TEL/FAX.: (514) 328-7499
TOLL FREE: (844) 488-7499 (USA/CANADA)
EMAIL: support@vpixx.com
www.vpixx.com

VPixx Release Guide

Version 3.9

Table of Contents

Overview

This release guide provides installation and usage information relating to the latest VPixx Software Tool release.

VPixx Technologies Inc. reserves the right to modify or otherwise update this document without notice as required by a constantly evolving marketplace, client requests or to adapt to new progress or constraints in engineering or manufacturing technology. The information contained in this document may change without prior notice.

For more information about our company and products, visit our Web site at ***www.vpixx.com***

For information, comments or suggestions, please contact us by e-mail at ***support@vpixx.com***

Our offices are located at:

630 Clairevue West suite 301
Saint-Bruno, Qc
Canada, J3V 6B4

Document Icons

The use of icons emphasizes helpful, caution or warning notes. Below is a list of the available icons.

TABLE 1 – DOCUMENT ICONS

Icon	Description	Description
	Helpful Hint	Information to help out during assembly, installation or usage
	Caution Notice	Important Information to prevent misuse and/or damage to equipment
	Warning	**Critical information to prevent damage to equipment and/or injury to personnel or participants**

New firmware

DATAPixx3 Revision 20 (was 19):

- DisplayPort stability update

DATAPixx2, PROPixx Controller, VIEWPixx /3D, VIEWPixx Revision 55 (was 53):

- PSync blank line bug fix

PROPixx Revision 49 (was 43):

- New 3D video mode
- New 480 Hz direct input

I1Pro3 Support

Please note that our software tools are only compatible with X-RITE devices purchased directly from VPIXX

The new I1Pro3 spectrophotometer is now supported on all operating systems. It is useable with the same exact functions as the I1Pro2 (the previous version).

I1 Widget in PyPixx
The i1Pro and i1Display widget can now be started in PyPixx without any VPixx devices connected, allowing you to take luminance and chromaticity readings on any screen. The widget can be accessed via the *System* tab:

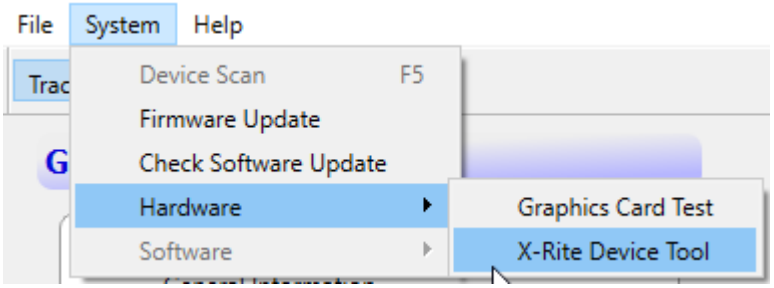


FIGURE 1 X-RITE DEVICE TOOL

I1Pro3 control in VPutil
The I1Pro3 can be controlled in VPutil via the `I13` command. The same options as the I1Pro2 exist, such as `I13 s` for a spectrum reading.

```
Scan of X-Rite USB devices:
>>> i1Pro 3 Plus device detected


-1:(ANY DEVICE) > i13
Measure: Hit 'return' for each measurement, 'q' to quit
(x,y) = (0.322,0.343), L = 175.635 cd/m^2 (51.261 fL)q
-1:(ANY DEVICE) >
```

I1Pro3 in MATLAB

The I1Pro3 can be controlled in MATLAB via the I1 command. This software should be included in the software key with your purchase

```
>> I1
Usage:

% All I1 (Pro, Pro3, Display) functions:
isConnected = I1('IsConnected');
I1('Calibrate');
Lxy = I1('GetTriStimulus');

% I1Pro / I1Pro3 functions:
I1('TriggerMeasurement');
keyPressed = I1('KeyPressed');
spectrum = I1('GetSpectrum');

% I1 Display ONLY functions:
I1('SetIntegrationTime');
time = I1('GetIntegrationTime');
I1('SetMeasurementMode');
mode = I1('GetMeasurementMode');
SERVER MODE VERSION: 3.9 [29/JUN/2021]
>>
```

TRACKPixx Bug Fix

The previous release (version 3.8) included MATLAB files which were not compatible with revision 20 of the DATAPixx3. This has been corrected in this release.

