VPixx Release Guide

Version 3.7

Table of Contents

Overview

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

For technical questions or product support information, do not hesitate to contact the VPixx support team by phone or by sending an E-mail to support@vpixx.com

By creating your *MyVPixx* account on the VPixx Technologies website, you will have access to additional product documentation, demos, source code examples and the latest firmware and software drivers.

New firmware

DATAPixx3 Revision 19 (was 14):

- Floating point support for TRACKPixx3
- Video input and output stability update
- New digital output on digital input mode (Double-edged triggers)
- Bug fixes

TRACKPixx3 Revision 18 (was 16):

- Now uses floating points for eye positions
- Eye-tracking edge case stability update
- Corrected issue concerning search limit with one eye visible

DATAPixx2, PROPixx Controller, VIEWPixx /3D, VIEWPixx Revision 53 (was 52):

- New digital output on digital input mode (Double-edged triggers)

PROPixx Revision 43 (was 42):

- 60 Hz input signal behavior changed and optimized (The PROPixx has been designed to run at 120 Hz+).

Code updated to Python3.7

With the phasing out of Python2.7 since the start of 2020, our software has been upgraded to 64-bit Python3.7. This means that the pypixxlib python module will only work with Python3 and require 64-bit installation. This is also true for our other software but should not have any effect once you have updated. For Windows, to ensure that the VPixx device server works, you will need the 2019 C++ redistributable which is automatically installed with the VPixxSetup.exe.

PyPixx improvements

Many of the widgets and utilities available in PyPixx have been revised and reviewed to be simplified and to increase stability. Widgets should now provide dynamic information.

PROPixx and PROPixx Controller

The PROPixx and the PROPixx Controller are now treated as two devices in PyPixx, now being independent of each other. You still need to use a PROPixx Controller to drive a PROPixx projector, but PyPixx will not generate errors if you forget to turn one of the devices on.

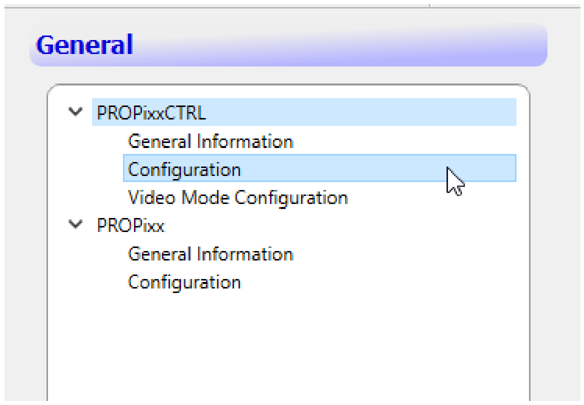

FIGURE 1 SELECTING PROPIXXCTRL - CONFIGURATION

PROPixxCTRL - Configuration allows you to set up both the EDID to change the resolution of the PROPixx projector and the active 3D* if you are using a software without low-level API access or change the behavior of the digital output. You can also select the resolution of the current video being sent to the controller.

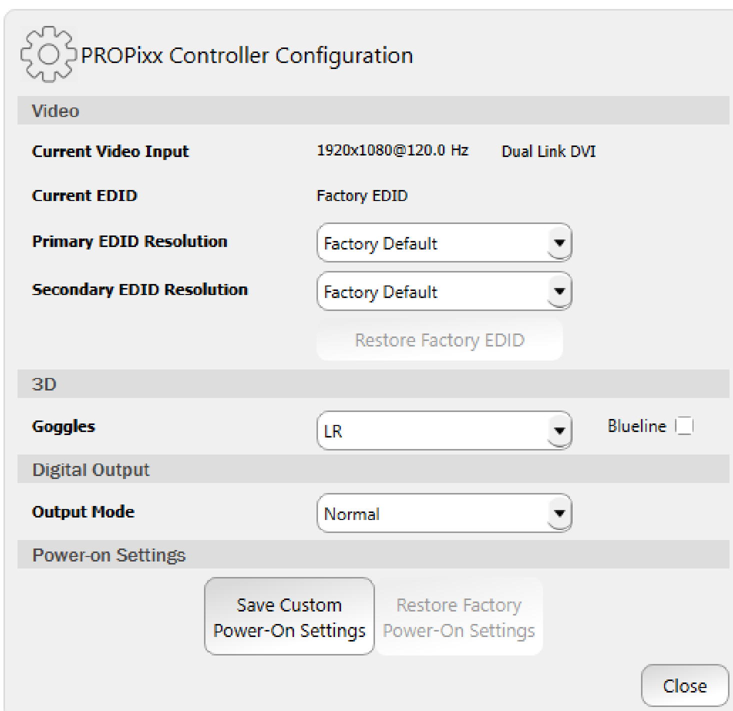

FIGURE 2 PROPIXX CONTROLLER CONFIGURATION

**The PROPixx system usually uses the passive 3D by using a polarizer in front of the PROPixx. The option to enable this is in the PROPixx configuration*

In the **PROPixx Configuration** window, you can set which sequencer the PROPixx currently uses (if it is possible to change it), the current calibration, the intensity of the LED and set up the mode of the PROPixx. You can also enable hotspot correction if it has previously been performed on this PROPixx. The 3D Freerun will enable the polarizer to be driven regardless of the chosen mode.

