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
23.2.1-1ubuntu3.1~22.04.2
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 23.2.1-1ubuntu3.1~22.04.2
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 1024×768 resolution wasn't available to me, let alone the 119.8Hz setting.

RESULTS of 2:

It displayed a setting for 1024×768, 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 3DPixx 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 Datapixx.mexa64, just put the directory holding the correct file at the top of my MATLAB path (`/usr/local/share/VPixx Software Tools/Software Tools/Matlab/mexfiles/linux/matlab/Datapixx.mexa64`). Before this, a copy of `Datapixx.mexa64` from the psychtoolbox-3 subdirectories was being used by MATLAB, and it wouldn't do `Datapixx('Open')`. Now it works.

`ProceduralGarboriumDemo.m` and `VBLSyncTest.m` work great.

Tested 3DPixx with MATLAB PTB

Tried `HolesCrowd_PTB_3StimCrowd_TEST.m` with the VPixx 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:

```
% 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.