VPixx Technologies
Vision Science Solutions

VideoBahn™
The NEW fast road to SUPERIOR data

Application Guide for VPixx Products

Version 1.5

Phone : (514) 328-7499
1 (844) 488-7499 - Toll Free USA/Canada
EMAIL: support@vpixx.com
www.vpixx.com

IMPORTANT

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

Version History of this document

Version Updated to	Date	Author	Reason
0.1	2019/04/10	P.Kakos	V0.1 first draft
0.2	2019/04/12	P.Kakos	Removed chinrest procedure (input from G.Marin)
0.3	2019/05/20	P.Kakos	General update (all sections)
1.0	2019/08/02	P.Kakos	First release
1.1	2019/11/14	P.Kakos	3.5 software update
1.2	2019/11/28	P.Kakos	Removed MATLAB and Python demos, added InfraRed Short-Range 940nm Illuminator
1.3	2020/02/10	P.Kakos	Multiple small changes/updates across many sections (theory and PyPixx mainly)
1.4	2020/02/28	P.Kakos	Added Primer section
1.5	2020/05/14	P.Kakos	Integrated sft rel. v3.7

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
	Applies only to the TRACKPixx3 eye-tracking solutions	When this symbol is present at the start of a paragraph, section or chapter, it indicates that the information is applicable only to the TRACKPixx3 eye-tracking solutions and EXCLUDES the TRACKPixx /mini.
	Applies only to the TRACKPixx /mini eye-tracking solution	When this symbol is present at the start of a paragraph, section or chapter, it indicates that the information is applicable only to the TRACKPixx /mini eye-tracking solution and EXCLUDES the TRACKPixx3 eye-tracking solutions.
	Applies only to the DATAPixx3	When this symbol is present at the start of a paragraph, section or chapter, it indicates that the information is applicable only to the DATAPixx3.
	Applies only to the PROPixx	When this symbol is present at the start of a paragraph, section or chapter, it indicates that the information is applicable only to the PROPixx.
	Applies only to the VIEWPixx	When this symbol is present at the start of a paragraph, section or chapter, it indicates that the information is applicable only to the VIEWPixx.
	Applies only to the VIEWPixx /EEG	When this symbol is present at the start of a paragraph, section or chapter, it indicates that the information is applicable only to the VIEWPixx /EEG.
	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**

Table of Contents

Table of Figures

List of Tables

Introduction

This manual provides safety, setup and operational information for VPixx Technologies Inc.'s vast range of vision research products, which include:

- The DATAPixx family of data acquisition products
- The PROPixx Projector
- The SHIELDPixx Faraday cage for MRI environments
- The SOUNDPixx audio stimulation system
- The TRACKPixx family of eye-tracking solutions
- The VPixx family of InfraRed Illuminators
- The VIEWPixx family of displays

Unless otherwise stated in the text, the terms ''TRACKPixx'', ''DATAPixx'' and ''VIEWPixx'' as used in this document refer to any of these VPixx products:

TRACKPixx:

- TRACKPixx3
- TRACKPixx /mini
- TRACKPixx /MRI

DATAPixx:

- DATAPixx
- DATAPixx2
- DATAPixx3

VIEWPixx:

- VIEWPixx
- VIEWPixx /3D
- VIEWPixx /XL
- VIEWPixx /EEG

Using this manual

This Application Guide for VPixx Products is divided into different application-specific chapters. As well, a Primer on VPixx Technologies Devices and Vision Technology chapter is included which offers useful information for VPixx products and the terms their users must become familiar with, including TTL triggers, registers and so on.

It is strongly suggested to start with the Primer chapter and become familiar with schedules and data acquisition terminology, and then to proceed to the application chapter specific to your product or research.

Technical support

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 (https://vpixx.com/login/),you will have access to additional product documentation, demos, source code examples and the latest firmware and software drivers.

WARNINGS AND SAFETY INFORMATION

This section contains safety and warning information **specific to the TRACKPixx line of products**. Since these products often require the use of infrared emitters, special caution must be taken when using them.

WARNINGS– GENERAL

The TRACKPixx systems are suitable only for research and not for medical applications or for diagnostic purposes.

PHOTOSENSITIVE EPILEPSY (PSE)

Eye-tracking equipment may expose participants to optical stimuli capable of provoking a seizure.

Participants that have never had any epileptic event may have a seizure when subjected to eye-tracking procedures.

Consequently, it is very important to monitor participant reactions and well-being constantly during experiments.

Any modification and/or unauthorized (previously approved in writing by VPIXX TECHNOLOGIES Inc.) or improper use of the equipment contained in this product package may void the warranty and may result in injury to users or damage to equipment. This

includes the opening of electronic equipment and connectors. There are no user serviceable parts inside.

CISPR WARNING: this is a Class A product. In domestic environments, this product may cause radio interference in which case the user may be required to take adequate measures.

Use of controls or procedures other than those specified herein may result in hazardous infrared radiation exposure.

WARNINGS – InfraRed Short-Range Illuminator and InfraRed Short-Range 940nm Illuminator

The 850 nm infrared light from the InfraRed Short-Range Illuminator device (VPX-TRK-9000) and the 940 nm light from the InfraRed Short-Range 940nm Illuminator device (VPX-TRK-9002) is invisible under most viewing conditions.

Participants should not have their eyes within 40 cm (16 inches) of the illuminator.

UNDER NO CIRCUMSTANCE SHOULD A PARTICIPANT HAVE HIS OR HER EYES CLOSER THAN 10 CM (4 INCHES) FROM THE ILLUMINATOR.

WARNINGS – InfraRed Long-Range Illuminator

The 850 nm infrared light from the InfraRed Long-Range Illuminator device (VPX-TRK-9010) is invisible under most viewing conditions.

Participants should not have their eyes within 100 cm (40 inches) of the illuminator.

UNDER NO CIRCUMSTANCE SHOULD A PARTICIPANT HAVE HIS OR HER EYES CLOSER THAN 10 CM (4 INCHES) FROM THE ILLUMINATOR.

SAFETY INFORMATION – InfraRed Short-Range Illuminator and InfraRed Short-Range 940nm Illuminator

As Class 1 LED devices, the InfraRed Short-Range Illuminator and InfraRed Short-Range 940nm Illuminator are compliant with the IEC-62471 LED safety standard, which regulates LED and laser eye and skin safety. Class 1 LED devices are safe under most operational and testing conditions. As the InfraRed Short-Range Illuminator (and 940nm Illuminator) devices may be used in test and laboratory conditions where a participant may be exposed to its infrared emissions for protracted periods, precautions must be taken, mainly to ensure maximum participant comfort.

- The InfraRed Short-Range Illuminator and InfraRed Short-Range 940nm Illuminator devices should not be aimed at a specific participant at distances of less than 40 cm (16 inches) from the eyes. This will ensure an exposure of less than 15 W/m². Exposure decreases as the square of the distance, so even slightly larger distances will reduce exposure significantly.
- UNDER NO CIRCUMSTANCE SHOULD A PARTICIPANT have his or her eyes closer than 10 cm (4 inches) from the InfraRed Short-Range Illuminator or InfraRed Short-Range 940nm Illuminator devices.
- Ensure that the InfraRed Short-Range Illuminator or InfraRed Short-Range 940nm Illuminator device is mounted in an area that allows for sufficient airflow all around it.

SAFETY INFORMATION – InfraRed Long-Range Illuminator

As a Class 1 LED device, the InfraRed Long-Range Illuminator device is compliant with the IEC-62471 LED safety standard, which regulates LED and laser eye and skin safety. Class 1 LED devices are safe under most operational and testing conditions. As the InfraRed Long-Range Illuminator device may be used in test and laboratory conditions where a participant may be exposed to its infrared emissions for protracted periods, precautions must be taken, mainly to ensure maximum participant comfort.

- The InfraRed Long-Range Illuminator device should not be aimed at a specific participant at distances of less than 100 cm (40 inches) from the eyes. This will ensure an exposure of less than 15 W/m². Exposure decreases as the square of the distance, so even slightly larger distances will reduce exposure significantly.
- UNDER NO CIRCUMSTANCE SHOULD A PARTICIPANT have his or her eyes closer than 10 cm (4 inches) from the InfraRed Long-Range Illuminator device.
- Ensure that the InfraRed Long-Range Illuminator device is mounted in an area that allows for sufficient airflow all around it.

Primer on VPixx Technologies Devices and Vision Technology

This primer section offers useful definitions, theory and hints meant to help new clients better understand their VPixx devices and the underlying technical principles governing them. Although it does not delve too deeply into theory, users wishing for more information on a specific subject will find, at the end of every sub-section, a link to more in-depth information found on the VPixx website.

Registers & Schedules

Registers and schedules are core mechanics of VPixx Technologies' microsecond-precise synchronization system. Along with a centralized clock, they allow us to record and align the timing of stimulus presentation, triggers and inputs such as eye tracking or button presses.

While you can absolutely program your experiment without fully mastering these concepts, having basic knowledge about them can help you write more efficient code and take full advantage of the synchronization abilities of your device.

VPixx devices have onboard system settings which include information about the device, its current state and operation. Collectively, these settings are saved in what we call the **device register**.

A copy of these settings is saved on your experiment computer as well. We call this copy the **local register**.

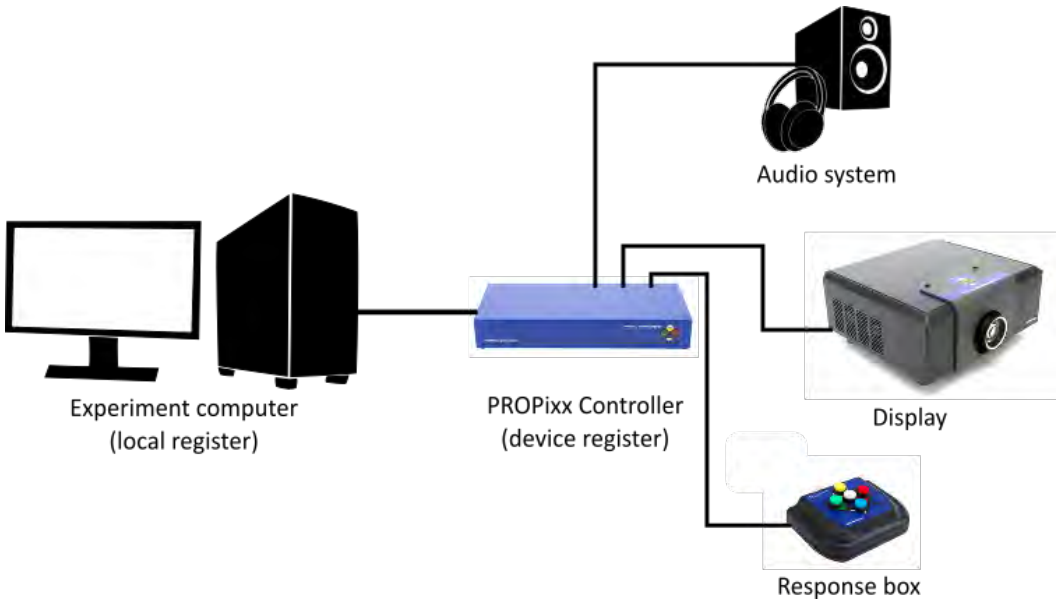


Figure 1 - Example of PROPixx setup with register in PROPixx controller

When a line of code results in a VPixx device taking an action, it means the contents of the *local register* have changed. In other words, we are changing the copy of our device settings that is stored on our computer. This code will generally not alter the state of the *device register*. To update the device register with the changes made to the local register, we must perform a **register write**.

Some cases might warrant the opposite action, and update the local register with whatever is stored on the device. For example, if the device records data, we might want to know how much new data was recorded since our last verification. We may also want to get information on the device status, or the timestamp of a certain event, or any information stored on the device register. In order to update our local register with this information, we must perform a **register read.**

It's important to keep in mind that register write-reads are blocking functions. That is, everything else in your script is put on hold until the read is performed. Other blocking functions include screen flips and listening functions like KBWait(). This is something to be aware of when choosing either a register write or a register write-read.

Having discussed how to communicate between local and device registers to control the settings of our VPixx device, let us now turn to using onboard memory to store and record data. This storage can be configured by defining buffers with a certain address and size. There are several types of buffers, that can be used for different types of data. While several types of data may be stored in different types of buffers, the strategy for managing them is the same. First, if we are writing content to the device for output (e.g., audio files), we must write this content to the appropriate buffer address. Here is an example of setting up an audio buffer with a single tone:

```
%Let's define a 1 s, 500 Hz tone using Psychtoolbox's MakeBeep

duration = 1;
freq = 500;
[tone, samplingRate] = MakeBeep(freq, duration);

Datapixx('InitAudio')
bufferAddress = 16e6;
[~, ~, ~] = Datapixx('WriteAudioBuffer', tone, bufferAddress);
```

If we are using the buffer to record input, e.g., from an eye tracker or button box, we can skip this step.

Next, we define a **schedule** that describes the type of data in the buffer. If we are recording new data, we must also define the address of the buffer in memory. For both input and output, we must set the maximum buffer size.

