

# grogu setup

## Third Try

In summary, Second Try to install VPiXX 3DPiXX was doomed by having copied the wrong file to `/usr/share/X11/xorg.conf.d/xorg.conf`.

However, I didn't understand that until part of the way through this here third try.

After declaring an end to the Second Try (on Monday, 2024-02-19), I decided to do a clean installation of the entire system. (Of course, the `/home` directory would persist, because it's on its own partition.)

## Clean Installation of Xubuntu

Installed 22.04 LTS from my USB drive.

## Installed DatapiXX tools from USB drive that came with 3DPiXX

This seemed better than the version that I could get from the VPiXX website, for some reason.

## Made link of DatapiXX.mex

From `'/usr/share/VPiXX Software Tools/Software Tools/DatapiXXToolbox_trunk/mexdev/build/octave/linux64/DatapiXX_octave5.mex'` to:

```
/usr/share/psychtoolbox-3/PsychBasic/Octave4LinuxFiles64/DatapiXX.mex
```

... after putting that directory at the top of my Octave path.

So that:

```
lrwxrwxrwx 1 root 117 Feb 20 10:53
/usr/share/psychtoolbox-3/PsychBasic/Octave4LinuxFiles64/Datapixx.mex
->
'/usr/share/VPixx Software Tools/Software
Tools/DatapixxToolbox_trunk/mexdev/build/octave/linux64/Datapixx_octave5.mex'
```

## Copied wrong, then correct xorg.conf file to /usr/share/X11/xorg.conf.d/

2024-02-20 12:20

First tried

```
xorg.conf.new.1024x768_120
```

This breaks PTB; it won't do anything, throws several kinds of errors. Can't even run XorgConfCreator (or whatever the function's name is), either.

Then tried

```
90-ptbconfig_single_xscreen_amdgpu_MODIFIED.conf
```

This allows DatapixxImagingStereoDemo to work!

- That is, the execution of the program on Octave.
- EdgeVR LCD glasses (the ones that came from VPixx) wouldn't work/sync, though.
- Third-party LCD glasses worked, but only for less than a minute, just like during the First Try, before stopping to work (both lenses remained clear, no alternation and thus no stereopsis).

## Tried switching Datapixx.mex link to point to Datapixx\_xenial.mex

2024-02-23

Summary: Datapixx octave function stopped working.

Did:

```
anthony@grogu:Octave4LinuxFiles64$ ln -sf "/usr/share/VPiXX Software  
Tools/Software  
Tools/DatapixxToolbox_trunk/mexdev/build/octave/linux64/Datapixx_xenial.mex"  
Datapixx.mex
```

Got message that Datapixx.mex not found when tried using Octave, specifically when trying to run Demos or trying to do Datapixx('Open').

So I changed the link back, so that it points to Datapixx\_octave5.mex.

## PTB messages after successfully running VBLSyncTest

```
octave:2> VBLSyncTest
```

```
PTB-INFO: This is Psychtoolbox-3 for GNU/Linux X11, under GNU/Octave 64-Bit  
(Version 3.0.19 - Build date: Dec 17 2023).
```

```
PTB-INFO: OS support status: Linux 6.5.0-21-generic Supported.
```

```
PTB-INFO: Type 'PsychtoolboxVersion' for more detailed version information.
```

```
PTB-INFO: Most parts of the Psychtoolbox distribution are licensed to you  
under terms of the MIT License, with
```

```
PTB-INFO: some restrictions. See file 'License.txt' in the Psychtoolbox root  
folder for the exact licensing conditions.
```

```
PTB-INFO: For information about paid support, support memberships and other  
commercial services, please type
```

```
PTB-INFO: 'PsychPaidSupportAndServices'.
```

```
PTB-INFO: Connected to Advanced Micro Devices, Inc. [AMD/ATI] Polaris 20 XL  
[Radeon RX 580 2048SP] GPU with DCE-11.0 display engine [6 heads].
```

```
PTB-INFO: OpenGL-Renderer is AMD :: AMD Radeon RX 580 2048SP (polaris10, LLVM  
15.0.7, DRM 3.54, 6.5.0-21-generic) :: 4.6 (Compatibility Profile) Mesa  
23.2.1-1ubuntu3.1~22.04.2
```

```
PTB-INFO: VBL startline = 768 , VBL Endline = 824
```

```
PTB-INFO: Measured monitor refresh interval from beamposition = 8.344144 ms  
[119.844525 Hz].
```

```
PTB-INFO: Will try to use OS-Builtin OpenML sync control support for accurate  
Flip timestamping.
```

```
PTB-INFO: Measured monitor refresh interval from VBLsync = 8.344126 ms
```

```
[119.844790 Hz]. (50 valid samples taken, stddev=0.001443 ms.)
PTB-INFO: Reported monitor refresh interval from operating system = 8.343972
ms [119.847000 Hz].
PTB-INFO: Small deviations between reported values are normal and no reason to
worry.
The refresh interval reported by the operating system is 8.33333 ms.
libptbdrawtext_ftgl: External 'DrawText' text rendering plugin initialized.
libptbdrawtext_ftgl: Maximum number of cacheable fonts is 40, minimum number
of supported concurrent windows is 10.
libptbdrawtext_ftgl: This plugin uses multiple excellent free software
libraries to do its work:
libptbdrawtext_ftgl: OGLFT (http://oglft.sourceforge.net/) the OpenGL-FreeType
library.
libptbdrawtext_ftgl: The FreeType-2 (http://freetype.sourceforge.net/)
library.
libptbdrawtext_ftgl: The FontConfig (http://www.fontconfig.org) library.
Version Id: 21301
libptbdrawtext_ftgl: Thanks!
```

```
Measured refresh interval, as reported by "GetFlipInterval" is 8.34413 ms.
(nsamples = 0, stddev = 0.00000 ms)
^[PTB missed 0 out of 213 stimulus presentation deadlines.
One missed deadline is ok and an artifact of the measurement.
PTB completed 0 stimulus presentations before the requested target time.
Have a look at the plots for more details...
```