M16G DATAPixx3 Video Mode

M16G is a video mode only available on the DATAPixx3. It is analogous to the M16 available on our devices with certain limitations. When M16G is enabled on the DATAPixx3, your console output behaves the same as the console output of other VPixx devices in M16. Your stimuli display, since it is only an 8-bit monitor, will use the information found in the red channel and display it in grayscale. There are no CLUT for the stimuli monitor. This mode does not include any transparency.

You can find more information on the M16 mode here: https://vpixx.com/vocal/m16/

PROPixx LED Intensity Control

You can now change the LED intensity (sometimes referred to as “brightness”) directly within your MATLAB code. All LED intensities are calibrated to D65 and have a gamma of one. This will only work for normal video modes and will take one second to take effect. This is the equivalent of the PPI command in VPutil, and here is the MATLAB function:

```
>> Datapixx('SetPropixxLedIntensity?')

Usage:

Datapixx('SetPropixxLedIntensity' [, mode=0])

Set PROPixx Intensity of the LEDs.
All these modes are calibrated to have a white point of D65 and a gamma of 1.
Index 5 and 6 are for custom calibration to be done yourself.
mode can take on one of the following values:
   0: 100.00%
   1:  50.00
   2:  25.00
   3:  12.50
   4:   6.25
   5: Custom #1
   6: Custom #2

See also:
>>
```

New TRACKPixx /mini functions

In our pypixxlib python package we have added the following functions to our _libdpx.py:

TPxMiniServerSetupDataRecording(numElem)
TPxMiniGetDataServer(numElem)
TPxMiniRecordData()
TPxMiniStopRecording()
eyeData=TPxMiniGetEyePosition()
status, frameCount, readIndex = TPxMiniGetThreadStatus()

These functions allow you to start a recording of eye-tracking data once the TRACKPixx /mini has been calibrated using either PyPixx or MATLAB. You first need to setup the recording and the number of elements in the buffer using TPxMiniServerSetupDataRecording(numElem) where numElem is the number of elements (with a maximum of 65535). Once setup is complete, you can start recording with TPxMiniRecordData().

You can query the status of the recording with the TPxMiniGetThreadStatus function, which will return three arguments:

- *status*, an integer with a specific definition for the first 3 bits: bit 0 is for read overflow, bit 1 is for thread running status and bit 3 is for write overflow.
- *frameCount* signifies how much data is available
- *readIndex*, which related where we are currently reading in the buffer.

If you need real-time data, you can use TPxMiniGetEyePosition(), or use TPxMiniGetDataServer(numElem) when you want to get more than one recorded data entry in real-time. These functions return either a list of data or one datum which can be parsed as follows:

Index	Value
0	Timestamp
1	Left eye, x position
2	Left eye, y position
3	Left eye, pupil size
4	Right eye, x position
5	Right eye, y position
6	Right eye, pupil size
7	Distance (cm)

Dynamic Libraries (libdpx)

We have updated our software libraries to be fully compatible with the VPixx device server. As well, we have decided to only distribute dynamic library (DLL, SO or DYLIB) files and header files (*.h) in order to keep updating these files more often.
You can find new examples which use the dynamic libraries in the software tools folder. Here are the general instructions to use the CMake file with your program and the libdpx library:
You can use CMake to build your experiment. To do so, you must create a file CMakeLists.txt in your example directory. This file should look like the following:

```
project(DPXDemo)

# This assumes the libdpx directory is directly as a subdirectory of your
# experiment directory. If this is not the case, you should replace
# ${CMAKE_HOME_DIRECTORY} with the actual path to the directory
set(libdpx_DIR ${CMAKE_HOME_DIRECTORY}/libdpx/cmake)
find_package(libdpx REQUIRED)

# DemoAnalog is the name of the experiment. Change as needed
# The other arguments are the source files of your experiments. If more than
# one, they are separated by spaces.
add_executable(DemoAnalog src/dpxdemo_analog.c)
target_link_libraries(DemoAnalog libdpx::libdpx)

# This is just to copy the libdpx dll to the right place
# Replace DemoAnalog with the name you put for your experiment
add_custom_command(TARGET DemoAnalog
    POST_BUILD
    WORKING_DIRECTORY $<TARGET_FILE_DIR:DemoAnalog>
    COMMAND ${CMAKE_COMMAND} -E copy_if_different $<TARGET_FILE:libdpx::libdpx> .
    )
```

This skeleton is in four parts. First, the project command sets the information related to your CMake project. We give it a name (e.g. DPXDemo). You can optionally give it a version and/or a description as follows:

```
project(DPXDemo VERSION 1.0.0 DESCRIPTION "Demonstration of libdpx")
```

In the next step, you want to use the libdpx CMake configuration file. This is done by setting `libdpx_DIR` to the path leading to the CMake directory in the libdpx directory. If the libdpx directory is a subdirectory of your experiment, you do not need to adjust anything. Otherwise, change the part after `libdpx_DIR` to the actual path. The command `find_package` will then retrieve the necessary configurations.