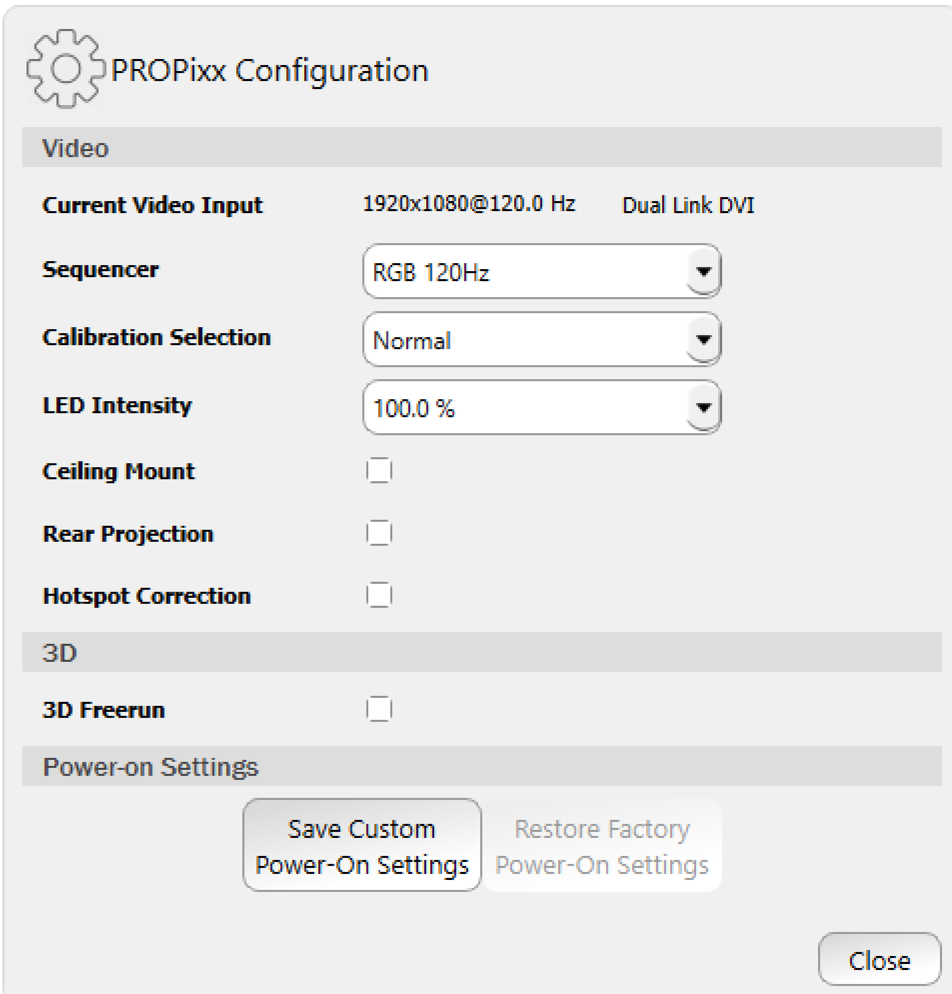

FIGURE 3 PROPIXX CONFIGURATION WINDOW

The **DATAPixx3 Configuration** widget allows you to see the current resolutions used and the name of the monitors. It allows you to configure the 3D goggles or change the output mode:

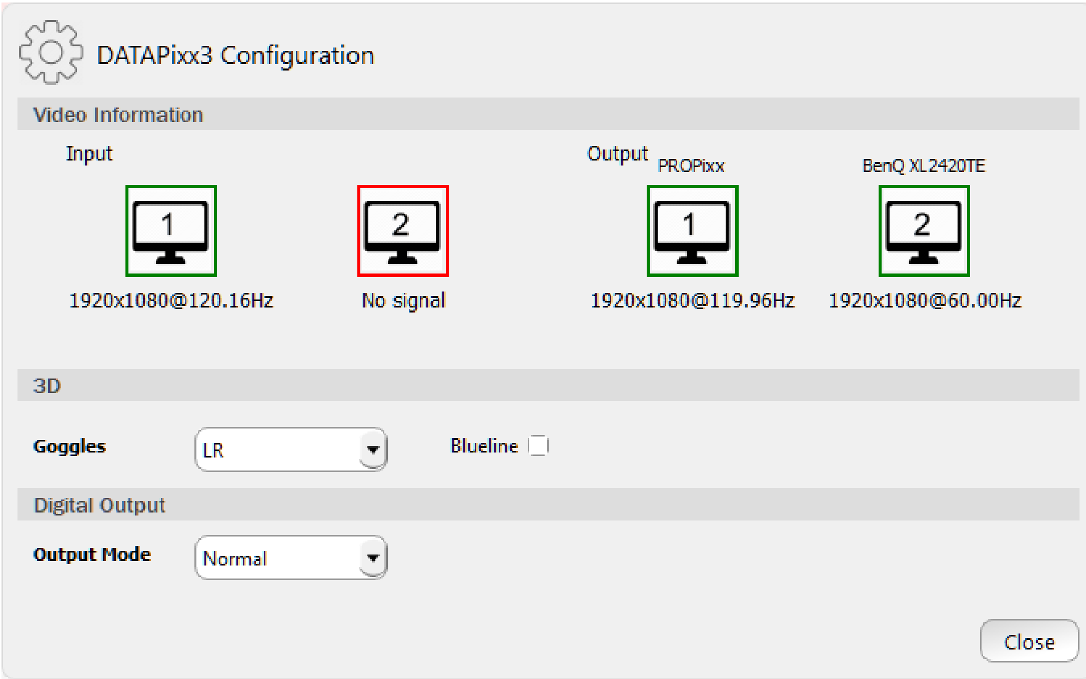

FIGURE 4 DATAPIXX3 CONFIGURATION

The **VIEWPixx Configuration widget** (applies to VIEWPixx /3D and VIEWPixx) allows you to change the 3D mode, adjust the backlight intensity (BLI), enable or disable the scanning backlight and change the output mode.