## PTB messages after running DatapixxImagingStereoDemo

(Unsuccessfully in that the LCD glasses that came from VPixx don't do anything at all. Pressing their button before execution of the demo's script does nothing. Pressing the button after the script has begun makes the glasses turn opaque [both lenses] and then alternate; and then a second press makes them alternate a few times again; and then after that additional button presses do nothing [lenses remain clear].)

PTB output:

```
octave:3> Datapixx('Open')
ans = 1
octave:4> DatapixxImagingStereoDemo
```

```
PTB-INFO: This is Psychtoolbox-3 for GNU/Linux X11, under GNU/Octave 64-Bit
(Version 3.0.19 - Build date: Dec 17 2023).
```

PTB-INFO: OS support status: Linux 6.5.0-21-generic Supported.  
 PTB-INFO: **Type** 'PsychtoolboxVersion' for more detailed version information.  
 PTB-INFO: Most parts of the Psychtoolbox distribution are licensed to you under terms of the MIT License, with  
 PTB-INFO: some restrictions. See file 'License.txt' in the Psychtoolbox root folder for the exact licensing conditions.

PTB-INFO: For information about paid support, support memberships and other commercial services, please type  
 PTB-INFO: 'PsychPaidSupportAndServices'.

PTB-INFO: Connected to Advanced Micro Devices, Inc. [AMD/ATI] Polaris 20 XL [Radeon RX 580 2048SP] GPU with DCE-11.0 display engine [6 heads].

PTB-INFO: OpenGL-Renderer is AMD :: AMD Radeon RX 580 2048SP (polaris10, LLVM 15.0.7, DRM 3.54, 6.5.0-21-generic) :: 4.6 (Compatibility Profile) Mesa 23.2.1-lubuntu3.1~22.04.2

PTB-INFO: VBL startline = 768 , VBL Endline = 824

PTB-INFO: Measured monitor refresh interval from beamposition = 8.344100 ms [119.845154 Hz].

PTB-INFO: Will try to use OS-Builtin OpenML sync control support for accurate Flip timestamping.

PTB-INFO: Measured monitor refresh interval from VBLsync = 8.344126 ms [119.844790 Hz]. (50 valid samples taken, stddev=0.000414 ms.)

PTB-INFO: Reported monitor refresh interval from operating system = 8.343972 ms [119.847000 Hz].

PTB-INFO: Small deviations between reported values are normal and no reason to worry.

PTB-INFO: SetDitherMode: Trying to disable digital display dithering on display head 0.

PTB-INFO: SetDitherMode: Dithering already disabled. Skipped.

LoadIdentityClut: Used GPU low-level setup code to configure (hopefully) perfect identity pixel passthrough.

ERROR: Invalid device for current operation

ERROR: Invalid device for current operation

warning: Matlab-style short-circuit operation performed for operator |

warning: called from

DatapixxImagingStereoDemo at line 203 column 5

warning: Matlab-style short-circuit operation performed for operator |

warning: called from

DatapixxImagingStereoDemo at line 199 column 5

^[N.Dots Mean (s) Max (s) %>20ms %>30ms

1000 0.008 0.033 0.00 0.00

# Basic info about the GPU, Linux system, Octave, PTB and Datapixx

2024-02-23

## Linux system info.

```
anthony@grogu:DatapixxDemos$ uname -a
Linux grogu 6.5.0-21-generic #21~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Feb 9
13:32:52 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
```

## lspci -v output (GPU info.)

```
01:00.0 VGA compatible controller: Advanced Micro Devices, Inc. [AMD/ATI]
Polaris 20 XL [Radeon RX 580 2048SP] (rev ef) (prog-if 00 [VGA controller])
Subsystem: Advanced Micro Devices, Inc. [AMD/ATI] Polaris 20 XL [Radeon RX
580 2048SP]
Flags: bus master, fast devsel, latency 0, IRQ 138
Memory at a0000000 (64-bit, prefetchable) [size=256M]
Memory at b0000000 (64-bit, prefetchable) [size=2M]
I/O ports at 3000 [size=256]
Memory at b2200000 (32-bit, non-prefetchable) [size=256K]
Expansion ROM at 000c0000 [disabled] [size=128K]
Capabilities: <access denied>
Kernel driver in use: amdgpu
Kernel modules: amdgpu
```