The next step will be to define your experiment. The `add_executable` allows us to create an executable file for our experiment. The first argument is the name you want to give your experiment. The next arguments are the relative paths to the source files. If there is more than one source file, they should be separated by a space. After that, we need to link libdpx to our experiment, by using `target_link_libraries`. Make sure to change the first argument to the name you gave your executable.

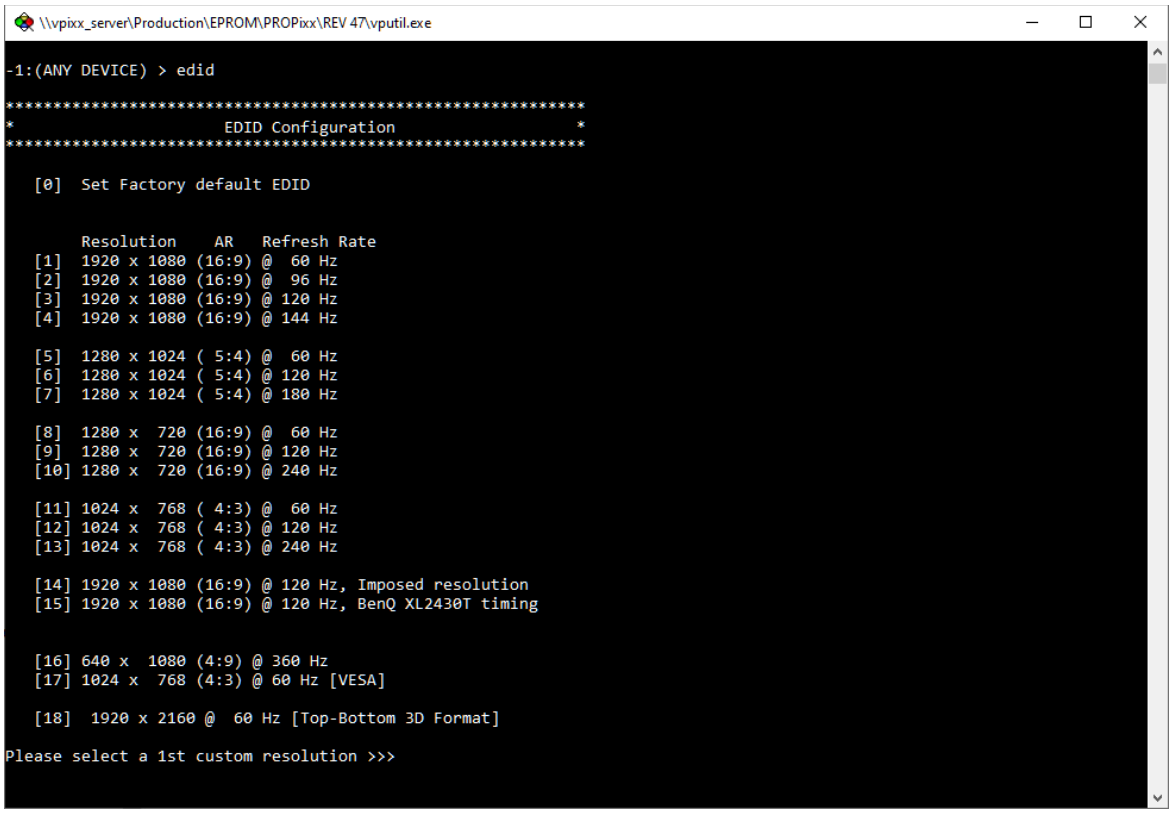
You can build more than one experiment with the same CMake file by duplicating the `add_executable` and `target_link_libraries`, adjusting the name of the executable and the source files for it.