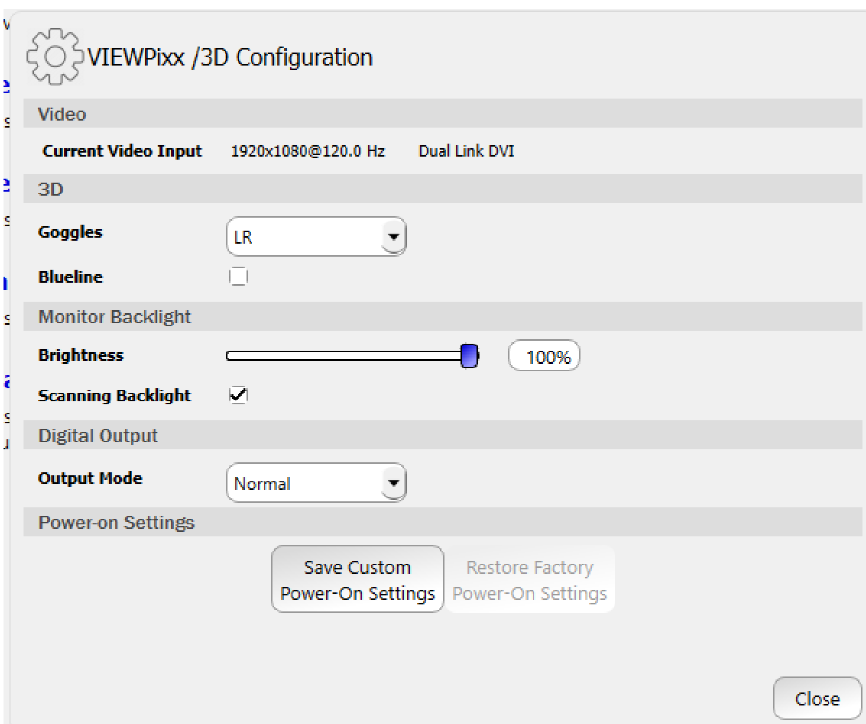

FIGURE 5 VIEWPIXX CONFIGURATION

The **DATAPixx2 Configuration widget** allows you to change the EDID to match the LCD monitors you are driving with the DATAPixx2 as well as the 3D settings and the digital output analog mode.

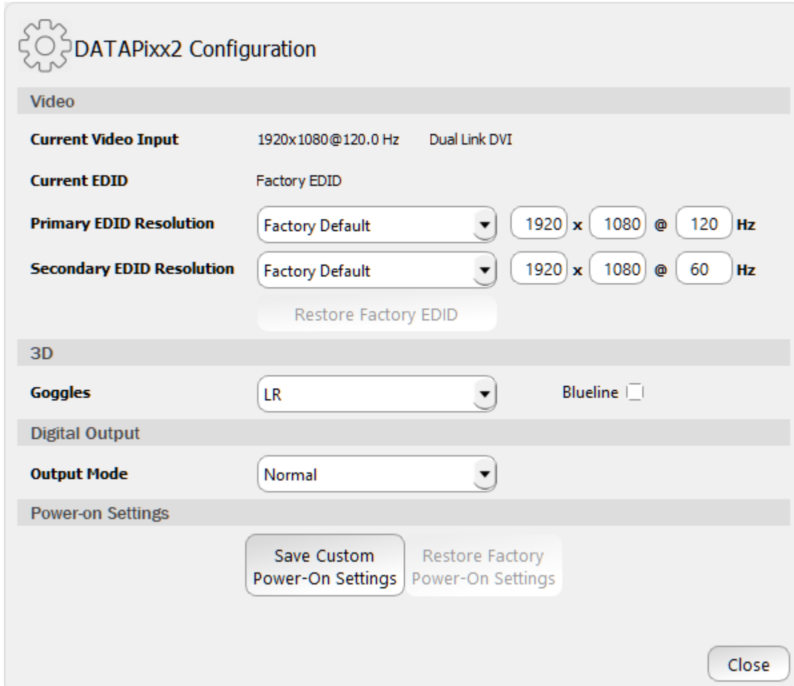

FIGURE 6 DATAPIXX2 CONFIGURATION

The **DATAPixx Configuration widget** allows you to change the EDID to match the CRT monitors you are driving with the DATAPixx as well as the 3D settings.

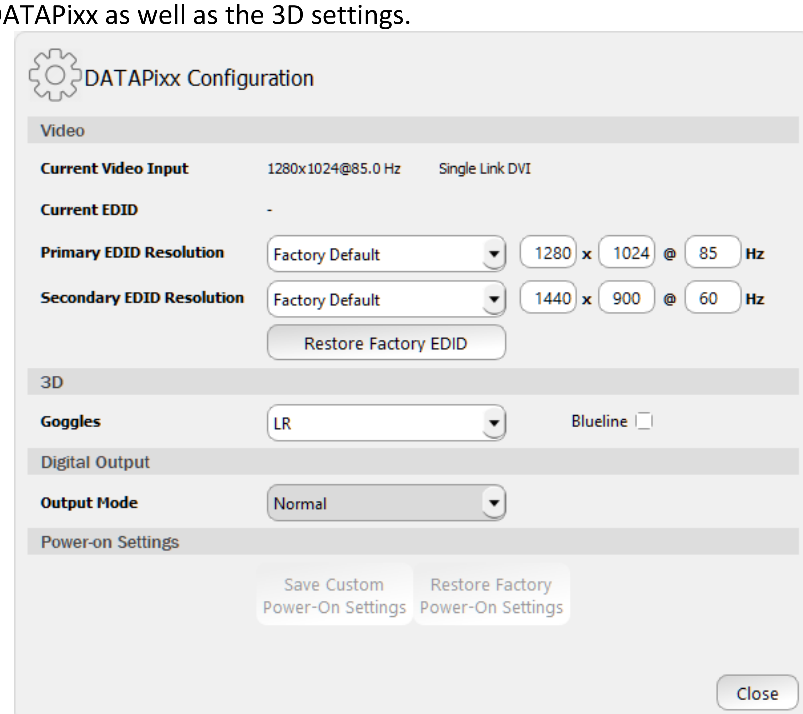

FIGURE 7 DATAPIXX CONFIGURATION

***Note**: Other VPixx devices do not have a configuration widget and are configured either somewhere else (TRACKPixx), or they have no configuration (VIEWPixx /EEG).*

Video mode widget

The video mode widget is now accessible for every device having an available video mode. Please note that if you are using the DATAPixx3 or PROPixx Controller to drive a VIEWPixx or PROPixx, applying a video mode will apply it to both devices.