## xorg.conf file

As it says above, I had copied 90-ptbconfig\_single\_xscreen\_amdgpu\_MODIFIED.conf to /usr/share/X11/xorg.conf.d/xorg.conf

Contents of this xorg.conf file:

```
# Auto generated xorg.conf - Created by Psychtoolbox XOrgConfCreator.

Section "ServerFlags"
    Option "AutoAddGPU"      "false"
EndSection
```

```

Section "Device"
    Identifier "Card0"
    Driver "amdgpu"
    Option "VariableRefresh" "off"
EndSection

Section "Screen"
    Identifier "Screen0"
    Device "Card0"
    # added
    Monitor "DVI-D-0"
    # commented out 2023-10-13
    #DefaultDepth 30
EndSection

# added section
Section "Monitor"
    Identifier "0-CRT1"
    Option "VendorName" "ATI Proprietary Driver"
    Option "ModelName" "Generic Autodetecting Monitor"
    Option "DPMS" "true"
    Option "PreferredMode" "1600x1200"
    Option "TargetRefresh" "85"
    Option "Position" "0 0"
    Option "Rotate" "normal"
    Option "Disable" "false"
EndSection

# added section
Section "Monitor"
    # 1024x768 119.80 Hz (CVT) hsync: 98.96 kHz; pclk: 137.75 MHz
    Identifier "DVI-D-0"
    Modeline "1024x768_120.00" 137.8 1024 1104 1208 1392 768 771 775 826 -
    hsync +vsync
    # Option "VendorName" "ATI Proprietary Driver"
    Option "ModelName" "Generic Autodetecting Monitor"
    Option "DPMS" "true"
    Option "PreferredMode" "1024x768"
    Option "TargetRefresh" "120"
    Option "Position" "0 0"
    Option "Rotate" "normal"
    Option "Disable" "false"
EndSection

```

# OpenGL info.

## glxgears output

```
anthony@grogu:DatapixxDemos$ glxgears
Running synchronized to the vertical refresh. The framerate should be
approximately the same as the monitor refresh rate.
687 frames in 5.0 seconds = 137.207 FPS
600 frames in 5.0 seconds = 119.857 FPS
600 frames in 5.0 seconds = 119.845 FPS
X connection to :0.0 broken (explicit kill or server shutdown).
```

## glxgears -info output

Complete output (GL\_EXTENSIONS field is very long):

```
GL_RENDERER = AMD Radeon RX 580 2048SP (polaris10, LLVM 15.0.7, DRM 3.54,
6.5.0-21-generic)
GL_VERSION = 4.6 (Compatibility Profile) Mesa 23.2.1-lubuntu3.1~22.04.2
GL_VENDOR = AMD
GL_EXTENSIONS = GL_ARB_multisample GL_EXT_abgr GL_EXT_bgra GL_EXT_blend_color
GL_EXT_blend_minmax GL_EXT_blend_subtract GL_EXT_copy_texture
GL_EXT_subtexture GL_EXT_texture_object GL_EXT_vertex_array
GL_EXT_compiled_vertex_array GL_EXT_texture GL_EXT_texture3D
GL_IBM_rasterpos_clip GL_ARB_point_parameters GL_EXT_draw_range_elements
GL_EXT_packed_pixels GL_EXT_point_parameters GL_EXT_rescale_normal
GL_EXT_separate_specular_color GL_EXT_texture_edge_clamp
GL_SGIS_generate_mipmap GL_SGIS_texture_border_clamp
GL_SGIS_texture_edge_clamp GL_SGIS_texture_lod GL_ARB_framebuffer_sRGB
GL_ARB_multitexture GL_EXT_framebuffer_sRGB GL_IBM_multimode_draw_arrays
GL_IBM_texture_mirrored_repeat GL_ARB_texture_cube_map GL_ARB_texture_env_add
GL_ARB_transpose_matrix GL_EXT_blend_func_separate GL_EXT_fog_coord
GL_EXT_multi_draw_arrays GL_EXT_secondary_color GL_EXT_texture_env_add
GL_EXT_texture_filter_anisotropic GL_EXT_texture_lod_bias
GL_INGR_blend_func_separate GL_NV_blend_square GL_NV_light_max_exponent
GL_NV_texgen_reflection GL_NV_texture_env_combine4 GL_S3_s3tc
GL_SUN_multi_draw_arrays GL_ARB_texture_border_clamp
GL_ARB_texture_compression GL_EXT_framebuffer_object
GL_EXT_texture_compression_s3tc GL_EXT_texture_env_combine
GL_EXT_texture_env_dot3 GL_MESA_window_pos GL_NV_packed_depth_stencil
GL_NV_texture_rectangle GL_ARB_depth_texture GL_ARB_occlusion_query
GL_ARB_shadow GL_ARB_texture_env_combine GL_ARB_texture_env_crossbar
GL_ARB_texture_env_dot3 GL_ARB_texture_mirrored_repeat GL_ARB_window_pos
GL_ATI_fragment_shader GL_EXT_stencil_two_side GL_EXT_texture_cube_map
```