Other schedule-specific arguments may be required. If, for example, we must set the sampling rate for audio output. the documentation for the `'Set`**`X`**`Schedule'` function lists all required arguments, where X is the type of schedule.

```
bufferSize = duration*samplingRate;
onsetDelay = 0;
Datapixx('SetAudioSchedule', onsetDelay, samplingRate, bufferSize);
Datapixx('SetAudioVolume', [0,0.5]);
```

The 'Set**X**Schedule' and 'Write**X**Buffer' series of commands often supply default buffer addresses and sizes, so it is not necessary to set them yourself. However, if you are running multiple concurrent schedules it is always a good idea to double check that you are not at risk of overlapping buffers, which would overwrite data!

After defining our schedule and settings, we write these to our device register, which implements this configuration.

```
Datapixx('RegWrRd');
```

TTL Triggers

Transistor-Transistor Logic (TTL) triggers are simple, rapid digital signals that can be sent between systems via cable. TTL triggers are recorded by our EEG system, to help characterize our EEG data.

TTL triggers make it possible to insert markers underlining the exact onset of a visual event. Ideally, we want these triggers to be precise, informative and minimally obtrusive. Our data acquisition and synchronization devices can be programmed to send out these TTL pulses, but we have also designed a special triggering mode called **Pixel Mode** which serves a similar purpose as a photodiode, but offers customizable triggers with no extra hardware except for trig cables, minimal onscreen distractions, and no programming required. In the next section, we will cover the basics of how Pixel Mode operates.

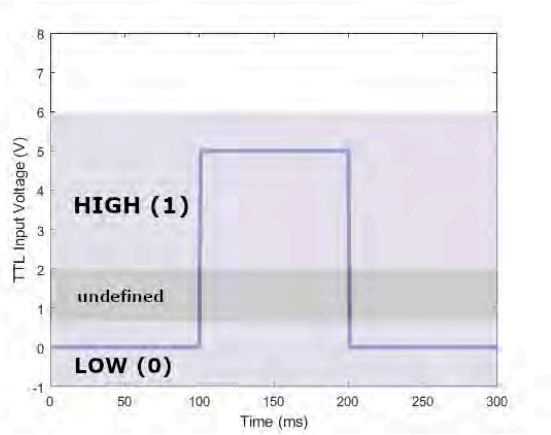


Figure 2 - Example of a 100ms, 1-pin TTL trigger

Pixel Mode

VPixx devices having digital outputs may use **Pixel Mode**, which allows your VPixx device to continuously generate digital outputs based on the color of the top left pixel of the display. This feature was developed specifically for the VIEWPixx/EEG, to allow users to send out triggers when line programming is not an option.

By changing the color of this specific pixel on a given frame during an experiment, researchers can send frame-accurate triggers to their recording system of choice. Pixel Mode can replace and enhance a photodiode-based triggering system—no programming required!

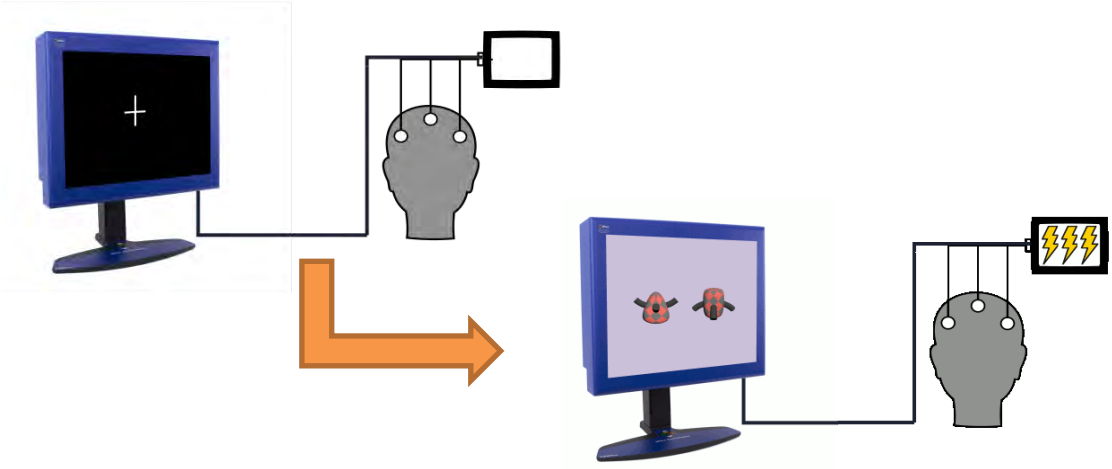


Figure 3 – Pixel mode example

VPixx displays also come with an onboard field-programmable gate array (FPGA) chip which intercepts the video signal after it is sent from the computer GPU. In Pixel Mode, the FPGA generates a trigger based on information in the video signal. This trigger is available on the Digital Output DB-25 connector, and the video signal continues to the display. Here is a simplified schematic illustrating the process:

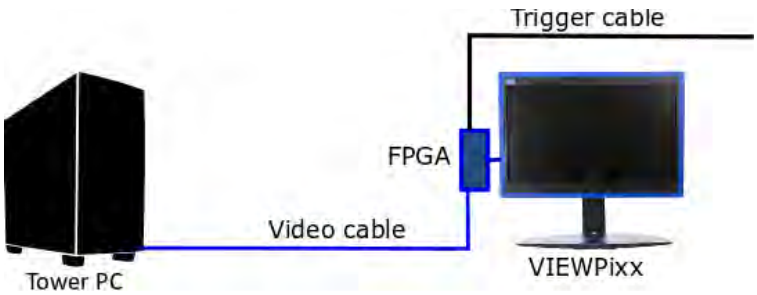


Figure 4 – Temporal offset between trigger and pixel stabilization

By knowing exactly when a TTL trigger is sent, we know the illumination profile of our display and can determine the exact temporal offset between the trigger and stabilization of the pixel in the top left corner of our display. On the VIEWPixx /EEG and VIEWPixx /3D displays, this offset is exactly 6 milliseconds.

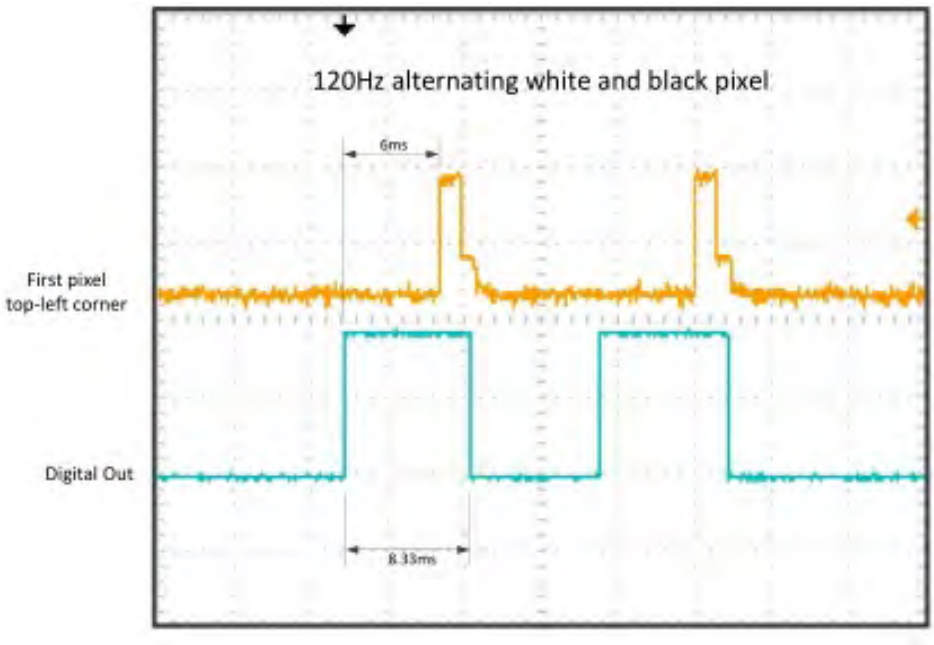


Figure 5 – Temporal offset between trigger and pixel stabilization

VPixx Device Server

The VPixx Device Server allows users to have VPixx devices open and detected using different software applications (Python, MATLAB, etc.). The USB connection for the VPixx devices is established by a single process which allows any other program to detect and communicate with them. This enables you to have PyPixx, MATLAB, VPutil, etc. opened at the same time. You can also have the programs run on a control computer while another computer presents stimuli.

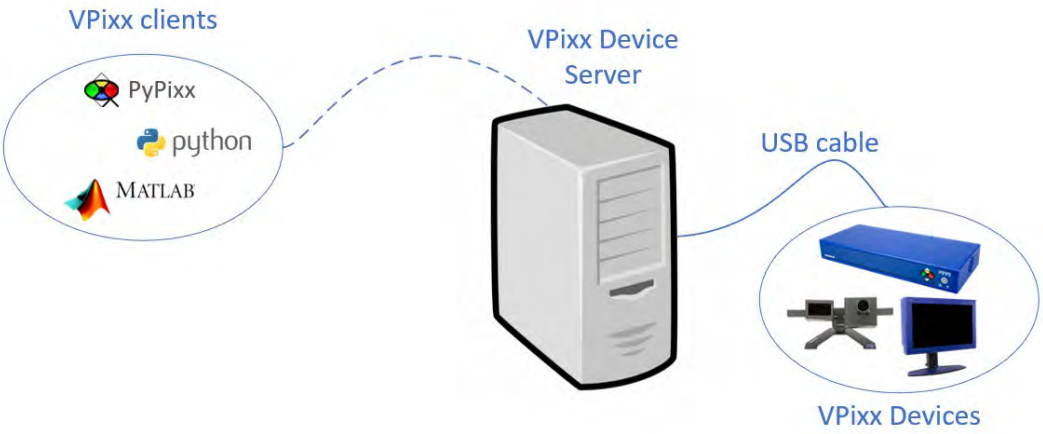


Figure 6 - VPixx Device Server Diagram

The VPixx Device Server uses TCP/IP to transmit its data. This is a very fast protocol used in many similar applications. The delay this causes when compared to accessing the devices directly is in the microsecond range. For timing critical experiments, the server is an approved solution for use with your VPixx devices.

As soon as your computer is turned ON, or when any VPixx devices are turned ON, the server will receive information from your devices and make them available to all programs. This verification is done in a loop and there should be no delay between turning a device ON and being able to use it.

The server is a multi-threaded tool, meaning that any number of queries can be sent simultaneously. However, some critical operations will lock the device such that multiple operations cannot run concurrently. For example, when a firmware update is being completed, no other device can access or modify the device in order to prevent firmware update issues.

Updating your setup to use the server

The fact that the server will be handling all USB transactions means that **any VPixx software tool you were previously using (MATLAB toolbox, PyPixx, VPutil, VPixx Program) must be updated to a version able to access the devices**. To know how to update your specific program, see their appropriate section.

The VPixx Device Server was added to Software release v3.5. If you own a software version older than v3.5, you may require a software/firmware update to fully use the Server.

The server is created such that you should not see any changes in your code after the update is completed. However, should you encounter any hardware issues which would normally be solved by restarting the program you were using, you will need to restart the VPixx Device Server. The procedure to do this depends on the operating system you are using. The next section will cover the installation, starting, stopping and debugging processes for the operating systems we support.

Linux

On Linux, the VPixx Device Server is automatically installed with our package and can be controlled via the 'launchd' command line tool.

Troubleshooting on Linux

If you cannot access your device, or the device is not behaving as it should, you might need to restart the server.

This is done very easily with a simple command line: `sudo systemctl [start/stop/restart] vpixx-device-server`. Simply pick which operation you wish to complete.

If you encounter a problem and would like to get in touch with VPixx, you can retrieve the Server's log by running the command: `sudo journalctl -u vpixx-device-server`. This will generate the file which we can later use to determine the problem's nature.

Windows

On Windows, the VPixx Device Server is automatically installed via our *VPixxSetup.exe* installer. The server is installed as a Windows Service, similar to many other processes that run automatically when your computer starts. This service starts after your computer boots up.

To access the server's status and verify that it is well installed, you must access the 'Services' information tool in Windows. This can be found by searching 'Services':

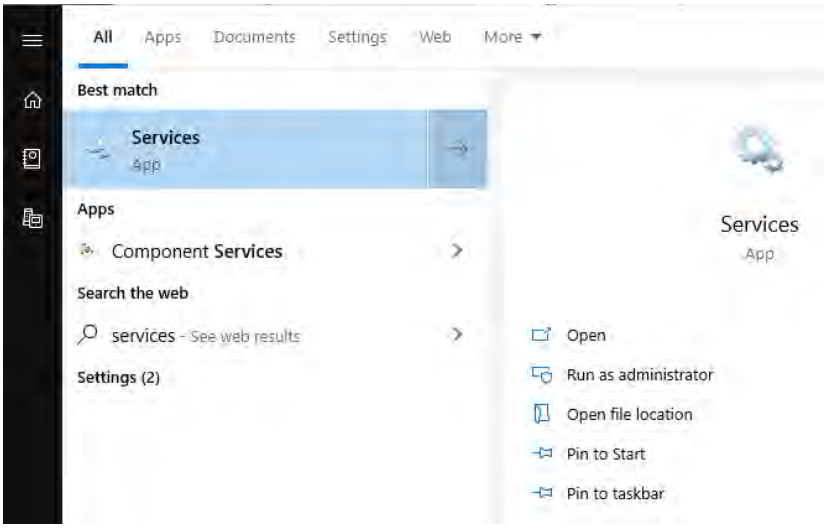


Figure 7 - Services information tool

In the Services window, you can locate the VPixx Device Server. At this location, you can start, stop, restart or uninstall the service should you need to.