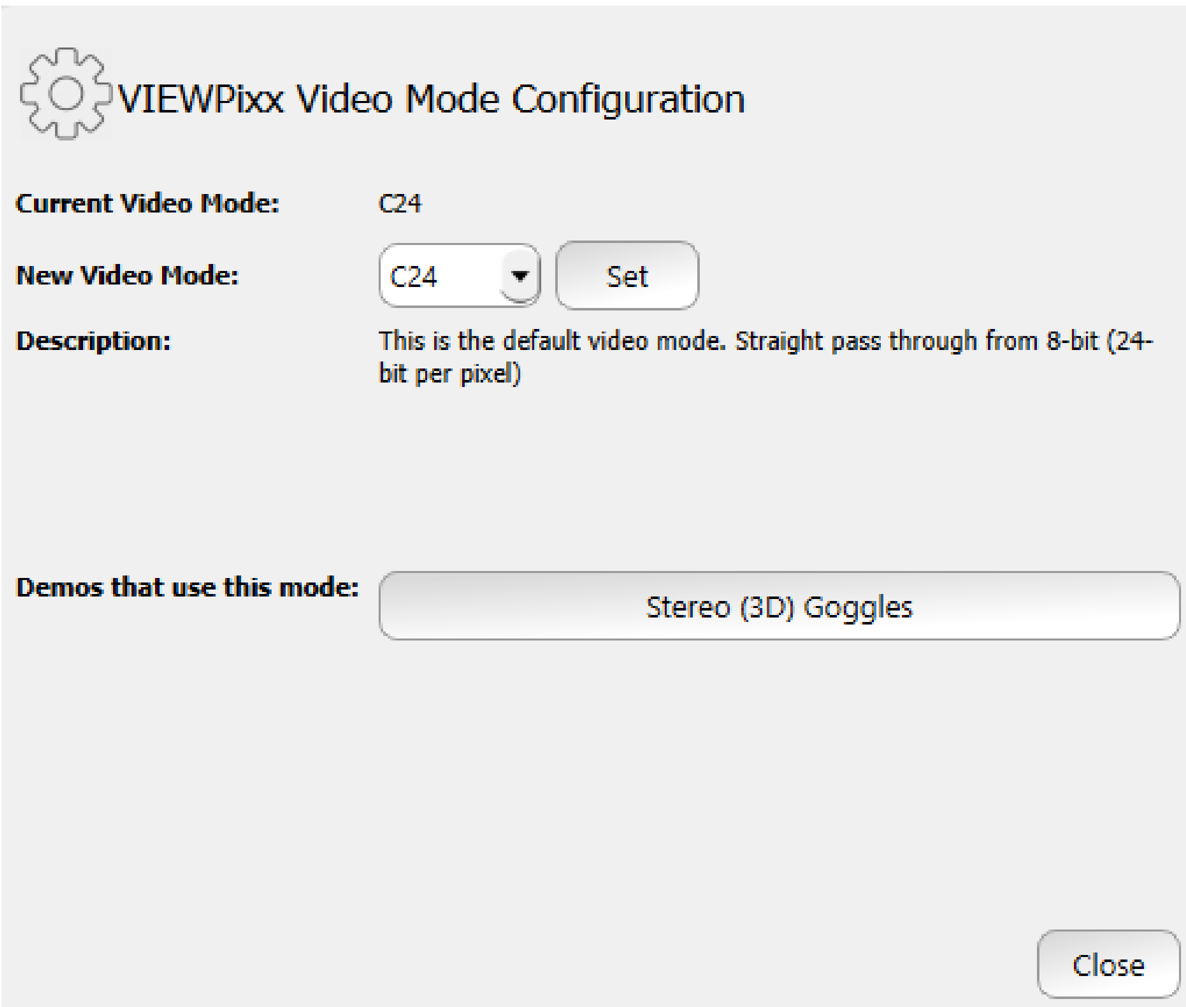

FIGURE 8 VIDEO MODE CONFIGURATION

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 screens. It can be accessed via the System menu:

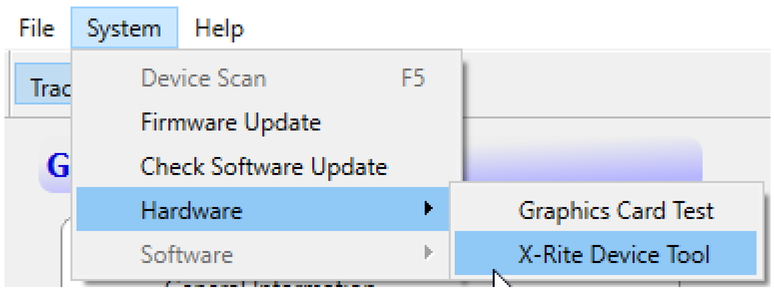

FIGURE 9 X-RITE DEVICE TOLL

VIEWPixx and VIEWPixx /3D Calibration in PyPixx

If you purchased an i1Display colorimeter or an i1Pro spectrophotometer from VPixx Technologies directly, you can now use it to recalibrate your VIEWPixx device directly in PyPixx.