GL\_NV\_copy\_depth\_to\_color GL\_NV\_depth\_clamp GL\_NV\_fog\_distance  
GL\_NV\_half\_float GL\_APPLE\_packed\_pixels GL\_ARB\_draw\_buffers  
GL\_ARB\_fragment\_program GL\_ARB\_fragment\_shader GL\_ARB\_shader\_objects  
GL\_ARB\_vertex\_program GL\_ARB\_vertex\_shader GL\_ATI\_draw\_buffers  
GL\_ATI\_texture\_env\_combine3 GL\_ATI\_texture\_float GL\_EXT\_depth\_bounds\_test  
GL\_EXT\_shadow\_funcs GL\_EXT\_stencil\_wrap GL\_MESA\_pack\_invert  
GL\_NV\_primitive\_restart GL\_ARB\_depth\_clamp GL\_ARB\_fragment\_program\_shadow  
GL\_ARB\_half\_float\_pixel GL\_ARB\_occlusion\_query2 GL\_ARB\_point\_sprite  
GL\_ARB\_shading\_language\_100 GL\_ARB\_sync GL\_ARB\_texture\_non\_power\_of\_two  
GL\_ARB\_vertex\_buffer\_object GL\_ATI\_blend\_equation\_separate  
GL\_EXT\_blend\_equation\_separate GL\_OES\_read\_format GL\_ARB\_color\_buffer\_float  
GL\_ARB\_pixel\_buffer\_object GL\_ARB\_texture\_compression\_rgtc  
GL\_ARB\_texture\_float GL\_ARB\_texture\_rectangle GL\_ATI\_texture\_compression\_3dc  
GL\_EXT\_packed\_float GL\_EXT\_pixel\_buffer\_object GL\_EXT\_texture\_compression\_dxt1  
GL\_EXT\_texture\_compression\_rgtc GL\_EXT\_texture\_mirror\_clamp  
GL\_EXT\_texture\_rectangle GL\_EXT\_texture\_sRGB GL\_EXT\_texture\_shared\_exponent  
GL\_ARB\_framebuffer\_object GL\_EXT\_framebuffer\_blit  
GL\_EXT\_framebuffer\_multisample GL\_EXT\_packed\_depth\_stencil  
GL\_ARB\_vertex\_array\_object GL\_ATI\_separate\_stencil GL\_ATI\_texture\_mirror\_once  
GL\_EXT\_draw\_buffers2 GL\_EXT\_draw\_instanced GL\_EXT\_gpu\_program\_parameters  
GL\_EXT\_gpu\_shader4 GL\_EXT\_texture\_array GL\_EXT\_texture\_compression\_latc  
GL\_EXT\_texture\_integer GL\_EXT\_texture\_sRGB\_decode GL\_EXT\_timer\_query  
GL\_OES\_EGL\_image GL\_AMD\_performance\_monitor GL\_EXT\_texture\_buffer\_object  
GL\_AMD\_texture\_texture4 GL\_ARB\_copy\_buffer GL\_ARB\_depth\_buffer\_float  
GL\_ARB\_draw\_instanced GL\_ARB\_half\_float\_vertex GL\_ARB\_instanced\_arrays  
GL\_ARB\_map\_buffer\_range GL\_ARB\_texture\_buffer\_object GL\_ARB\_texture\_rg  
GL\_ARB\_texture\_swizzle GL\_ARB\_vertex\_array\_bgra GL\_EXT\_texture\_swizzle  
GL\_EXT\_vertex\_array\_bgra GL\_NV\_conditional\_render GL\_AMD\_conservative\_depth  
GL\_AMD\_depth\_clamp\_separate GL\_AMD\_draw\_buffers\_blend  
GL\_AMD\_seamless\_cubemap\_per\_texture GL\_AMD\_shader\_stencil\_export  
GL\_ARB\_ES2\_compatibility GL\_ARB\_blend\_func\_extended GL\_ARB\_compatibility  
GL\_ARB\_debug\_output GL\_ARB\_draw\_buffers\_blend GL\_ARB\_draw\_elements\_base\_vertex  
GL\_ARB\_explicit\_attrib\_location GL\_ARB\_fragment\_coord\_conventions  
GL\_ARB\_provoking\_vertex GL\_ARB\_sample\_shading GL\_ARB\_sampler\_objects  
GL\_ARB\_seamless\_cube\_map GL\_ARB\_shader\_stencil\_export  
GL\_ARB\_shader\_texture\_lod GL\_ARB\_tessellation\_shader  
GL\_ARB\_texture\_buffer\_object\_rgb32 GL\_ARB\_texture\_cube\_map\_array  
GL\_ARB\_texture\_gather GL\_ARB\_texture\_multisample GL\_ARB\_texture\_query\_lod  
GL\_ARB\_texture\_rgb10\_a2ui GL\_ARB\_uniform\_buffer\_object  
GL\_ARB\_vertex\_type\_2\_10\_10\_10\_rev GL\_ATI\_meminfo GL\_EXT\_provoking\_vertex  
GL\_EXT\_texture\_snorm GL\_MESA\_texture\_signed\_rgba GL\_NV\_copy\_image  
GL\_NV\_texture\_barrier GL\_ARB\_draw\_indirect GL\_ARB\_get\_program\_binary  
GL\_ARB\_gpu\_shader5 GL\_ARB\_gpu\_shader\_fp64 GL\_ARB\_robustness  
GL\_ARB\_separate\_shader\_objects GL\_ARB\_shader\_bit\_encoding  
GL\_ARB\_shader\_precision GL\_ARB\_shader\_subroutine  
GL\_ARB\_texture\_compression\_bptc GL\_ARB\_timer\_query GL\_ARB\_transform\_feedback2  
GL\_ARB\_transform\_feedback3 GL\_ARB\_vertex\_attrib\_64bit GL\_ARB\_viewport\_array  
GL\_EXT\_direct\_state\_access GL\_EXT\_shader\_image\_load\_store

