------


 C2P TECHNIQUE
 -------------

 The driver uses the blitter. Due to that the maximum throughput speed will be
 ~=320*216 at a 25fps rate ~=1.728mb/s.
 But you probably won't use bigger windows for (graphics intensive) game &  ↩
    animation playing.

 Note also that the maxiumum amount of memory the cpu can move to chipram is only  ↩
    a few times
 higher than that, so for instance having a 640*480 25fps cpu_only conversion  ↩
    routine would mean
 that almost all or more than all :-) of the cpu time would go to chipmem data  ↩
    shuffling.


 C2P TECHNIQUE, drivers in "B1230" dir.
 --------------------------------------

 Normal "modern" table technique chunky_to_planar code.

 C2P TECHNIQUE, drivers in "A4K040" dir.
 ---------------------------------------

 "Classic" register manipulation chunky_to_planar code.



 Change noticing technique EC drivers
```

```
 ------------------------------------


 The driver uses a table keeping track of differences,
 sometimes called "delta buffer".

 Checking the difference with every pixel at every refresh takes a lot of cpu time ↩
    .
 Therefore this driver uses a few simple methods to minimize the time:


 MAINLY:
   Blocks that haven't been changed for 3 seconds
   will during low graphics load get checked between:
     every 5th and every 2nd refresh depending on which _refresh extension of the  ↩
        driver you are using.
   and during high graphics load (change ~>=21% of 320*200 ) :
     twice a second when using the driver with correct _refresh extension.

   ( The "heavy graphics load" time gain will ofcourse not be seen
     when running cpu benchmark tests like the Speedometer "benchmark mix". )


   This method will give a limited "slow on reactions effect", usually not
   so disturbing. (except mainly for 2d platform games where the
                   characters are thin enough and moving fast enough)

   ( A more sophisticated way would be to instead patch the MacOS Quickdraw
     function as I understand emplant1200 does )


 Change noticing technique MMU drivers
 -------------------------------------

 The above mentioned method is used, with nicer settings in situations
 when it gives any contribution.
 (mainly so those 2d platform games get playable)

 On the mmu side, the modified bits of the page descriptors are used to
 further "light down" the diff. checking.
```

## 1.8 Other notes.

```
OTHER NOTES
-----------

 The code is mainly written according to C.Bauer's guidelines.
   However, it will not give more than one screen if used with a possible future  ↩
      implementation of
   Shapeshifter allowing multiple screens.

   You should not mix this up with the current ability of using two monitors.
   Ofcourse you can use AB with one monitor and a gfxcard with the other.
```

## 1.9 History.

```
HISTORY
-------

v0.8  (1996-05-04) : Initial release.


v0.9  (1996-05-26) : * Bugfix: A memfree did at exit previously leave a few
                       bytes less than the corresponding memalloc at start.
                     * 256->64 driver for highspeed operation also with 35ns
                       screenmodes implemented.
                     * 256->halfbrite64 driver for ecs lores implemented.
                     * Corrected / modified definition of "heavy graphics load",
                       leaves more cpupower when the macprogram itself is too
                       slow to update at each refresh.
                     * Version with A4000/040 optimized c2p loop implemented.
                     * Borderblank set as default.


v0.92 (1996-06-30) : * Senility fix: Borderblank was previously only put into the
(not distributed)                    256>64 driver.
                     * Version utilizing 040mmu.
                     * With pal, euro36 & ntsc lores screenmodes difference
                       check and refresh only work with the currently visible
                       part of screen.

                     These changes are only done with the aga drivers.


v0.94 (1996-07-15)   * Game playing modes. ( see games file for descriptions )
                     * Small functional change of the mmu version.
                     * Fix: The 256>64 driver previously allocated more chipram
                            than it uses.
```

## 1.10 Author.

```
AUTHOR
------

 Dennis Arkeryd.

 EMAIL: z94arde@mtek.chalmers.se
```

## 1.11 Distribution.

AGABoost is freeware.

The archive is freely distributable as long as you keep it intact in it's original form with all files included.