Simply navigate to the Calibration tab and select White Calibration:

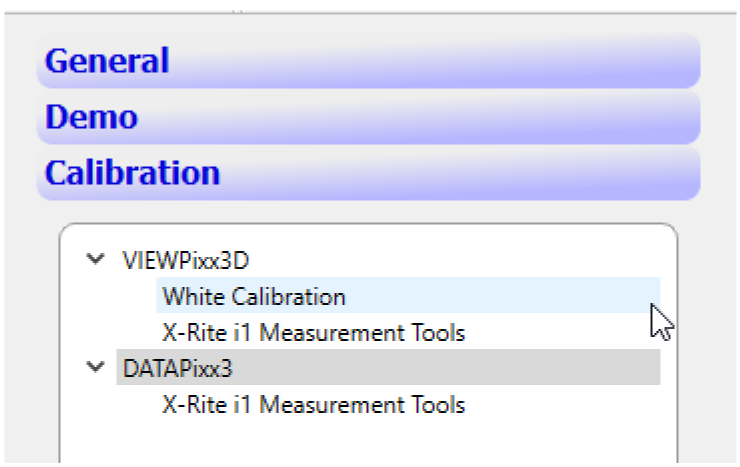

FIGURE 10 WHITE CALIBRATION

This will open the calibration widget which offers you some choices for the calibration:

- Which device to use (this will be detected automatically should you only have one connected)
- What chromaticity your white point should aim for (D65 by default)
- What Luminance you wish to target (in cd/m^2)
- If you started a previous calibration and wish to resume
- Whether to calibrate with the scanning backlight enabled or disabled

Simply follow the instruction on the screen to start the calibration. Once the calibration is started, the VIEWPixx will ignore incoming video therefore if you have a single monitor on your computer, you will lose the GUI. If you are unable to finish the calibration, you can simply restart the VIEWPixx to stop it.

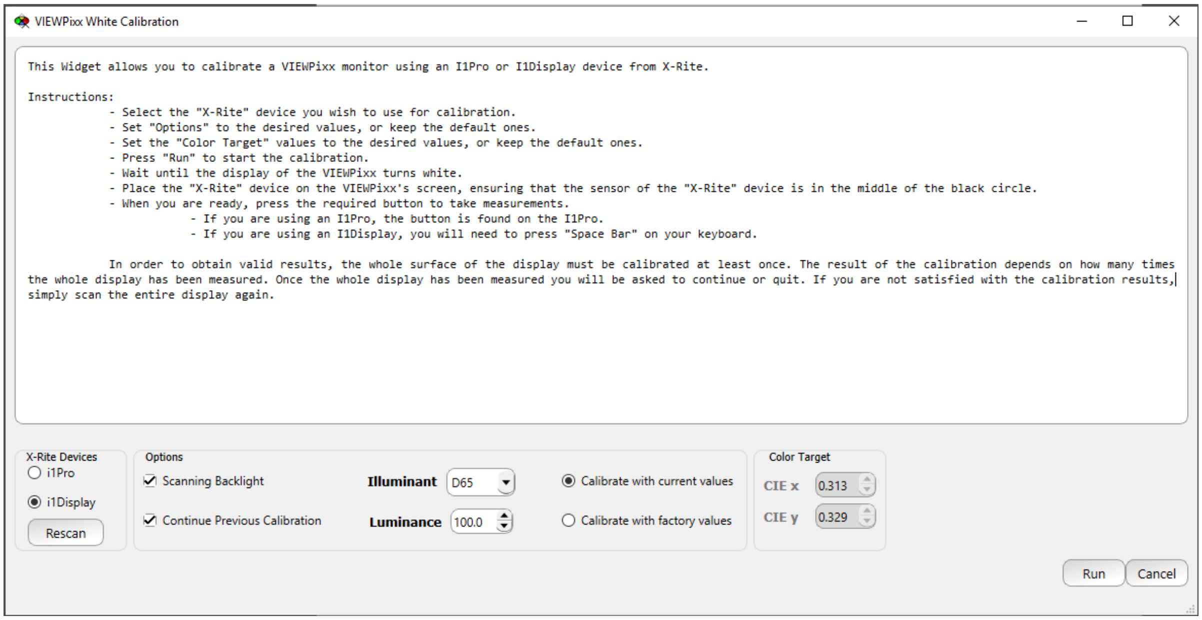

FIGURE 11 WHITE CALIBRATION

New digital output on digital input mode

The digital output on digital input mode, allowing you to send triggers whenever there is an input registered on a VPixx device, now has three different modes:

-Standard: trigger happens on a low edge transition (such as RESPONSEPixx press)
-MRI: trigger happens on a rising edge transition (such as a RESPONSEPixx /MRI)
-Double-edge: trigger happens on both a low edge and rising edge transition.

For more details on this demo, please see the MATLAB Digital I/O Demo 3.

DATAPixx3 Test patterns update

The test patterns available on the DATAPixx3 now support higher than 1920x1080 resolution, such as 2560x1440 or 4K. With this change, all the test patterns can now work on a TRACKPixx3 console display or any higher resolution consumer LCD.

TRACKPixx3 improved schedule

The TRACKPixx3 schedule format has been updated to include much more information and use floating point numbers for higher precision. The schedule behaves the same as before, with two big differences:

- we now send NaN (not a number) whenever tracking has failed (instead of values of *9999*).
- the order of columns has been updated to a more logical order.

<u>These two changes mean that your script must be updated to the new column</u>. As well, if you had any script which removed the blinks by looking for a high number, it must now be made to look for NaN.

Here are the new descriptions for every column:

Column 1: Time tag

Time, in seconds, since the DATAPixx3 was last turned on. Uses the same clock as all other I/O on the DATAPixx3 for easy synchronization.

Column 2: Left Eye X

X screen coordinate, in pixels, corresponding to the calibrated X position of the left eye. Uses a Cartesian system where (0,0) corresponds to the center of the display.