GL\_EXT\_vertex\_attrib\_64bit GL\_NV\_vdpau\_interop GL\_AMD\_multi\_draw\_indirect  
GL\_ANGLE\_texture\_compression\_dxt3 GL\_ANGLE\_texture\_compression\_dxt5  
GL\_ARB\_base\_instance GL\_ARB\_compressed\_texture\_pixel\_storage  
GL\_ARB\_conservative\_depth GL\_ARB\_internalformat\_query  
GL\_ARB\_map\_buffer\_alignment GL\_ARB\_shader\_atomic\_counters  
GL\_ARB\_shader\_image\_load\_store GL\_ARB\_shading\_language\_420pack  
GL\_ARB\_shading\_language\_packing GL\_ARB\_texture\_storage  
GL\_ARB\_transform\_feedback\_instanced GL\_EXT\_framebuffer\_multisample\_blit\_scaled  
GL\_EXT\_transform\_feedback GL\_AMD\_query\_buffer\_object  
GL\_AMD\_shader\_trinary\_minmax GL\_AMD\_vertex\_shader\_layer  
GL\_AMD\_vertex\_shader\_viewport\_index GL\_ARB\_ES3\_compatibility  
GL\_ARB\_arrays\_of\_arrays GL\_ARB\_clear\_buffer\_object GL\_ARB\_compute\_shader  
GL\_ARB\_copy\_image GL\_ARB\_explicit\_uniform\_location  
GL\_ARB\_fragment\_layer\_viewport GL\_ARB\_framebuffer\_no\_attachments  
GL\_ARB\_invalidate\_subdata GL\_ARB\_multi\_draw\_indirect  
GL\_ARB\_program\_interface\_query GL\_ARB\_robust\_buffer\_access\_behavior  
GL\_ARB\_shader\_image\_size GL\_ARB\_shader\_storage\_buffer\_object  
GL\_ARB\_stencil\_texturing GL\_ARB\_texture\_buffer\_range  
GL\_ARB\_texture\_query\_levels GL\_ARB\_texture\_storage\_multisample  
GL\_ARB\_texture\_view GL\_ARB\_vertex\_attrib\_binding GL\_KHR\_debug  
GL\_KHR\_robustness GL\_KHR\_texture\_compression\_astc\_ldr GL\_AMD\_pinned\_memory  
GL\_ARB\_bindless\_texture GL\_ARB\_buffer\_storage GL\_ARB\_clear\_texture  
GL\_ARB\_compute\_variable\_group\_size GL\_ARB\_enhanced\_layouts  
GL\_ARB\_indirect\_parameters GL\_ARB\_internalformat\_query2 GL\_ARB\_multi\_bind  
GL\_ARB\_query\_buffer\_object GL\_ARB\_seamless\_cubemap\_per\_texture  
GL\_ARB\_shader\_draw\_parameters GL\_ARB\_shader\_group\_vote  
GL\_ARB\_shading\_language\_include GL\_ARB\_texture\_mirror\_clamp\_to\_edge  
GL\_ARB\_texture\_stencil8 GL\_ARB\_vertex\_type\_10f\_11f\_11f\_rev GL\_EXT\_debug\_label  
GL\_EXT\_shader\_integer\_mix GL\_NVX\_gpu\_memory\_info GL\_ARB\_ES3\_1\_compatibility  
GL\_ARB\_clip\_control GL\_ARB\_conditional\_render\_inverted GL\_ARB\_cull\_distance  
GL\_ARB\_derivative\_control GL\_ARB\_direct\_state\_access  
GL\_ARB\_get\_texture\_sub\_image GL\_ARB\_pipeline\_statistics\_query  
GL\_ARB\_shader\_texture\_image\_samples GL\_ARB\_texture\_barrier  
GL\_ARB\_transform\_feedback\_overflow\_query GL\_EXT\_polygon\_offset\_clamp  
GL\_EXT\_shader\_image\_load\_formatted GL\_KHR\_blend\_equation\_advanced  
GL\_KHR\_context\_flush\_control GL\_KHR\_robust\_buffer\_access\_behavior  
GL\_NV\_shader\_atomic\_int64 GL\_ARB\_ES3\_2\_compatibility GL\_ARB\_gpu\_shader\_int64  
GL\_ARB\_parallel\_shader\_compile GL\_ARB\_shader\_atomic\_counter\_ops  
GL\_ARB\_shader\_ballot GL\_ARB\_shader\_clock GL\_ARB\_shader\_viewport\_layer\_array  
GL\_EXT\_shader\_samples\_identical GL\_EXT\_texture\_sRGB\_R8 GL\_KHR\_no\_error  
GL\_KHR\_texture\_compression\_astc\_sliced\_3d GL\_ARB\_gl\_spirv  
GL\_ARB\_spirv\_extensions GL\_EXT\_window\_rectangles  
GL\_MESA\_shader\_integer\_functions GL\_ARB\_polygon\_offset\_clamp  
GL\_ARB\_texture\_filter\_anisotropic GL\_EXT\_memory\_object GL\_EXT\_memory\_object\_fd  
GL\_EXT\_semaphore GL\_EXT\_semaphore\_fd GL\_KHR\_parallel\_shader\_compile  
GL\_NV\_alpha\_to\_coverage\_dither\_control GL\_AMD\_framebuffer\_multisample\_advanced  
GL\_EXT\_EGL\_image\_storage GL\_EXT\_texture\_shadow\_lod GL\_INTEL\_blackhole\_render  
GL\_MESA\_framebuffer\_flip\_y GL\_NV\_compute\_shader\_derivatives GL\_EXT\_EGL\_sync

```
GL_EXT_demote_to_helper_invocation GL_NV_ES1_1_compatibility
VisualID 1300, 0x514
```