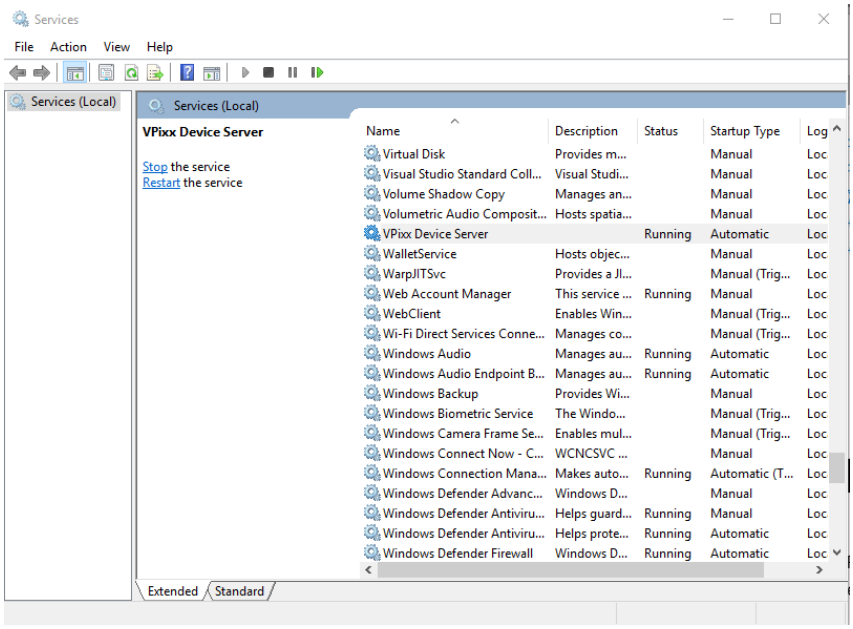


Figure 8 - Locating the VPixx Device Server

Troubleshooting on Windows

After Windows has loaded, the VPixx Device Server should start automatically and while running, you should always be able to access your devices.

If you cannot access your devices, or the devices are not behaving as they should, you might need to restart the server. This can be completed by going to the Services window, right-clicking the VPixx Device Server which allows you to start, stop or restart the server.

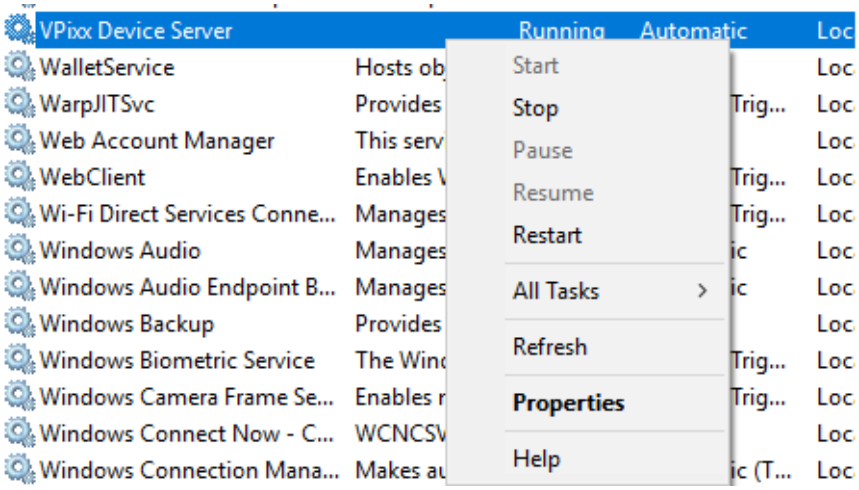


Figure 9 - Right clicking on VPixx Device Server

APIs included with your VPixx product

Most VPixx products include a full set of I/Os and can be accessed through the DATAPixx toolbox in our VPixx Software Tools libraries. VPixx Software Tools include a low-level ANSI C API (Application Programming Interface) as well as Psychtoolbox MATLAB / Octave and Python libraries for macOS, Windows 7, Windows 10 and Linux. In addition, the VPixx devices are directly supported by the VPixx high-level applications (PyPixx and LabMaestro).

With the VPixx Software Tools, you can:

- Acquire data on analog inputs;
- Set up audio playback, feedback and stream;
- Set up voltage outputs, synchronize and generate audio using digital to analog convertors;
- Record, monitor and synchronize with a RESPONSEPixx or any digital signal;
- Set up a Gaze Contingent display using a DATAPixx;
- Show 3D Stimuli with perfect synchronization;
- Display stimuli using high bit depth;
- Record a microphone input;
- Display stimuli at 480 and 1440 Hz using a PROPixx;
- Set up a touchscreen using a TOUCHPixx;
- And much more!

We strongly recommend you create your MyVPixx account by visiting:

http://vpixx.com/register/

By registering, you will have access to the latest software versions, demos and user manuals to support your PROPixx and all of your VPixx products.
This documentation will give you the necessary information to use your Python, MATLAB or Low-Level C toolbox with your VPixx product.

The three APIs and their functions are detailed on the VPixx Technologies website. Simply create an account, login and access these APIs by navigating to MyVPixx→User Manuals→TOOLBOX. Then you can select which API user manual you want to open to explore the various functions.

VPixx Software Installation

Most VPixx products require that you go through the software installation procedure explained in this section. When your VPixx product requires software installation, it will include a USB key containing software packages and toolboxes that must be installed on the experiment PC used to monitor and manage the experiment.

The *VPixx Software Tools* contained on the supplied USB key can also be found on the VPixx Technologies website after logging in to your MyVPixx account and navigating to the ***Downloads&Updates*** page! You may also find information on new software releases in the ***What's New*** section of the website.

Note that some components, such as MATLAB, require that you go through some additional installation steps if they are to be used with Psychtoolbox. For MATLAB, refer to the Installing and Using Psychtoolbox (PTB) with MATLAB section for more information.

These software packages contain the following:

- **VPutil**, a console to communicate with hardware systems and obtain their status (for more information, refer to the ***ANNEX – VPUTIL SOFTWARE*** section.)
- **DATAPixx_Toolbox_trunk**, a MATLAB toolbox for VPixx Technologies products
- The latest **Firmware** for VPixx Technologies products
- The latest **Drivers** for VPixx Technologies products
- **Low Level C API**, the libdpx.dll and its sources and header files
- **PyPixx ,** standalone Python GUI for VPixx products (refer to the PyPixx section)
- **Pypixxlib**, standalone Python module for the management of VPixx products
- Documentation

Table 2 – Software Component Usage

Usage	PyPixx	VPutil	DATAPixx_Toolbox_trunk	pypixxlib	Low Level C API
Building an experiment			✓	✓	✓
GUI interface for configuration	✓				
MATLAB library and DEMO			✓		
Python Library and DEMO				✓	
Low level register access (*)		✓			✓

(*): does not apply to the TRACKPixx /mini application.

Software Installation Instructions for Windows (7, 8, 10)

1. Ensure that your VPixx product is not powered.
2. Insert the USB key into your computer and run the *VPixxSetup.exe* application.
3. When the following box appears, click *Next* and follow the instructions.

Figure 10 - VPixx Tools Setup Wizard (Windows)

Shortcuts to *PyPixx.exe* is copied onto the *Windows desktop.* Documentation, drivers, libraries and low-level C API folders are copied into the *VPixx Technologies* folder.

Following the installation, a VPixx Technologies folder is created on your computer. The default path is:

C:\Program Files\VPixx Technologies

This folder contains the following:

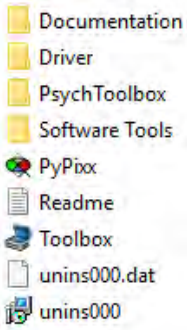


Figure 11 – VPixx Technologies folder contents

Clicking on the *Software Tools* folder gives you access to the following applications and tools.

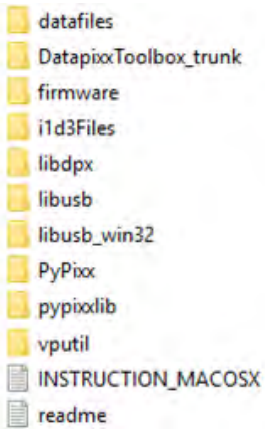


Figure 12 – Software Tools folder contents

Software Installation Instructions for macOS

For OS X, VPixx Software Tools comes packaged as a DMG file which allows you to install our application directly in the app folder (VPutil and PyPixx). It will transfer our files automatically to the appropriate folder. The installation process is simple.

1- Mount the DMG file by double-clicking on the VPixx Software Tools for Mac icon
2- Double click the *VPixx Software Tools.pkg*

Figure 13 - VPixx Tools Setup Wizard (MAC OS X)

You will be guided towards the next steps. If you want to move the documentation files to a different folder, you can do so now. Otherwise, the installer will copy the files to **/Library/Application Support/VPixx Technologies/**

When this installation is completed, one of the scripts will install the server **as a "LaunchDaemon", an application which runs in the background on its own.** Once installed, the server should always be started whenever you restart your computer.

Server status can be controlled via the 'launchd' command line tool.

Software Installation Instructions for Linux

When the VPixx Software Tools installation package (.deb) is installed on Linux, one of the scripts will install the server through the use of a simple command:

```
dkpg -i VPixx_Software_Tools.deb
```

This will install all the required dependencies, the newest firmwares, the newest MATLAB mex file, the server as well as the VPixx Technologies tools PyPixx and VPutil.

The VPixx files can be found in /usr/share/VPixx Software Tools

The server is installed as a Service: an application which runs in the background automatically and managed by SystemD.

Once installed, the server should always start when you power up your computer.

Server status can be controlled via the 'launchd' command line tool.

Installing and Using Psychtoolbox (PTB) with MATLAB

Since all VPixx devices can be controlled by MATLAB, Psychtoolbox (PTB) is not required, but it is nonetheless one of the most used Stimuli generation toolboxes for MATLAB. The VPixx demos are based on Psychtoolbox. In this document you will be able to find instructions on how to install both Psychtoolbox and the DATAPixx toolbox. The DATAPixx toolbox is included in PTB, but we suggest updating to our most recent version. Our toolbox is distributed with the sources; to use it in MATLAB, you will need to load the appropriate mex file. On 64-bit Windows, for example, you will use Datapixx.mexw64, while on 64-bit Mac, you will use Datapixx.mexmaci64.

Installing Psychtoolbox

To install Psychtoolbox, follow the instructions found on the Psychtoolbox website (http://psychtoolbox.org/). During the installation, you will be asked for the directory in which you will install Psychtoolbox. Ensure that you remember that directory.

Updating DATAPixx Toolbox (*mex* files)

Once the installation is complete, please install VPixx Software tools if you have not done so already, and follow these instructions depending on your OS:

Windows:

- Navigate to the VPixx Software Tools (C:\Program Files\VPixx Technologies\Software Tools\DatapixxToolbox_trunk\mexdev\build\matlab), and then navigate to the folder representing your 64-bit MATLAB installation.
- Copy the mex file from that folder to your psychtoolbox_folder\PsychBasic\MatlabWindowsFilesR2007a folder. Note that the name of the folder (MatlabWindowsFilesR2007a) is invariable, no matter the the MATLAB version you are running.

macOS/Linux:

- Navigate to the VPixx Software Tools, then navigate to the folder representing your MATLAB installation (VPixx Software Tools\DatapixxToolbox_trunk\mexdev\build\matlab)
- Copy the mex file from that folder to your psychtoolbox_folder\PsychBasic folder.

For more information on PsychToolbox, please refer to
https://www.vpixx.com/manuals/psychtoolbox/html/index.html

Installing and Using pypixxlib with Python

Installing Python

Python is a programming language (like C) which may be used to manage your vision research experiments using VPixx equipment.

To install Python, simply go to www.Python.org/downloads and download the latest Python version that is compatible with your operating system.

Before users can start using this API, they must ensure that the required material is present. The first step is to verify that Python 3.7 is installed on the computer.

Most of the pypixxlib development has been done on Python 3.7.

Installing our package is user-friendly and done using **pip**. If you do not have pip installed, please follow these instructions:

1. Download **get-pip.py** and run it: `Python get-pip.py`
2. To enable the use of pip from the command line, ensure the Scripts subdirectory of your Python installation is available on the system variable PATH. This is not done automatically. Alternatively, navigate to the Python 27/Scripts folder and install our package: `pip install -U PATH_TO_FILE/pypixxlib.tar.gz`
3. You are now ready to use Python and combine it with VPixx Technologies high performance systems.

pypixxlib

The 64-bit **pypixxlib** package includes a series of classes to define our devices. For each possible device there exists a corresponding class. Each device class has a file with corresponding classes for all the devices (ex: viewpixx.py has the class VIEWPixx3D).