Column 3: Left Eye Y

Y screen coordinate, in pixels, corresponding to the calibrated Y position of the left eye. Uses a Cartesian system where (0,0) corresponds to the center of the display.

Column 4: Left Eye Pupil Diameter

The diameter of the major axis of the left pupil, in pixels. The major axis always reflects the longest axis of the pupil.

Column 5: Right Eye X

X screen coordinate, in pixels, corresponding to the calibrated X position of the right eye. Uses a Cartesian system where (0,0) corresponds to the center of the display.

Column 6: Right Eye Y

Y screen coordinate, in pixels, corresponding to the calibrated Y position of the right eye. Uses a Cartesian system where (0,0) corresponds to the center of the display.

Column 7: Right Eye Pupil Diameter

The diameter of the major axis of the right pupil, in pixels. The major axis always reflects the longest axis of the pupil.

Column 8: Digital Input

An integer value which represents the 24-bit digital input to the DATAPixx3. This value will change in response to button box presses, incoming triggers, and any other input coming in from the Digital In port.

Column 9: Left Eye Blink Detection

0 if the left eye is open, 1 if the left eye is closed

Column 10: Right Eye Blink Detection

0 if the right eye is open, 1 if the right eye is closed

Column 11: Digital Output

An integer value which represents the 24-bit digital output being sent from the DATAPixx3. This value will change in response to outgoing triggers, e.g. from Pixel Mode.

Column 12: Left Eye Fixation Flag

Default 0, changes to 1 if the conditions for a left eye fixation event are met. By default, the fixation flag raises when the eye has moved less than 2500 pixels/second for the last 25 consecutive frames. These default thresholds can be changed by the user.

Column 13: Right Eye Fixation Flag

Default 0, changes to 1 if the conditions for a right eye fixation event are met. By default, the fixation flag raises when the eye has moved less than 2500 pixels/second for the last 25 consecutive frames. These default thresholds can be changed by the user.

Column 14: Left Eye Saccade Flag

Default 0, changes to 1 if the conditions for a left eye saccade are met. By default, the saccade flag raises when the eye has moved more than 10,000 pixels/second for the last 10 consecutive frames. These default thresholds can be changed by the user.

Column 15: Right Eye Saccade Flag

Default 0, changes to 1 if the conditions for a right eye saccade are met. By default, the saccade flag raises when the eye has moved more than 10,000 pixels/second for the last 10 consecutive frames. These default thresholds can be changed by the user.

Column 16: Message code

Placeholder column. Functionality will be implemented in a future release.

Column 17: Left Eye Raw X

X value, in pixels, of the vector between the left eye pupil center point and the left eye corneal reflection point.

Column 18: Left Eye Raw Y

Y value, in pixels, of the vector between the left eye pupil center point and the left eye corneal reflection point.

Column 19: Right Eye Raw X

X value, in pixels, of the vector between the right eye pupil center point and the right eye corneal reflection point.

Column 20: Right Eye Raw Y

Y value, in pixels, of the vector between the right eye pupil center point and the right eye corneal reflection point.

TRACKPixx3 N-point calibration

If you have designed your own calibration process, or if you wish to, you can present any number of points (minimum 3) before you call the method which finishes the calibration and calculates the mapping between raw data and screen coordinates.

TRACKPixx3 analog output control

We have enabled 16 different ways to control each analog output channel (DAC) on the DATAPixx3 with TRACKPixx3 data.

You must first decide what you want to send on every DAC channel, from the following list:

- 0: Default schedule-controlled analog.
- 1: Left eye screen X.
- 2: Right eye screen X.
- 3: Left eye screen Y.
- 4: Right eye screen Y.
- 5: Left eye pupil diameter.
- 6: Right eye pupil diameter.
- 7: Average eye screen X.
- 8: Average eye screen Y.
- 9: Average eye pupil diameter.
- 10: Blink detection flag.
- 11: Left eye raw X.
- 12: Right eye raw X.
- 13: Left eye raw Y.
- 14: Right eye raw Y.
- 15: Left eye saccade & fixation flag.
- 16: Right eye saccade & fixation flag.

Here are some details on each mode:

Mode 0: Default schedule-controlled analog.

This will not be available on a DATAPixx3 Lite, it is the default way of controlling analog outputs.

Mode 1,2,3,4: X,Y Positions

This will report gaze position in pixels with the original being (0,0) at the center of the screen, the X axis directed to the right and the Y axis going up. You can convert from ±5.0 V to pixels by using:

$$GazeX = Voltage \times 819.670 \frac{pixel}{V}$$

$$GazeY = Voltage \times 409.575 \frac{pixel}{V}$$

Mode 5,6: Pupil diameters

This will report the pupil diameter in pixels of camera space. It will go from 0.0V to 5.0V. You can convert between units by using this:

$$PupilSize = Voltage \times 51.190 \frac{pixel}{V}$$

Mode 7,8,9: Average gaze and pupil diameter

This will be the average of both eyes. In the case where one fails to track, the data will be equal to the eye that successfully tracks. In case both eyes fail to track, the resulting fallback voltage will be -5.00 V

Mode 10: Blink detection flag

This mode has 4 different possible values, depending on the state of the blinks:

Data	-2.00 V	0.00 V	2.00 V	4.00 V
Blink	Left	None	Right	Left & Right

Mode 11,12,13,14: Raw eye data

These represent the values of the vector between the center of the pupil and the corneal reflection. This will be in pixel space and the same conversion applies to both X and Y. You can convert from voltage to pixel as follows:

$$Raw = Voltage \times 12.799 \frac{pixel}{V}$$

Mode 15,16: Saccade and Fixation flag

This mode has 3 different possible values, depending on the eye movements:

Data	-2.00 V	0.00 V	2.00 V
Eye Movement	Saccade	In between	Fixation

-

Setup in MATLAB

You can set this up using the new function “EnableTPxAnalogOut” with the correct arguments:

Datapixx('EnableTPxAnalogOut', [DAC0=0, DAC1=0, DAC2=0, DAC3=0]);

With each DAC argument representing one of the modes described above.