## Octave version

I installed Octave by doing

```
sudo apt install octave
```

```
anthony@grogu:DatapixxDemos$ octave --version
GNU Octave, version 6.4.0
Copyright (C) 2021 The Octave Project Developers.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.

Octave was configured for "x86_64-pc-linux-gnu".

Additional information about Octave is available at https://www.octave.org.

Please contribute if you find this software useful.
For more information, visit https://www.octave.org/get-involved.html

Read https://www.octave.org/bugs.html to learn how to submit bug reports.
```

## Psychtoolbox (PTB) version

I installed PTB for Octave using NeuroDebian (see: [this site](#)).

```
octave:5> PsychtoolboxVersion
ans = 3.0.19 - Flavor: Debian package - psychtoolbox-3
(3.0.19.7.dfsg1-1~nd22.04+1)
For more info visit:
http://neuro.debian.net/pkgs/octave-psychtoolbox-3.html
```

```
octave:6> Screen('Version')
ans =

    scalar structure containing the fields:

    version = 3.0.19.673758066
    major = 3
    minor = 0
```

```
point = 19
build = 6.7376e+08
date = Dec 17 2023
time = 03:01:06
module = Screen
project = OpenGL Psychtoolbox
os = GNU/Linux X11
language = GNU/Octave 64-Bit
authors =
```

## VPixx Datapixx Firmware version

```
octave:13> firmwareRev = Datapixx('GetFirmwareRev')
firmwareRev = 27
```

# Following advice from VPixx to fix LCD glasses functioning

2024-02-26

VPixx (Lindsey Fraser from technical support responding on behalf of Sophie Kenny) replied to me within an hour! Based on the detailed email instructions, I did both of these things:

## 1. Changed line 62 of DatapixxImagingStereoDemo.m

Changed the argument from "2" to "5":

```
Datapixx('SetVideoStereoVesaWaveform',5)
```

to

```
Datapixx('SetVideoStereoVesaWaveform',2)
```

This made it so that the LCD glasses at least started flickering. They were badly out of phase or frequency, so that there was no stereopsis at all, however.

## 2. Deleted /usr/share/VPixx Software Tools/ directory and installed latest version of VPixx software tools from their website

(<https://docs.vpixx.com/download/>).

This installed everything to /usr/local/share/VPixx Software Tools.

The version of DatapixxImagingStereoDemo.m that came with this had line 62 already corrected to have Datapixx('SetVideoStereoVesaWaveform',5).

I made a new symlink to Datapixx\_octave5.mex.

```
cd /usr/share/psychtoolbox-3/PsychBasic/Octave4LinuxFiles64/  
sudo ln -sf /usr/local/share/VPixx\ Software\ Tools\ Software\  
Tools\ Matlab\ mexfiles\ linux\ octave\ Datapixx_octave5.mex Datapixx.mex
```

Datapixx('Open') works.

DatapixxImagingStereoDemo.m “works,” but only in the way described above. It *does* work with the generic IR LCD glasses – there was good stereopsis – but there was also some flickering.

Also, the Datapixx demo still has its usual problem where the disparities get constantly larger (more crossed) until fusion is no longer possible. It starts out with the moving disparity region having a small uncrossed (concave) disparity, which then becomes zero (flat) and then eventually ridiculously large crossed (convex) disparity.

## Tried adjusting phase

2024-02-26

Following second email from Lindsey Fraser, I tried a number of different values for the waveform phase in

```
Datapixx('SetVideoStereoVesaPhase',100);
```

100 is the value that VPixx uses with their VIEWPixx monitor.

Nothing changed with different values for the phase parameter. Also, in any case, the flickering of the glasses stopped entirely after about 2 or 3 seconds past the onset of DatapixxImagingStereoDemo.m, after which there was no perceptible effect of the glasses, except for a dimming of the lower half of the screen produced by the left lens – weird, but not in a flickering way.

2024-02-27

Was advised that need to do

```
Datapixx('RegWr')
```

after the Set..Phase command in order for it to take effect.