The last part simply copies the library file libdpx to your build directory. This is needed if the library file is not set in your path. You need to change `DemoAnalog` to the name of one of your executables both on the first line of the command and in the `WORKING_DIRECTORY` line.

PROPixx New 3D Mode (TOP/BOTTOM 3D)

The PROPixx is now able to display 3D sent in 1920x2160@60Hz format. The top image would be the first one presented, while the bottom image would be the second.

To enable top-bottom 3D resolution, with the PROPixx Controller turned on and connected via USB, run the edid command, select option 18 twice, and finally *no* for the NVIDIA 3D option:

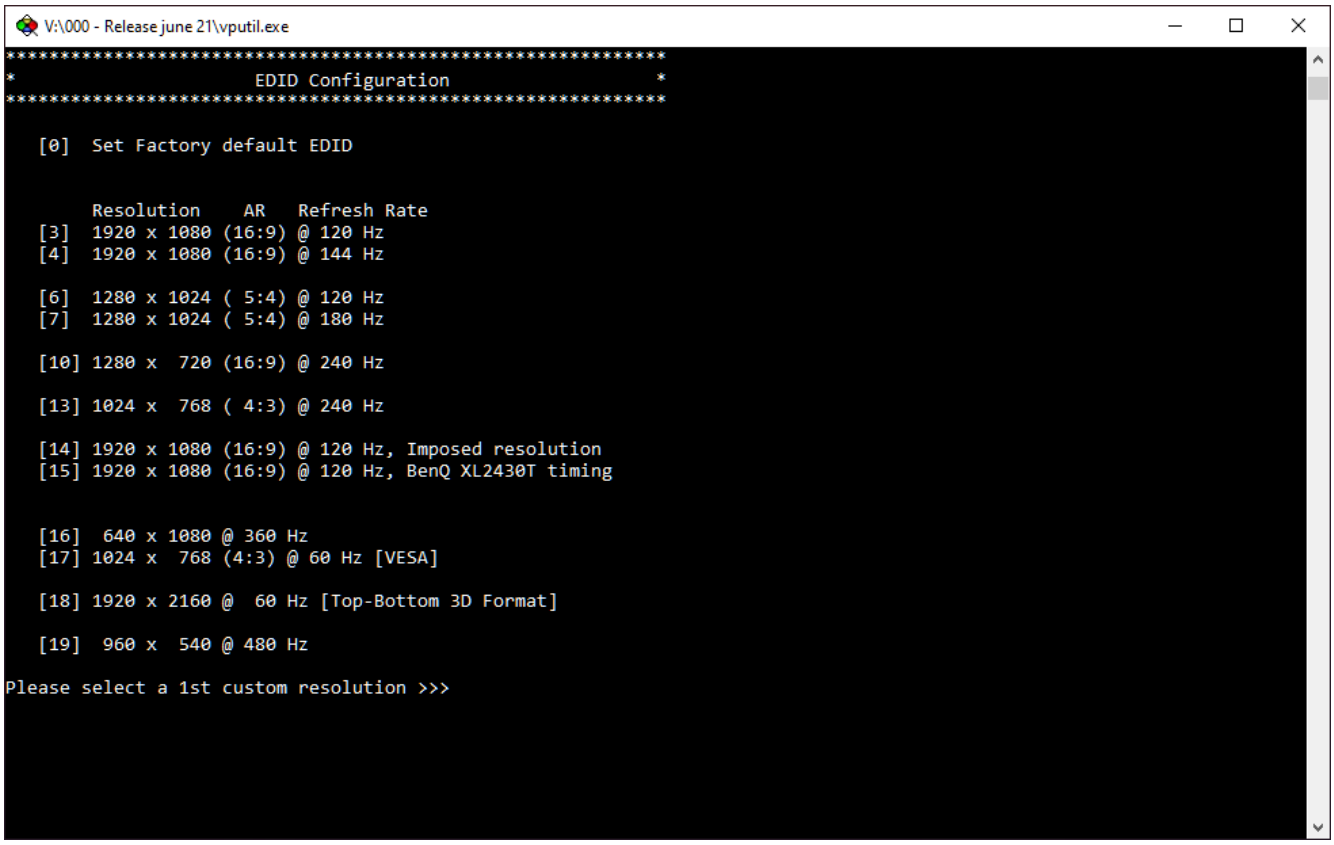


Once the edid command is done, turn off the PROPixx Controller, then turn it back on. You might need to select the 1920x2160 mode, in case the system does not adjust automatically.