You can disable the analog output by calling EnableTPxAnalogOut with all arguments set to 0 or simply by calling Datapixx('DisableTPxAnalogOut');

Setup in Python (pypixxlib)

There is a new function in _libdpx called EnableTPxAnalgOut which is setup with the correct arguments:

```
def EnableTPxAnalogOut(modeDAC0 = 0, modeDAC1 = 0, modeDAC2 = 0, modeDAC3 = 0):
```

With each DAC argument representing one of the modes described above.

You can set the analog output to its default behavior (schedules) by calling:

```
def DisableTPxAnalogOut():
```

There is also the tracker.py class for the object-oriented approach which allows you to:

```
mydevice = Tracker()
mydevice.setAnalogOutState(modeDAC0 | modeDAC1 << 8 | modeDAC2 << 16 | modeDAC3 < 24)
```

Simply replace the modes above with the values you need.

Setup in PyPixx

To set this up in PyPixx, we have created a graphical user interface to select the modes. The modes will also be clearly described in PyPixx. Simply go to Demo tab -> TRACKPixx -> Analog Settings.

In the left column, you can select the options you need and move them to the column on the right.

You can also select some pre-made modes at the bottom right using the available choices.

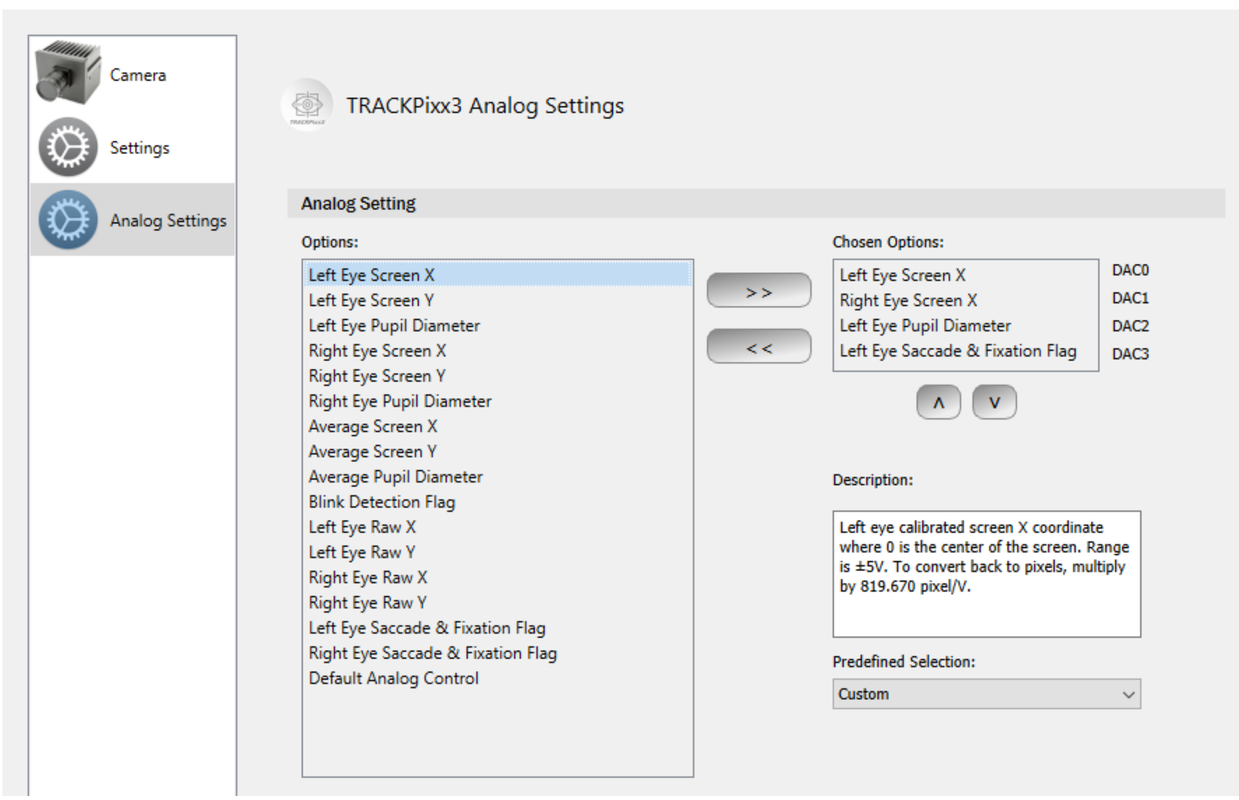

FIGURE 12 ANALOG SETTINGS

The analog option is still present in the previous settings widget. Please note that with this update you might have to select your settings again.

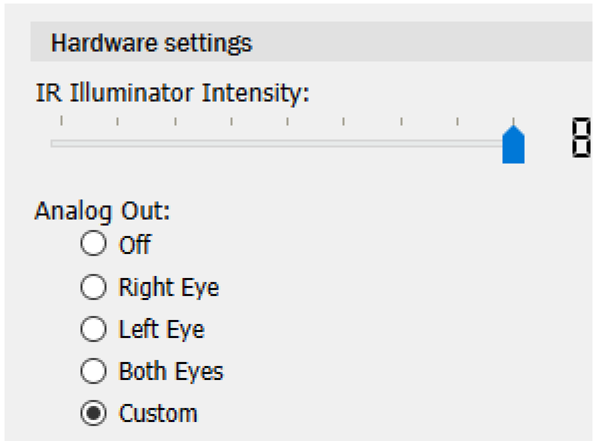

FIGURE 13 HARDWARE SETTINGS

New/Improved MATLAB demos

Our MATLAB documentation has been freshly updated to give a more comprehensive idea of what the demo covers. As well, multiple TRACKPixx3 and PROPixx demos have been added to start users off with eye tracking as well as high refresh rate video projection for the TRACKPixx3 and PROPixx, respectively.