I tried many different settings in the command's range of 0 to 255, and this did start the glasses flickering even before I started running the demo. However, this produced no real perceptible differences on the demo. Also, (alternating-eye) flicker still stops after approx. 2 seconds. After that, there must be some kind of high frequency flickering going on, because the effect is that I perceive a steady (non-flickering) dark horizontal band across the screen that drifts in the up/down direction. The vertical position of the band is slightly different for the two eyes/lenses.

## Modified xorg.conf file

2024-02-27

xorg.conf file refers to **two** monitors. I'm really not sure why I included two "monitor" blocks, especially since there is only one physical monitor. I might have tried to copy part of the file from the Diego lab PC from VT, but I can't remember.

### Monitor configuration:

AMD Radeon RX 580 GPU DVI-D port → Datapixx DVI-D-in → Datapixx VGA-out → Dell CRT

In any case, I'm going to remove the block referring two the "0-CRT1" monitor and leave the block for the "DVI-D-0" monitor, and see what happens.

I'm going to

1. Make a copy of my xorg.conf file that has the CRT monitor block removed.

I made a copy of the file 90-ptbconfig\_single\_xscreen\_amdgpu\_MODIFIED.conf named 90-ptbconfig\_single\_xscreen\_amdgpu\_MODIFIED\_DVI-D-0\_ONLY.conf and copied it to /usr/share/X11/xorg.conf.d/xorg.conf

2. Make a copy of my xorg.conf file that has the DVI-D-0 "monitor" block removed.

I named this ...MODIFIED\_0-CRT1\_ONLY.conf and I copied the 1024x768\_120Hz modeline line to the 0-CRT1 block and replaced the reference to DVI-D-0.

### RESULTS of 1:

The 1024x768 resolution wasn't available to me, let alone the 119.8Hz setting.

### RESULTS of 2:

It displayed a setting for 1024x768, but not the 119.8Hz option.

- **Summary:** It appears I *do* need blocks for *both* monitors in xorg.conf in order to get the 1284x768\_120Hz mode.

## Where I came up with the labels for the two monitors in xorg.conf

Now I recollect the process of making that xorg.conf file (in 2023-10). I couldn't get the 1024x768\_120Hz mode to show up as an option in the display settings, so I examined the (bash terminal) output of `xrandr -q`. It included references to both DVI-D-0 and 0-CRT1. So that's why I included them both in xorg.conf.

## Reason for choosing the 1024x768 resolution

This was the resolution I used with the same CRT monitor in 2014-15 when I ran the first HolesCrowd study using a VPixx 3DPixx system, also on Linux. I need to do a second experiment for that old study now, so I want to use the same viewing conditions.

## Changed setting of Datapixx('SetVideoStereoVesaWaveform',X) to zero

2024-02-28

Following advice from Lindsey Fraser again, I tried changing `Datapixx('SetVideoStereoVesaWaveform',5)` (see above for when changed to that value) to `Datapixx('SetVideoStereoVesaWaveform',0)`. (Followed by `Datapixx('RegWr')`.)

This worked (to a great extent)!

`DatapixxImagingStereoDemo.m` ran with perfect glasses function. (However, the problem with the demo where the dot disparities become increasingly larger over time until fusion is no longer possible persists.)

I tried running one of my own experiment scripts that I wrote based on `DatapixxImagingStereoDemo.m` back in 2014. Unfortunately, this didn't work as well.

There was a strongly noticeable dim horizontal band across the screen, which drifted and jumped around in the up/down direction.

I played with `Datapixx('SetVideoStereoVesaPhase',X)` to see if this had any effect on the band. It didn't reduce the presence of the band, for any values of X. The best I seemed able to do was to make the band dwell slightly longer around the top and/or bottom of the screen, instead of near the middle (while it drifted and jumped around). I got this behavior with the setting `Datapixx('SetVideoStereoVesaPhase',0)`.

**Plans:** The main thing I can do to recreate the conditions of the 2014 experiment where the 3DPixx

worked is to use MATLAB instead of Octave. I haven't tried that with my current 3DPiix yet, so I think I will.

## Installed MATLAB

2024-03-06

Installed MATLAB from Mathworks website, logged in under my Roanoke College account.

Downloaded R2023a

Unzipped the .zip file in ~/Downloads directory.

Did:

```
sudo ./install
```

MATLAB was installed to /usr/local/MATLAB/R2023a

Had to do `restoredefaultpath;matlabrc` first time opening MATLAB. Then I clicked "Save" in the path dialog. Hopefully that will prevent future can't find path errors.

## Installed PTB for MATLAB

I think when I installed the MATLAB-specific version from NeuroDebian, it overwrote the Octave-specific version.

I did this:

```
sudo apt-get install matlab-psychtoolbox-3
```

Instead of making symlink to Datapiix.mexa64, just put the directory holding the correct file at the top of my MATLAB path (/usr/local/share/VPiix Software Tools/Software Tools/Matlab/mexfiles/linux/matlab/Datapiix.mexa64). Before this, a copy of Datapiix.mexa64 from the psychtoolbox-3 subdirectories was being used by MATLAB, and it wouldn't do `Datapiix('Open')`. Now it works.