If you create an object that instantiates a class for a device you have connected, all possible functions attached to the object are guaranteed to work and to be applied on the correct device, assuming the operations are permitted.

If you prefer to program at a lower level or if you do not know which device will be used, you can use the _libdpx Wrapper Module, which does not guarantee the functions will work, but does guarantee all the functions will be called on the correct devices if you provide the appropriate commands.

Low-level ANSI C API

Insert the USB key in your computer and browse it. The **"Low Level C API\libdpx\src"** folder in the **Software** folder contains the ANSI C APIs, which are compatible with Windows, macOS and Linux.

*For more information on low level ANSI C API, please refer to the **DATAPixx Toolbox low level ANSI C API user manual** on MyVPixx.*

PyPixx

PyPixx is a Python graphical user interface (GUI) allowing a user to control, test, configure and run any VPixx Technologies device. It can also provide information on the devices, perform calibrations, calibration tests and run demos. PyPixx operates on Windows, Mac and Linux operating systems.

PyPixx can, for example, be used to calibrate the TRACKPixx, display demonstrations of the high-bit-depth stimuli and showcase the high refresh rate of the PROPixx.

PyPixx should be used to configure devices according to your specific needs.

More information can be found on PyPixx in the relevant VPixx product chapters.

Firmware update

The firmware of connected VPixx devices can be updated with one button push in the **Firmware Update** widget which can be accessed from the **System** tab in the PyPixx menu bar.

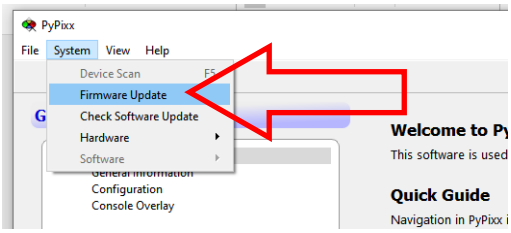


Figure 14 - Accessing the Firmware Update widget

Regardless of how many devices you have connected, you can update all of your devices by simply clicking ‘Install Updates’. The UI also visually informs users of the current progress in detail.

Figure 15 - Firmware Update Widget and Install button

TRACKPixx /mini initialization

The first time you use your TRACKPixx /mini, you must associate your device with a license contained in the provided USB software key.

1. Click on the *PyPixx* icon. This will open the PyPixx application. Click on the *Demo* tab to the left of the screen.

Figure 16 – PyPixx Demo tab

2. In the left window, click on *TRACKPixx /mini*
3. Navigate to the location of the TRACKPixx /mini license, for example E:\TRACKPixx-mini.lic

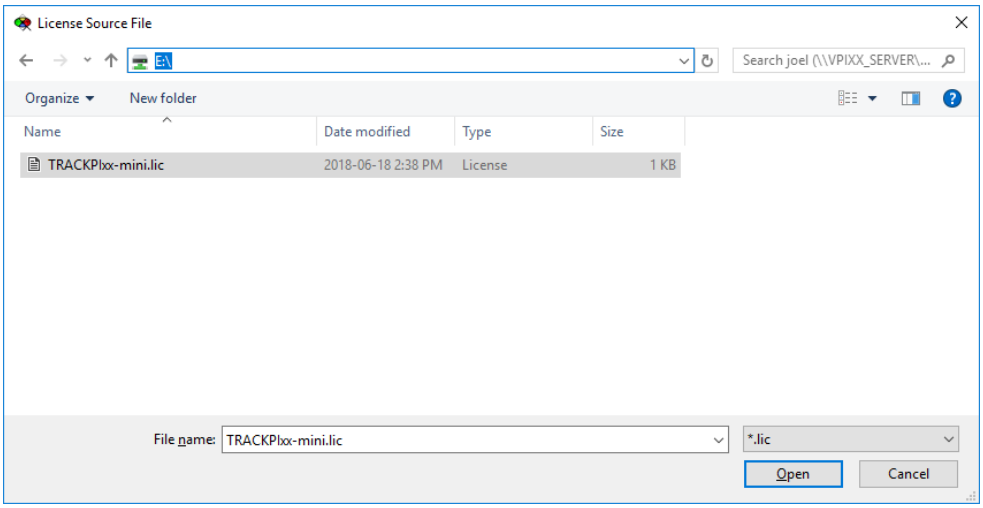


Figure 17 – TRACKPixx /mini license path

4. Click on the license file and then on the **Open** button.
5. Follow the on-screen instructions to complete license installation. This will make it possible to use all of the installed software.

PyPixx on Mac

The PyPixx graphical user interface can be used on Mac OS X. When the VPixx Software Tools installation is completed, you will find PyPixx in your Application folder. You can simply start PyPixx from that folder.

Figure 18 – PyPixx on Mac OS X

PyPixx on Linux

The PyPixx graphical user interface can be used on Linux. Start it using the PyPixx command in a terminal.

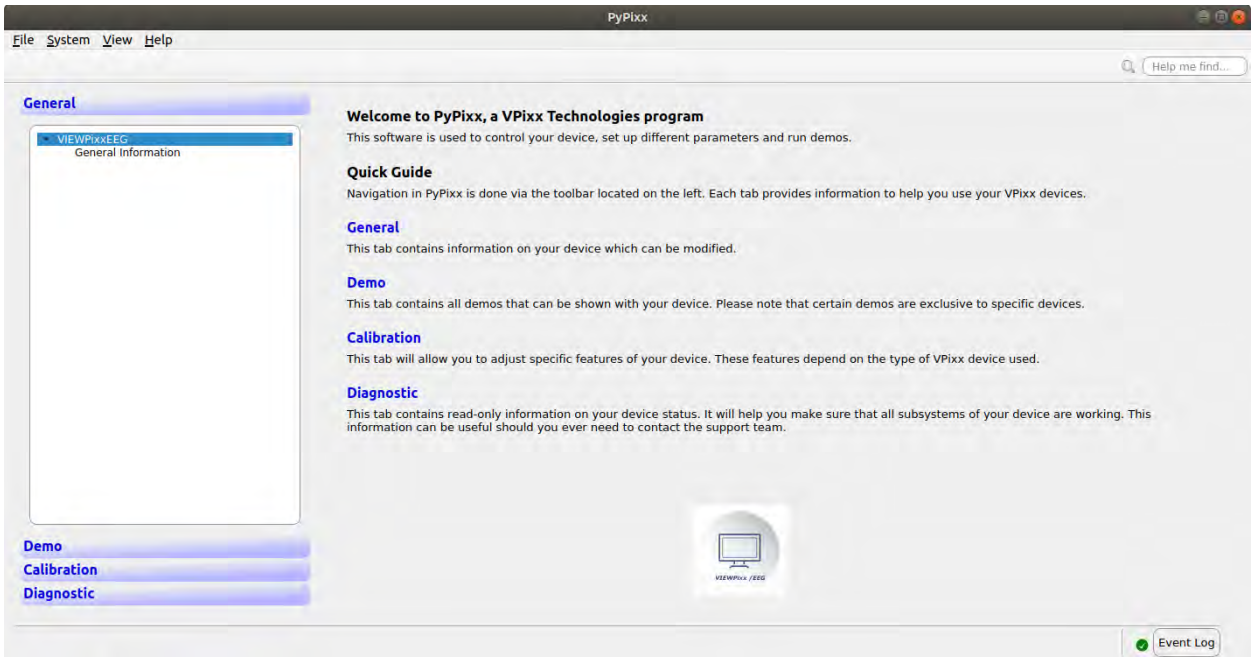


Figure 19 – PyPixx on Linux

Adding your USB I/O HUB device in E-Prime (E-STUDIO)

The present section gives you the procedure necessary to properly integrate the USB I/O Hub supplied with your TRACKPixx /mini using E-Prime.

The present procedure assumes you are using E-STUDIO 3, a software product part of the E-Prime suite. Note that the same procedure can be used in E-STUDIO 2.

1. When a new experiment is started, double click on the **Experiment** box located in the Experiment Explorer panel.

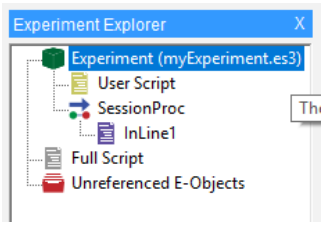


Figure 20 – Experiment symbol

2. This will display a pop-up window containing the properties for the experiment. Select **Devices**.

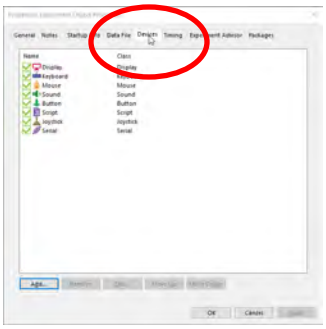


Figure 21 – Devices tab

3. The USB I/O Hub contains two devices: a **Joystick** and a **Serial** port. We use the *Add* button to insert them both into the experiment, as shown in the following figures.

Figure 22 – Joystick device icon

4. Enter the device properties, including the name of the device.

Please note that the name of the device can be arbitrary but must correspond to the object used in the script. In this example, we name the joystick device *MyJoystick*. Therefore, in the code should be a line containing: *MyJoystick.GetPosition.Buttons*. A similar naming process is necessary for the serial port.

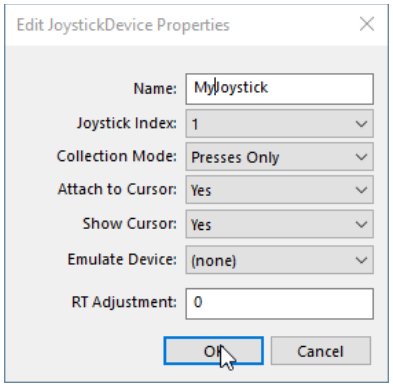


Figure 23 – Naming the joystick

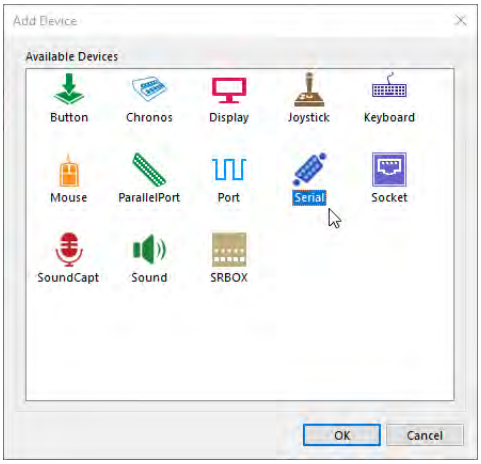


Figure 24 – Serial device icon

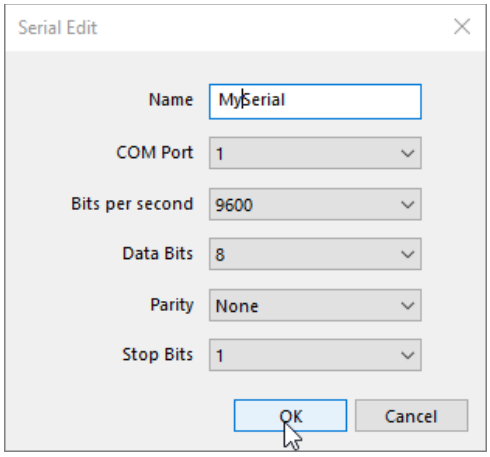


Figure 25 – Naming the Serial device

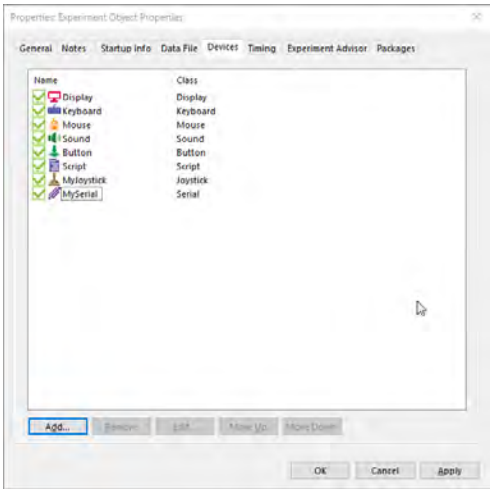


Figure 26 – USB I/O Hub devices

Eye Tracking

Theory

Eye tracking is defined as the measurement of eye rotation and may include, when coupled with absolute pupil position data, gaze and point of gaze information. An eye-tracking solution is therefore defined as any device or system capable of measuring eye (pupil, corneal reflex, or other ocular landmarks) position precisely. Such technology finds theoretical/medical/commercial use in the fields of psychology, psycholinguistics, vision science and neuroscience as well as marketing, gaming and various other commercial applications. The TRACKPixx eye-tracking solution can measure the following oculomotor behavior:

Fixations

Fixations occur when a participant foveates a specific point in space. Fixations generally include micromovements of the eyes to keep the subject of interest on the fovea.