PROPixx New Native Resolution

The PROPixx is now able to be driven at 960x540@480Hz directly from the PC.

To enable 960x540@480Hz resolution, with the PROPixx Controller turned on and connected via USB, run the edid command, select option 19 twice, and finally *no* for the NVIDIA 3D option:



Once the edid command is done, turn off the PROPixx Controller, then turn it back on. You might need to select the 960x540@480Hz resolution in case the system does not adjust automatically. When driven with that resolution, the PROPixx switches to this mode automatically.

Simulator Update to 1.1

Stability update

Many fixes have been made on functionalities which may have caused issues in earlier versions. With this latest version, the program seldom needs to be reset. However, if an issue does arise, it might still be needed for you to restart the VPixx Device Server as well as the simulator.

Non-blocking Psync

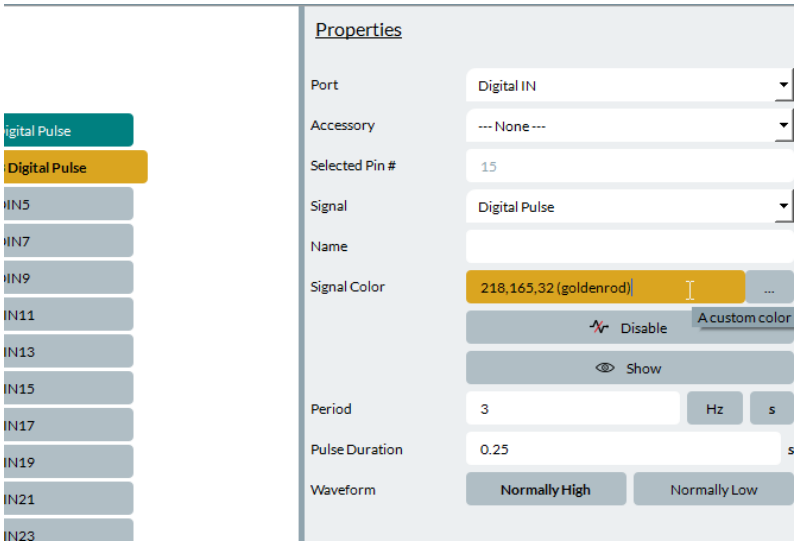
From version 1.0 to 1.1 the behavior of the Psync has been changed to mimic a real device without any change to your code. This means that a device will wait/hang until a Psync is found (first line of the selected display in the option) like a real VPixx device would. This allows you to test your synchronization the same way as you would with your VPixx devices.

Numbers can now accept local separator

This is possible if your Windows is not in English and your locale uses comma to separate decimals without any errors.

Colors can be named

When you select a signal to show in the graph, you can use color names instead of the color picker. You can also use hexadecimal number (#RRGGBB) or RGB values (RRR, GGG, BBB).



REPONSEPixx Hotkey

When you select a RESPONSEPixx to simulate, you can trigger the button presses using the numpad. This is done as follows:

4-5 buttons

Buttons	Shortcut
Red	CTRL + Numpad 6
Yellow	CTRL + Numpad 8
Green	CTRL + Numpad 4
Blue	CTRL + Numpad 2
White	CTRL + Numpad 5

10 buttons

Buttons	Shortcut
Red	CTRL + Numpad 0
Yellow	CTRL + Numpad 1
Green	CTRL + Numpad 2
Blue	CTRL + Numpad 3
White	CTRL + Numpad 4
Red	CTRL + Numpad 5
Yellow	CTRL + Numpad 6
Green	CTRL + Numpad 7
Blue	CTRL + Numpad 8
White	CTRL + Numpad 9

Linux release

The simulator version 1.1 now includes a Linux version with all the same features as Windows. The installation is simple: once you have downloaded our package, you can install it using:

```
sudo dpkg -i vpixx vpixx-software-tools_*.*.***.***-bionic-ubuntu1_amd64.deb
```

Simply replace the * with the version you have downloaded.

If the installation fails, you might need to install dependencies. This is done using the following call:

```
sudo apt --fix-broken install
```

Log values start like a real device

In version 1.0, when you started a digital input log, you needed to empty the buffer yourself as the initial values were being set to the same as a real device. This is no longer the case. Now, when you start the log, the appropriate value will already be set and no false-triggers are generated.