ProceduralGarbodiumDemo.m and VBLSyncTest.m work great.

## Tested 3DPiix with MATLAB PTB

Tried HolesCrowd\_PTB\_3StimCrowd\_TEST.m with the VPiix LCD glasses. It was somewhat better than



with Octave. There was still a dark band at the bottom of the screen, but it stayed at the bottom, so that its problem is that it causes a distraction, but not interference with the stimuli themselves.

The glasses did flicker noticeably, which is a bigger problem.

I checked the script, and this is the script that has `Datapixx('SetVideoStereoWaveform',0)` like I used with Octave last Friday.

**Update** (2024-02-06 13:20): The display had become set to 85 Hz somehow. I reset it to 119.8 Hz, and the flickering went away. The dark horizontal bar persists, however.

## The Datapixx code lines from the script HolesCrowd\_PTB\_3StimCrowd\_TEST.m

I am 100% sure that I copied these lines from `DatapixxImagingStereoDemo.m` in 2014:

```
% Hardware stereo buffers are a great idea, but they just seem to be broken on
so many systems.
% Set to 1 to try it out, or set to 0 to implement software frame-alternate
buffers.
useHardwareStereo = 0;

[... many lines later: ...]

% Prepare pipeline for configuration. This marks the start of a list of
% requirements/tasks to be met/executed in the pipeline:
PsychImaging('PrepareConfiguration');

% Tell PTB we want to display on a DataPixx device:
PsychImaging('AddTask', 'General', 'UseDataPixx');

% Decrease GPU workload
PsychImaging('AddTask', 'AllViews', 'RestrictProcessing', CenterRect([0
512 512], Screen('Rect', screenNumber)));

% Enable DATAPixx blueline support, and VIEWPixx scanning backlight for
optimal 3D
Datapixx('Open');
%Datapixx('EnableVideoScanningBacklight'); % Only required if a
VIEWPixx.
Datapixx('EnableVideoStereoBlueline');
% Datapixx('SetVideoStereoVesaWaveform', 2); % If driving NVIDIA
glasses %commented out 2024-02-28 by adc
Datapixx('SetVideoStereoVesaWaveform', 0); % If driving 3rd party
emitter %uncommented 2024-02-28 by adc
% Datapixx('SetVideoStereoVesaPhase',0); % Added 2024-02-28 by adc to try
getting rid of drifting horizontal band
Datapixx('RegWr');
```

```

% Consolidate the list of requirements (error checking etc.), open a
% suitable onscreen window and configure the imaging pipeline for that
% window according to our specs. The syntax is the same as for
% Screen('OpenWindow'):
if useHardwareStereo == 1
    [w, wRect]=PsychImaging('OpenWindow', screenNumber, 0, [], [], [], 1);
else
    [w, wRect]=PsychImaging('OpenWindow', screenNumber, 0);
end

% There seems to be a blueline generation bug on some OpenGL systems.
% SetStereoBlueLineSyncParameters(w, wRect(4)) corrects the
% bug on some systems, but breaks on other systems.
% We'll just disable automatic blueline, and manually draw our own
bluelines!
if useHardwareStereo == 1
    SetStereoBlueLineSyncParameters(w, wRect(4)+10);
end
blueRectLeftOn    = [0,                                wRect(4)-1, wRect(3)/4,
wRect(4)];
blueRectLeftOff   = [wRect(3)/4,    wRect(4)-1, wRect(3),    wRect(4)];
blueRectRightOn   = [0,                                wRect(4)-1, wRect(3)*3/4,
wRect(4)];
blueRectRightOff  = [wRect(3)*3/4, wRect(4)-1, wRect(3),    wRect(4)];

% Initially fill left- and right-eye image buffer with black background
% color:
if useHardwareStereo == 1
    Screen('SelectStereoDrawBuffer', w, 0);
    Screen('FillRect', w, BlackIndex(screenNumber));
    Screen('SelectStereoDrawBuffer', w, 1);
    Screen('FillRect', w, BlackIndex(screenNumber));
    Screen('Flip', w);
else
    Screen('FillRect', w, BlackIndex(screenNumber));
    Screen('Flip', w);
end

```

## Tinkering with HolesCrowd\_PTB\_3StimCrowd\_TEST.m

Commented out the line that skips the PTB sync tests. I had included this line because it was the only way I could get the stereopsis working in 2014, I think. But no reason to keep it now.

The line (line 6):

```
Screen('Preference', 'SkipSyncTests', 1);
```

Ironically, perhaps, there are now some extra flashes and flickers that seem to happen when the background of the display changes, although fortunately not when the letter stimuli appear.

I tried uncommenting-out (turning on) this line:

```
Datapixx('SetVideoStereoVesaPhase',0); % Added 2024-02-28 by adc to try  
getting rid of drifting horizontal band
```

Result: band moves around now = worse than before. Will comment the line out again.

From:

<https://wiki.anthonycate.org/> - **Visual Cognitive Neuroscience**

Permanent link:

[https://wiki.anthonycate.org/doku.php?id=resources:grogu\\_setup&rev=1709825299](https://wiki.anthonycate.org/doku.php?id=resources:grogu_setup&rev=1709825299)

Last update: **2024/03/07 10:28**