Saccades

Saccades are rapid, ballistic eye movements that occur when a participant changes fixation. Saccades are characterized by a rapid "jump" of the eyes from one position to another. Saccades may be made to fixate on a specific stimulus or location. Antisaccades are made away from a specified direction or target. A series of saccades may be used to scan the contents of a visual scene.

Smooth Pursuit

Smooth pursuit occurs when a participant moves their eyes to continuously foveate a moving target. In contrast to saccades, during smooth pursuit the eye's position undergoes online correction to maintain tracking of the target.

Gaze path

Video-based eye-tracking technology invariably measures movement of the pupil. When combined with relative position of the corneal reflection, rotation of the eye can be extracted. Combined with precise knowledge of the eye's absolute position in a frame of reference, gaze path (line of sight) information can be determined.

Eye-tracking research applications

The applications for eye-tracking technology grow yearly.

1. **General research** in vision and neuroscience.
2. For **medical research**, medical professionals can better assess ocular/neurological disease, trauma and conditions (including autism) using the latest eye-tracking solutions.
3. Vision research leads to advancements in **reading techniques** for the general population and vision-impaired participants.
4. In **marketing**, eye-tracking technology helps companies determine which marketing strategies to promote based on a target population's eye reaction to various commercial ad/branding approaches.

5. Eye tracking has applications in the **transportation sector**, from airport security to road safety and intelligent automobile technology.

VPixx Eye-tracking technology

VPixx Technologies' **TRACKPixx** line of products affords vision researchers a range of eye-tracking products with which to advance their research. All TRACKPixx products offer customization potential, versatility and a large tracking range while the TRACKPixx3 camera offers zoom and focus adjustment allowing for greater configuration options. Consequently, the TRACKPixx3 camera can be adjusted to function in most laboratory environments.

The TRACKPixx3 is the first eye tracker in the world to have a pupil size calibration feature. Scientific papers point to the flaws in reported gaze positions given by video-based tracker systems depending on pupillary responses. With this pupil size calibration feature, the gaze position is corrected to reflect pupil constriction or dilation.

TRACKPixx systems do not require a dedicated and separate computer to process gaze information; the devices handle the data internally. Eye positions are transmitted either through a digital or analog signal, logged in the device for future access or stored in an external device through a USB interface (Windows, Mac, Linux).

The TRACKPixx3 video feed can be accessed directly through a console display for real-time visualization and adjustment of the tracker. A scene camera may also be connected to the DATAPixx3 DisplayPort interface to monitor the experiment. These video feeds can also be accessed through a USB interface for remote control of the TRACKPixx system.

The TRACKPixx3 systems also include all DATAPixx3 subsystems (digital, analog and audio I/Os) which can be used to synchronize and obtain microsecond-accurate timetags for responses or triggers. New eye positions are computed every 0.5 milliseconds; combined with a PROPixx projector, this is a perfect and optimal combination for a gaze-contingent display.

All TRACKPixx systems generally use the same software support applications. Only the camera, lenses and supporting hardware vary according to the specific system and application. All systems produce eye-tracking data files using the same format so that data obtained using one TRACKPixx system is comparable to data obtained using a different TRACKPixx system.

Principle of operation

All VPixx eye-tracking systems make use of the pupil and corneal reflection ocular landmarks. A lamp illuminator bathes the participant's face and eyes with infrared light. The TRACKPixx3 uses a CMOS image sensor to generate a 2 kHz video stream of the participant's face. The corneal reflection can be recognized as a region much brighter than the rest of the face, whereas the pupil is recognized as a region much darker than the rest of the face.

The camera's algorithm calculates the centers of pupil and corneal reflection at a rate of 2 kHz (2000 measurements per second). It also determines the best fitting elliptical shape of the pupil.

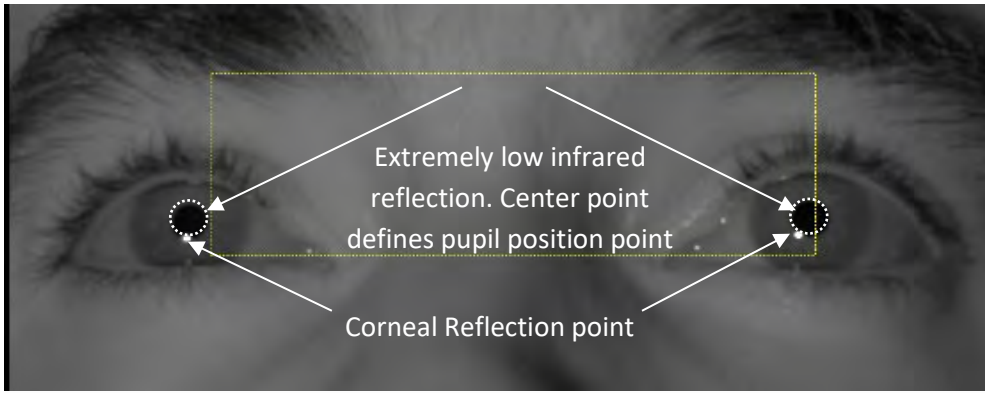


Figure 27 – Pupil position and corneal reflection points

Once both the pupil center and corneal reflection points have been determined, the camera determines the **Pupil-Corneal Vector** necessary to calculate the participant's **gaze point** and, ultimately, overall **gaze path**.

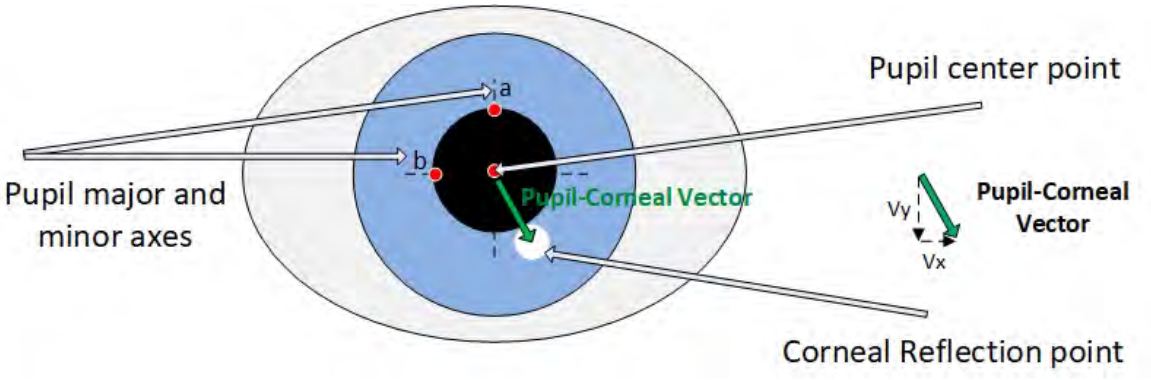


Figure 28 – TRACKPixx Detection points

When combined with the data collected during the preliminary calibration step, the Pupil-Corneal vector information (Vx and Vy, for both eyes) allows the TRACKPixx system's algorithm to determine the gaze point for each of the participant's eyes.

To extract gaze point information and generate fixation points and heat maps, the participant's head position must be precisely known. For TRACKPixx3 systems, this is possible using a **chinrest** (an equipment option that may be purchased with your TRACKPixx3 product).

Figure 29 – Heat Map

TRACKPixx3 systems send all the measured data points to the DATAPixx3 for analysis and processing using the fiber-optic VideoBahn interface. The DATAPixx3 then sends the data via USB cable to the test facility's computer where the researcher will manage the test case accordingly. The TRACKPixx /mini is an entirely self-contained product, and does not require the use of a DATAPixx3.

No separate data processing computer or tracker computer is necessary. The DATAPixx3 handles all the data processing/synchronization chores necessary to run a multi-equipment test session as precisely and efficiently as possible, allowing for perfect synchronisation between any number of VPixx products connected to the DATAPixx3. This ensures that the timestamps associated with the data collected by the TRACKPixx3 system are perfectly matched to the timestamps of any other measurement points /data processed by VPixx equipment also connected to the DATAPixx*3*.

The lab researcher can monitor the experiment using the **console monitor**, which is included with all TRACKPixx products except the TRACKPixx /mini. Connected to the DATAPixx3, the console monitor allows the researcher to observe the information displayed on the stimulus monitor alongside the TRACKPixx camera feed. The researcher can therefore verify that the TRACKPixx camera properly tracks the participant's eyes and if needed, effect real-time adjustments.

Figure 30 – Console Monitor

To allow our tracker systems to have online saccade and fixation detection, we use a total of 9 samples to detect these events. Velocity is calculated using 4 past frames and 4 future frames compared to the current eye position. This allows us to remove any possible error to the eye tracking experiment coming from external factors, such as camera noise.

We compare the calculated velocity (which is the vector created from the speed in both X and Y direction) to certain thresholds for saccade and fixation and if the value is under or above these thresholds for a large enough number of consecutive samples, we declare that a fixation or saccade is occurring. The currently used thresholds are as follows:

1. For a sample to be flagged as a **fixation**, the velocity of the eye movement must be under 2500 pixels per second for 25 consecutive samples.

2. For a sample to be flagged as a **saccade**, the velocity of the eye movement must be over 10 000 pixels per second for 10 consecutive samples.

These thresholds can be changed in PyPixx and MATLAB.

TRACKPixx3 Short-Range and Long-Range Systems

The TRACKPixx3 short- and long-range systems are versatile 2 kHz eye/gaze-tracking solutions supporting both monocular and binocular tracking with a single mechanical configuration. Interchangeable lenses support tracker distances from 24'' to over 63'' (60 cm to over 160 cm) depending on your application needs.

The TRACKPixx3 does not require a dedicated PC to process eye images and generate gaze information; all image processing is performed within the TRACKPixx3 and DATAPixx3 hardware.

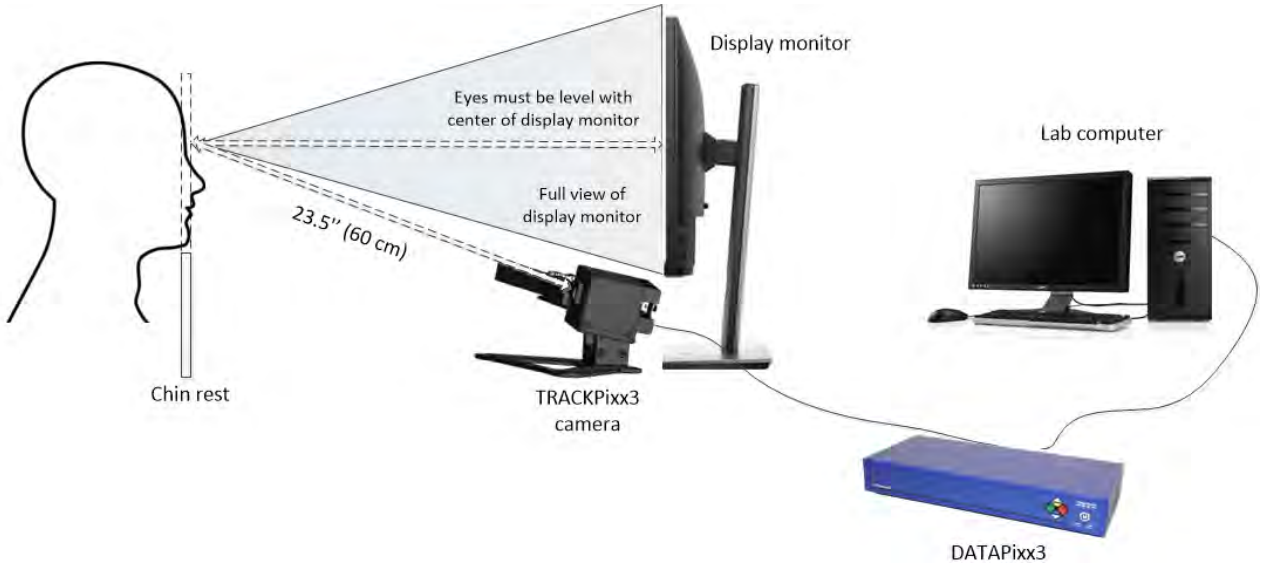


Figure 31 – Typical TRACKPixx3 Short-Range System Setup

This high-level figure only shows general connectivity setup and arrangement. For complete cabling and hardware installation instructions for your TRACKPixx3 Short- or Long-Range system, please refer to the **TRACKPixx3 Installation Guide** supplied with your product.

TRACKPixx3 MRI

The TRACKPixx3 MRI is a versatile eye/gaze-tracking solution. It is specifically-designed to be used for MRI experiments.

The system does not require a dedicated PC to process eye images and generate gaze information; all image processing is performed within the TRACKPixx3 MRI hardware.