Our MATLAB documentation exists in two formats: A searchable website or a PDF version. It has been optimized for the website version and therefore we suggest you use that instead of the PDF if it is available.

The MATLAB documentation can be found here: https://www.vpixx.com/manuals/psychtoolbox/html/
The PDF can be found in the User Manual folder of the installed documentation.

The new demos or greatly modified demos are listed below.

TRACKPixx3 Demos

TRACKPixx3 Calibration

Devices: TRACKPixx3 and DATAPixx3

Description: This is an improvement on the complete calibration demo previously written. It includes a pupil calibration, a gaze calibration, a validation and a gaze follower.

File TPxCalibrationTesting.m

Demo: TRACKPixx Demo 9

TRACKPixx Saccade to Target

Devices: TRACKPixx3 and DATAPixx3

Description: A quick saccade to target experiment with targets positioned around the middle of the screen. This demo also performs data organization and plotting and may start with a call to the calibration script.

File: TPxSaccadeToTarget.m

Demo: TRACKPixx Demo 7

TRACKPixx3 Saccade to Target using analog output

Devices: TRACKPixx3 and DATAPixx3

Description: A quick saccade to target experiment with targets positioned around the middle of the screen. This demo also performs data organization and plotting and may start with a call to the calibration script. It uses the analog output looped back to gather its data.

File: TPxSaccadeToTargetAnalog.m

Demo: TRACKPixx Demo 8

TRACKPixx3 Yarbus task

Devices: TRACKPixx3 and DATAPixx3

Description: Eye tracking during presentation of a painting (or any image, files required). Each trial is preceded by a different question about the painting. Free viewing data is recorded and plotted over each image. May start with a call to the calibration script.

File: TPxYarbusTask.m

Demo: TRACKPixx Demo 5

TRACKPixx3 Fixation demo

Devices: TRACKPixx3 and DATAPixx3

Description: This simple eye-tracking demo displays a central dot on the screen and records 5 seconds of fixation data. This data is then displayed and saved in a .csv file for further analysis in your software of choice.

File: TPxFixation.m

Demo: TRACKPixx Demo 1

TRACKPixx3 Scotoma Test

Devices: TRACKPixx3 and DATAPixx3

Description: This demo continuously samples eye position from the TRACKPixx3 eye tracker and uses it to update the location of a circular grey mask obscuring part of the displayed image.

File: TPxScotomaDemo.m

Demo: TRACKPixx Demo 3

TRACKPixx3 Pursuit Tracking

Devices: TRACKPixx3 and DATAPixx3

Description: This simple eye-tracking demo displays a dot moving in a circle around the center of the screen. We record 5 seconds of pursuit data. This data is then superimposed over the target path and saved into a .csv file for further analysis in your software of choice.

File: TPxFixation.m

Demo: Demo: TRACKPixx Demo 2

PROPixx Demos

480Hz Draw Dots

Devices: PROPixx Controller or DATAPixx3 and PROPixx

Description: This demo is a simple demonstration of the PROPixx 480 Hz sequencer. We show four random-color dots around the center of the screen, cycling at a rate of 480 Hz.

File: PPxDraw480HzDots.m

Demo: PROPixx Demo 1

1440Hz Draw Dots

Devices: PROPixx Controller or DATAPixx3 and PROPixx

Description: This demo is a simple demonstration of the PROPixx 1440 Hz sequencer. We display twelve white dots around the center of the screen, cycling at a rate of 1440 Hz.

File: PPxDraw1440HzDots.m

Demo: PROPixx Demo 2

<u>1440Hz Draw Dots with blending</u>

Devices: PROPixx Controller or DATAPixx3 and PROPixx

Description: This demo is a simple demonstration of the PROPixx 1440 Hz sequencer. We display a single white dot in the center of the screen, refreshing at 1440 Hz.

File: PPxDraw1440HzDots2.m

Demo: PROPixx Demo 3

<u>1440Hz Draw Dots using Psychtoolbox helper function</u>

Devices: PROPixx Controller or DATAPixx3 and PROPixx

Description: This demo is a simple demonstration of the PROPixx 1440 Hz sequencer. We display a white dot rotating in a circle around the middle of the display, refreshing at 1440 Hz. This demo uses Psychtoolbox's PsychProPixx functions to automate quadrant and color channel assignment.

File: PPxDraw1440HzDots3.m

Demo: PROPixx Demo 4

<u>RESPONSEPixx Demos</u>

Devices: RESPONSEPixx (or any digital input trigger) and any device with digital input

Description: This demo allows the user to create a mapping that can be used in a different file to map the digital input values to the actual button press (color or name).

File: RPxSetButtonConfiguration.m

Demo: Digital I/O Demo 6

New Python demos

As mentioned above, pypixxlib is now compatible with Python3.7. With this update, we also added two new demos for use with the RESPONSEPixx:

RESPONSEPixx Sequential Demo

Devices: RESPONSEPixx (or any digital input trigger) and any device with digital input

Description: Simple demonstration of button labelling and logging using libdpx wrapper

File: buttonPressDemo.py

RESPONSEPixx Object-Oriented Demo

Devices: RESPONSEPixx (or any digital input trigger) and any device with digital input

Description: Same demo as above but instead of using the libdpx wrapper, uses the object-oriented digital input subsystem class.

File: buttonPressDemoOOP.py