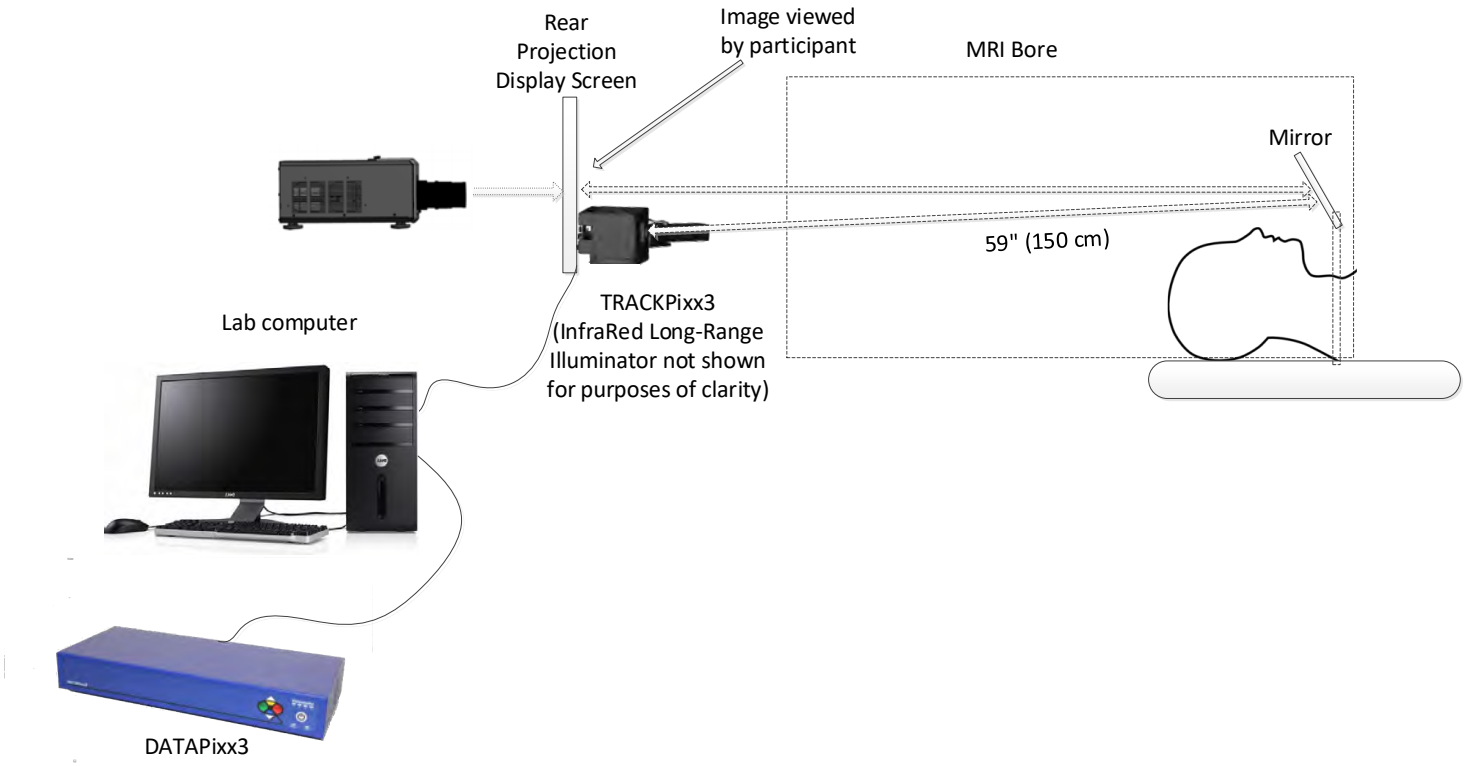


Figure 32 – Typical TRACKPixx3 MRI Camera Placement (screen mount example shown)

This high-level figure only shows general connectivity setup and arrangement. For complete cabling and hardware installation instructions for your TRACKPixx3 MRI system, please refer to the **TRACKPixx3 /MRI Installation Guide** supplied with your product.

TRACKPixx /mini

The TRACKPixx /mini is a versatile eye/gaze-tracking solution. It does not require a dedicated PC to process eye images and generate gaze information; all image processing is performed within the TRACKPixx /mini hardware. The system comes standard with a USB I/O hub and a RESPONSEPixx 5-Button box to record participant responses to various experimental stimuli.

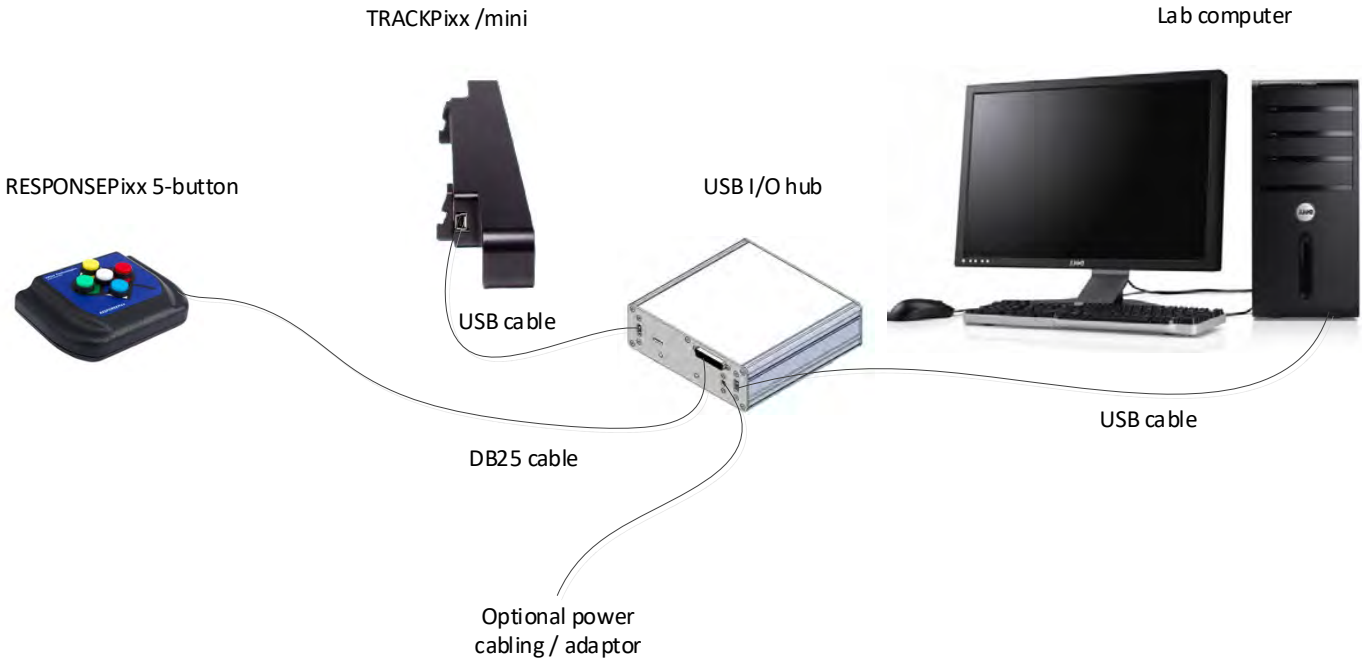


Figure 33 – TRACKPixx /mini System connections using RESPONSEPixx 5-button box

This high-level figure only shows general connectivity setup and arrangement. For complete cabling and hardware installation instructions for your TRACKPixx /mini system, please refer to the **TRACKPixx /mini Installation Guide** supplied with your product.

Getting Started – Eye-Tracking Experiments

This chapter gives you the necessary information on how to rapidly get up and running with your VPixx eye-tracking product(s) such as the TRACKPixx3, TRACKPixx3 /MRI and TRACKPixx /mini.

The chapter is divided according to your specific product and research needs.

Before setting up your first eye-tracking session, CAREFULLY READ the safety information contained in the WARNINGS AND SAFETY INFORMATION section AND the information contained in the Installation Guide(s) relating to your product(s):

- TRACKPixx3_InstallationGuide.pdf
- TRACKPixx_mini_InstallationGuide.pdf
- TRACKPixx3_MRI_InstallationGuide.pdf
- DATAPixx3_InstallationGuide.pdf
- InfraRed_Short_Range_UserManual.pdf
- InfraRed_Long_Range_UserManual.pdf

Transporting & Storing your Equipment

Following initial equipment unpacking and setup, your TRACKPixx system should be moved only when necessary, such as when your experiment requires a different equipment setup. When not in use, cover the system with a soft protective sheet in order to protect the equipment from dust accumulation.

When transporting your system from one location to another, NEVER pick up the TRACKPixx3/ illuminator/arm support by the camera or illuminator. If they must be carried together without being disassembled and packaged, pick up the assembly by placing one hand on the arm support's vertical support bar and the other hand firmly holding it by the base.

TRACKPixx Setup – General Notes

This section contains general notes and hints to help you with your TRACKPixx setup, no matter what the exact system you have (TRACKPixx3 Short-Range, Long-Range, TRACKPixx3 /MRI or TRACKPixx /mini). These notes apply to all TRACKPixx systems.

Participant comfort and safety considerations

It is of crucial importance for all eye-tracking setups that the participant be comfortable at all times. In addition to the safety issues discussed in the introduction chapter of this document, the following issues should be considered when positioning the participant and configuring the equipment for your experiment.

- Constant communication between the researcher and participant must be possible so that the participant may relay any discomfort he may be experiencing at any stage of the experiment. This is especially important for MRI setups.
- The participant should be positioned in such a way as to minimize the physical or emotional discomfort he may experience during the experiment.
- Mascara should be removed from the participant.
- The participant should be able to view the entire display screen with as little strain or eye fatigue as possible.
- The experiment should be completed in a timely manner.
- **FOR MRI SYSTEMS**, follow all standard participant preparation procedures in regards to MRI rooms (removing jewelry, metal objects, mascara, etc.) and replace the participant's glasses (if any) with specially-designed MRI glasses. Soft contact lenses are allowed, but rigid contact lenses should be replaced with MRI glasses.

Lighting considerations

Better tracking results are obtained when the participant's pupils are at their most dilated. It is therefore suggested that you turn off ambient lighting.

Since they contain infrared light, it is of crucial importance to prevent halogen light and exterior sunlight from reaching the participant's face and eyes.

System first power up

After your TRACKPixx3 system is properly setup and the DATAPixx3 is connected to your facility's power outlet, its front panel power button will be RED. This signifies that the system is powered but in an OFF/Standby state. Pressing the DATAPixx3's power button will activate the system and the power button will turn GREEN. The STATUS and FAULT LEDs should be unlit. If the FAULT LED is RED, it implies an internal fault that should be investigated.

Once the system is activated, one or more of the device's VideoBahn LEDs will be GREEN indicating that it is connected by fiber optic cable to an appropriate device.

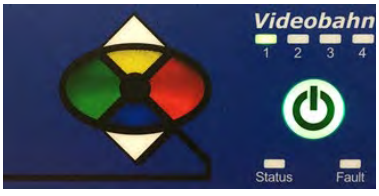


Figure 34 – DATAPixx*3* front panel power button and LEDs

The TRACKPixx systems require 15 minutes of warm-up time after power up before use. This warm-up period will ensure the best image and tracking quality.

Configuring your TRACKPixx using PyPixx

PyPixx is a Python application allowing a user to configure any VPixx Technologies device. It can also provide information on the devices, perform calibrations, calibration tests, check for and perform firmware updates and run demos. PyPixx operates on Windows, Mac and Linux operating systems.

This section briefly presents the PyPixx operating environment. It introduces the environment's menus, tabs, toolbars and status bars. Figure 35 shows PyPixx's main window when the application is first opened (for purposes of this example, we connected two VPixx devices: one TRACKPixx3 and one DATAPixx3).

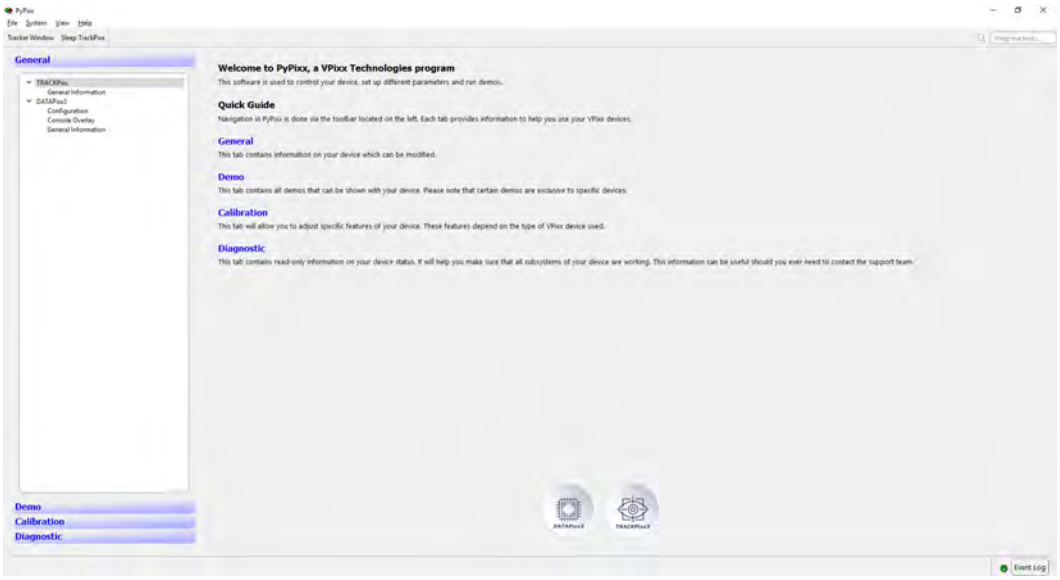


Figure 35 – PyPixx main window (TRACKPixx3 and DATAPixx3 devices detected)

The main window's left portion is divided into four tabs: *General, Demo, Calibration* and *Diagnostic*. Each tab displays the VPixx devices currently connected to the computer and which are powered ON.

- The *General* tab contains VPixx device information and allows you to configure certain device features.
- The *Demo* tab contains demos relevant to your device(s). Please note that certain demos are exclusive to specific devices.
- The *Calibration* tab allows you to recalibrate analog sub-systems in your device(s). These features depend on the type of VPixx device(s) used.
- The *Diagnostic* tab contains read-only information pertaining to your device status. It allows you to verify that all of your device subsystems function properly. This information can be useful if you contact the support team.

Device Toolbar

The device toolbar, located in the top portion of the main window, offers shortcuts to key device features. It can hold different feature categories, such as hardware and/or software features. Some devices have their own toolbar with useful shortcuts. It is possible for you to hide or show some or all of these shortcuts or the device toolbar entirely.

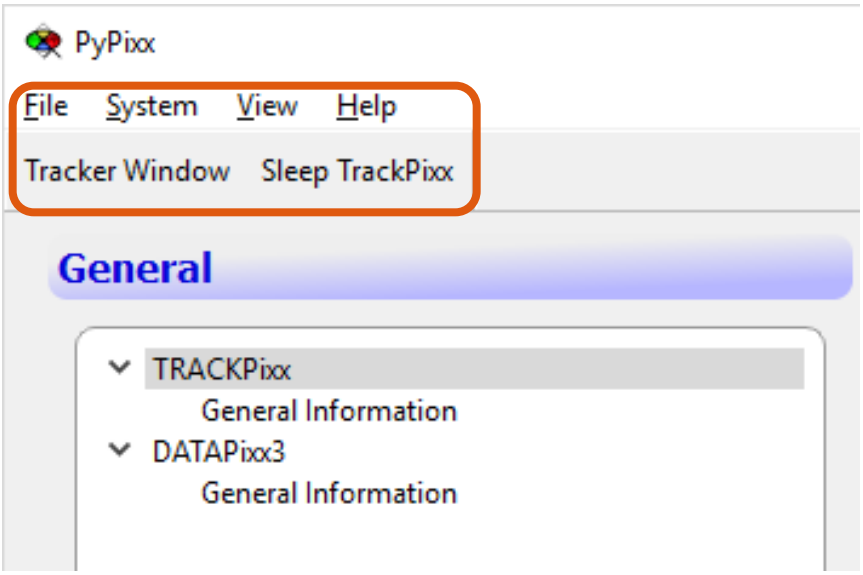


Figure 36 – Device Toolbar

Central window

The central window provides information specific to the active device and the actions that can be performed on it. The figures below show PyPixx's Central window when two devices are detected.

Figure 37 – Central Window (two devices detected)

Note that depending on which device you are using and which tab you clicked on, the central window will display a different message. Most of the information displayed is to quickly let you know where you are and what you can do.

Status bar

The status bar at the bottom of the main window provides additional information to the user as to the functioning of PyPixx and the devices currently detected by the application. The status bar will also warn the user when the device is busy processing instructions.

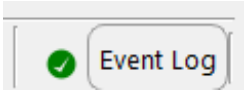


Figure 38 – PyPixx status bar

Should an error occur on a VPixx device or in PyPixx, information related to this error may be obtained by clicking on the *Event Log* button.

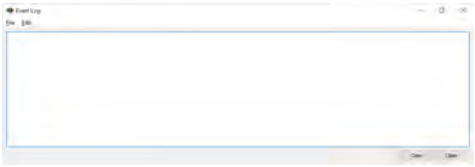


Figure 39 – Event Log pop-up window

General Information Widget

The General Information window includes general information such as firmware and serial number.

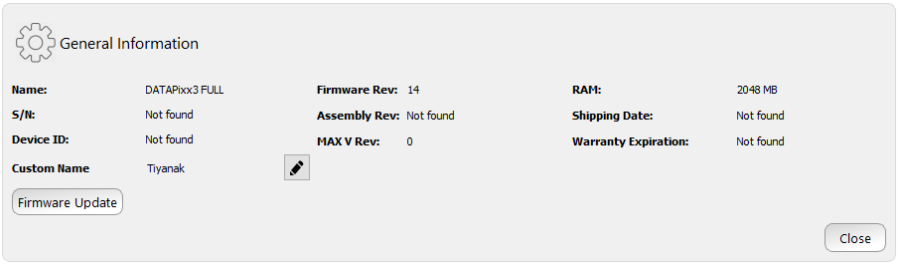


Figure 40 - The new compact GUI for the General Information window

This UI simply lists information on the device and allows it to be renamed. Simply click the pencil icon to edit the name of the device. This is saved directly on the device and it is kept when the device is restarted. To remove the name of the device, simply leave the box empty.

Figure 41 - Custom name feature

Console Overlay Option Widget for DATAPixx3 and TRACKPixx3

This feature is for a setup using a DATAPixx3 and a TRACKPixx3 only and is only possible on the LCD monitor connected in display port output 2 of the DATAPixx3.

The Console Overlay Options widget can be accessed from the general toolbox under DATAPixx3. It can be used to preview and set the overlay settings for the tracker window and the stimuli.

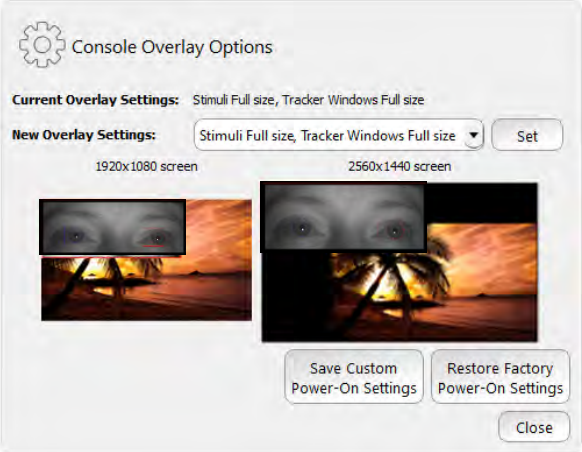


Figure 42 - Console Overlays widget

Once you have selected your setting, it is possible to save it to the device such that it is the default option when the device is powered on. This can be done with the buttons located at the bottom of the window.

Tracker Demo

PyPixx contains a TRACKPixx demo that may be used to quickly get up and running using your TRACKPixx device.

You may access the demo by following this procedure:

1. From the PyPixx main window, click on the *Demo* tab. This will display the PyPixx demo tab main window.

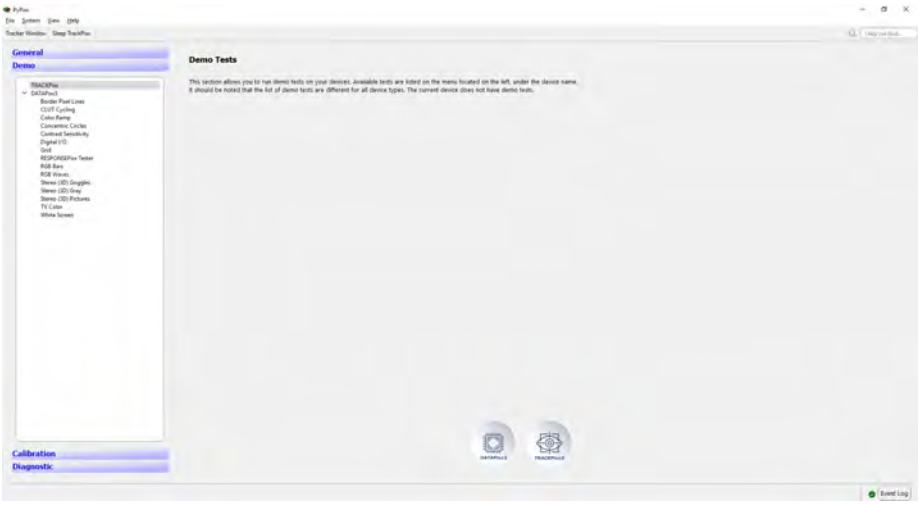


Figure 43 – Tracker demo window (connected TRACKPixx3 and DATAPixx3 devices)

2. In the left device selection menu, click on your *TRACKPixx* device. This will instantly display the *Tracker General Settings* demo window specific to your TRACKPixx system.

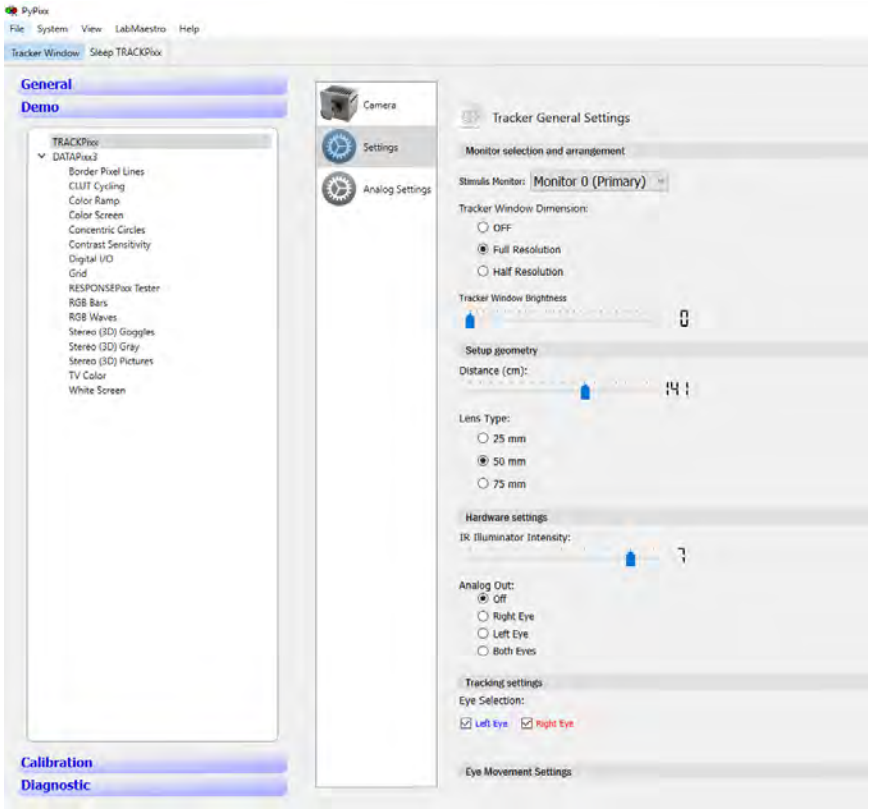


Figure 44 – *Tracker General Settings* window (top portion not showing Eye Movement Settings)

The **Stimuli Monitor** checkbox allows you to select the screen where the calibration will be run. Take note that it is also possible to change the display screen using the left and right arrows.

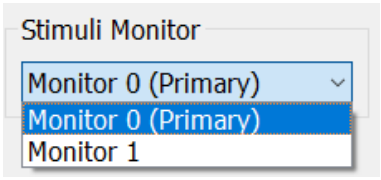


Figure 45 – Stimuli Monitor checkbox

Checking either the *FULL Resolution* or *HALF Resolution* boxes of the ***Tracker Window*** settings displays the camera image on the console monitor. Adjust the brightness using the *Tracker Window Brightness* scroll bar from a value of 0 to 255.

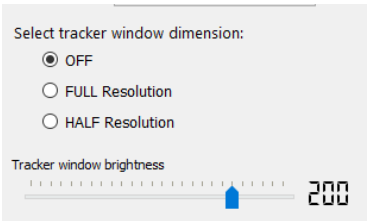


Figure 46 – Tracker Window controls

The **distance (cm)** represents the distance between the Tracker camera lens and the participant's eyes.

The **Lens type** checkboxes allow you to select the lens type currently installed on your camera.

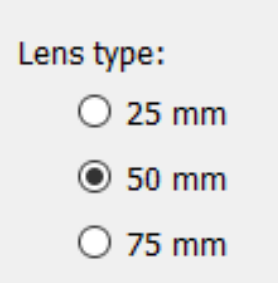


Figure 47 – Lens Type checkboxes

When it comes to Lens Type, select the lens that is presently installed on the tracker. Typically, MRI/MEG setups should use a 75 mm lens.

The TRACKPixx /mini system does not require lens selection. The **Lens Type** choices will be greyed out.

The **IR Illuminator Intensity** scroll bar allows you to vary the intensity of the LEDs of the InfraRed Illuminator to optimize image brightness for your TRACKPixx3 system (not applicable to TRACKPixx /mini systems). Notice that as you move the scroll bar, the IR Illuminator intensity displayed on a scale of 0-8 to the right of the scroll bar changes.

Figure 48 – InfraRed Illuminator Intensity scroll bar

Although it is safe to use the maximum intensity (8), it may not always be required. If a participant with dark-colored eyes is being tracked correctly, for example, reducing illumination can improve signal-to-noise ratio. The TRACKPixx calculates pupil and corneal reflex thresholds automatically.

IR Illuminator LED intensity controls are not available for TRACKPixx /mini devices.

The **Analog Out** checkboxes allow you to select which **screen eye** (or **BOTH EYES** or no eye by checking the **OFF** box) information drives the output voltage (409.6 pixels/Volt) corresponding to the on-screen position values (X and Y coordinates) for the selected eye. The maximum voltage value (both horizontally and vertically) is +/-10 V. The 0 V value

corresponds in all cases to pixel 0 (where 0,0 is the center of the screen) and the maximum screen size the Analog Out can output to is 8000 x 8000 pixels.

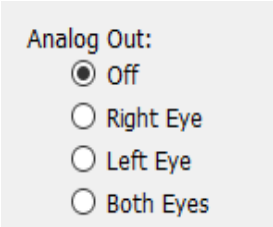


Figure 49 – Analog Out checkboxes

Note that the left or right screen eye refer to the left or right eye SHOWN ON THE CONSOLE OVERLAY. This may or may not correspond to the participant's left or right eye depending on your setup. A setup involving a mirror (as for MRI setups, for example) means that the screen eye corresponds to the participant's corresponding eye (for example, left screen eye corresponds to participant's left eye). A setup without a mirror (tabletop or MEG applications, for example) inverts this logic (left screen eye would correspond to participant's right eye).

If you only select one eye, you will get calibrated X and Y positions and Pupil Diameter on DAC0, 1 and 2, respectively (12.8 pixels/Volt).
DAC 0, 1 and 2 are channels of the digital to analog converter, which generates the analog eye position signal. We have a maximum of four analog output channels on DB 25 pins 11, 12, 24 and 25 (refer to Figure 50 below).
These channels can be configured according to the researcher's needs using one of three 3 modes:

- Left Eye X, Y and Pupil Diameter on DAC 0, 1, 2
- Right Eye X, Y and Pupil Diameter on DAC 0, 1, 2
- Left Eye X, Y and right Eye X, Y on DAC 0, 1, 2, 3

The Pupil diameter is in pixels (camera space pixels) and the values of pixel size 0 to 100 are mapped from 0 V to 10 V with the above-stated ratio of pixels/Volt (12.8 pixels/Volt).

If both eyes are selected for the analog output, the data format mode is Left X, Left Y, Right X, Right Y (on DAC0, 1, 2, 3).

Note that if the tracker is not yet calibrated, no output will be generated and if the participant blinks or the tracking is otherwise lost, the value will be set to NaN (not a number).

5	ADC8	18	ADC9
6	ADC10	19	ADC11
7	ADC12	20	ADC13
8	ADC14	21	ADC15
9	REF0	22	REF1
10	GND	23	+5 VDC **
11	DAC0	24	DAC1
12	DAC2	25	DAC3
13	GND	Shield *	

Connector type: D-SUB, 25 pins
Gender: Female

Figure 50 – Connector pin information

For MOST calibrations, BOTH of the **Tracking Settings** checkboxes must be checked. This signifies that both eyes will be considered during the next calibration procedure. **Deselecting** one or both of the eyes in these checkboxes means the calibration will NOT take the unselected eye(s) into consideration when determining the validity of a calibration measurement.

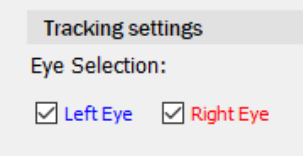


Figure 51 – Eye Selection checkbox

At the very bottom of the *Settings* window, the *Eye Movement Settings* can be set.

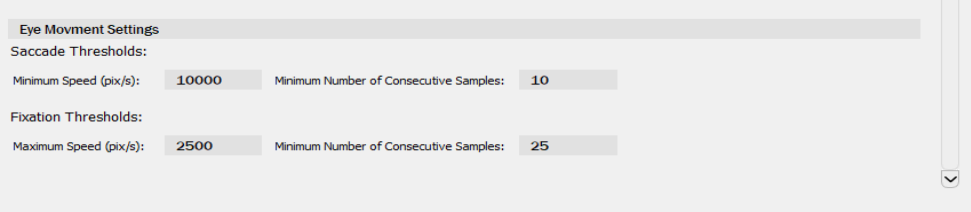


Figure 52 – TRACKPixx Demo *Settings* window (bottom portion)

The currently used thresholds are as follows:

1. For a sample to be flagged as a **saccade**, the velocity of the eye movement must be over 10 000 pixels per second for 10 consecutive samples.
2. For a sample to be flagged as a **fixation**, the velocity of the eye movement must be under 2500 pixels per second for 25 consecutive samples.

Click on the *Camera* icon to access the TRACKPixx demo *Camera* window.

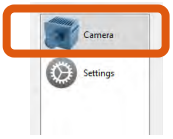


Figure 53 –Camera icon

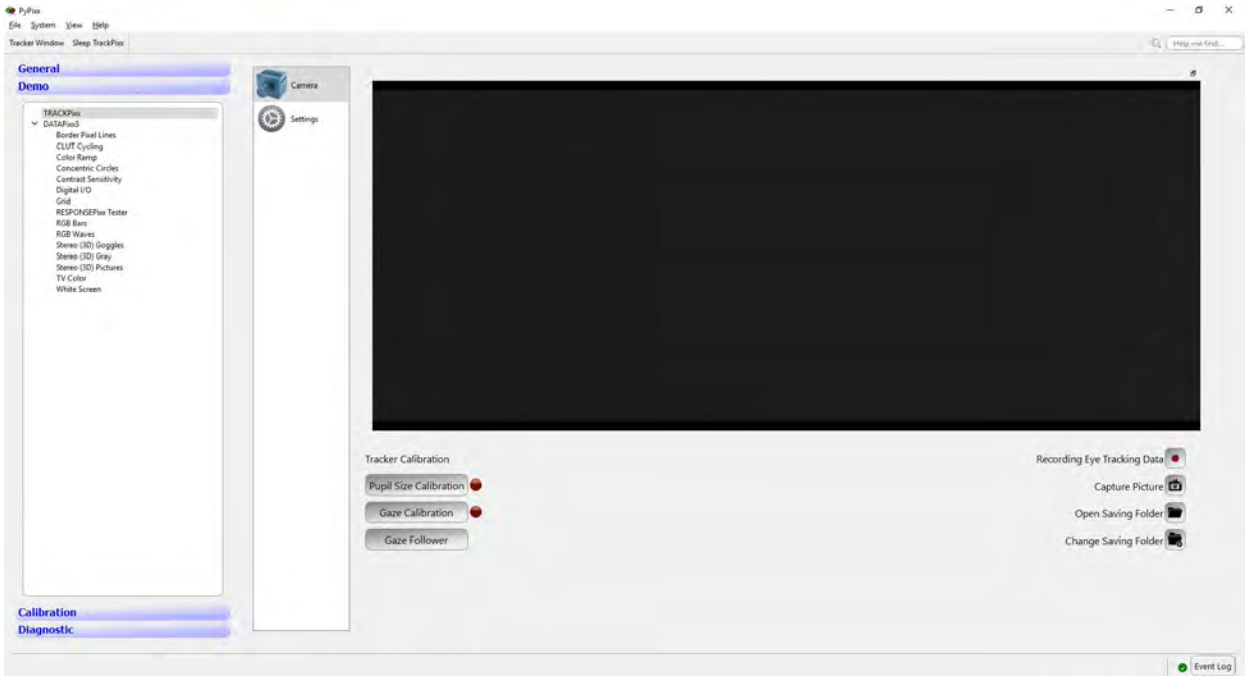
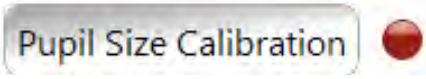
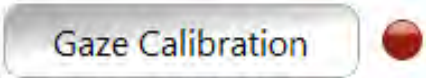
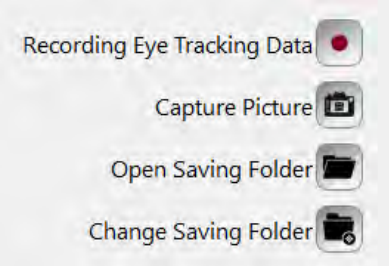


Figure 54 –TRACKPixx Demo Camera window

The following table describes the buttons of the TRACKPixx demo Camera window allowing you to control your image capture session.

Table 3 – Tracker Control Buttons

Button	Description
	Allows you to start the Pupil Size Calibration procedure. This will offer the experiment better accuracy. The red dot will turn green once a calibration has been completed.
	Allows you to start the Gaze Calibration procedure prior to running your experiment. Clicking on the Save Calibration button will erase any previous calibration results. Once a calibration is completed, it can be used by any software. For example, a calibration done in MATLAB can be used in Python. The red dot will turn green once a calibration has been completed.
	Allows you to verify that the previous calibration session was successfully completed by having the participant confirm that the blue and red gaze indicators appear on screen where they are looking. The Gaze follower automatically starts after a calibration. This button allows you to run the Gaze follower at any given time. **NOTE:** if no calibration is currently loaded in the system, the Gaze Follower will not start.

Button	Description
	The **Recording Eye Tracking Data** button allows you to record eye position data. When a recording is active, a blinking  icon will appear to the right of this button and the button itself will change from a red circle to a black square. Pressing this red square while a recording is active will stop the recording. The recording is saved in a .csv file in the "Saving Folder" location, with the format: *TPx_year-month-day_hour-minute.csv* (where date/time is the time at which the recording was started) The **Capture Picture** button allows you to take an instantaneous ''snapshot image'' of the image currently captured by the camera (see below for additional information). Clicking on this icon will display, for a few seconds, the name of the file created.

*For the TRACKPixx /mini system, the **Tracker Window** checkboxes are greyed out.*

- The **Capture Picture** will save a PNG file. On a Windows system, the default path for this file is:

  C:\Users\USERNAME\Documents\VPixx\pictures

  On Linux, the default path is:

  /tmp/VPixx/pictures
- The **Recording Eye Tracking Data** button will save a CSV file. Data will be saved into this file every 15 seconds in the case of the TRACKPixx3 products and every 16 milliseconds in the case of the TRACKPixx /mini. On a Windows system, the save path can be chosen by clicking on the *Change Saving Folder* (the default path is: **C:\Users\USERNAME\Documents\VPixx**)
  On Linux, the default path is:
  /tmp/VPixx/data
  Inside the CSV file, the following information can be found: *Timetag, Left Eye X, Left Eye Y, Left Pupil Diameter, Right Eye X, Right Eye Y, Right Pupil Diameter, Digital Input Values (24 bits), Left Blink Detection, Right Blink Detection, Digital Output Values (24 bits), Left Eye Fixation Flag, Right Eye Fixation Flag, Left Eye Saccade Flag, Right Eye Saccade Flag, Message code, Left Eye Raw X, Left Eye Raw Y, Right Eye Raw X, Right Eye Raw Y.*

Here is a detailed explanation of each field:

Name of Field	Description
Timetag	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.
Eye X, Y	Screen coordinate, in pixels, corresponding to the calibrated position of the eye. Uses a Cartesian system where (0,0) corresponds to the center of the display.
Eye Blink Detection	0 if the eye is open, 1 if it is closed
Eye Pupil Diameter	The size of the pupil's major axis in pixels
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.
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.
Eye Fixation Flag	Default 0, changes to 1 if the conditions for an 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.
Eye Saccade Flag	Default 0, changes to 1 if the conditions for an 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.
Message Code	Placeholder column. Functionality will be implemented in a future release.
Eye Raw X, Y	Value, in pixels, of the vector between the left eye pupil center point and the left eye corneal reflection point.

Performing a Calibration using the PyPixx Tracker Demo

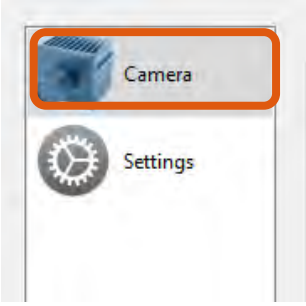
If you choose to run a calibration using the Tracker Demo in PyPixx, follow this procedure.

A calibration is only valid for ONE participant. Changing participants will therefore invalidate the current calibration and require a new calibration step.

ONLY ONE CALIBRATION can be loaded into the tracker at a given time. The system will use the latest calibration data only.

A change in lighting conditions, participant head placement or changing any other experimental condition may require a new calibration.

1. Sit the participant comfortably and adjust the chinrest properly. If in an MRI setting, ensure that the shadows caused by the head coil are minimized.
2. Adjust the distance between the user and the camera according to the information found in the *Installation Guide* specific to your TRACKPixx product.
3. Display the TRACKPixx overlay by checking, in the TRACKPixx *Settings* demo window, either the **Half Resolution** or **Full Resolution** boxes of the **Tracker Window** checkbox.
4. Adjust IR Illuminator Intensity to the maximum or near maximum (7 or 8).
5. Select **Lens Type**.
6. Click on the *Camera* icon.



7. **Pupil Size Calibration** button. This will display the pupil calibration screen prompting you to follow the on-screen directives.

8. Click on the **Gaze Calibration** button. This will display the Calibration screen and follow the on-screen directives.

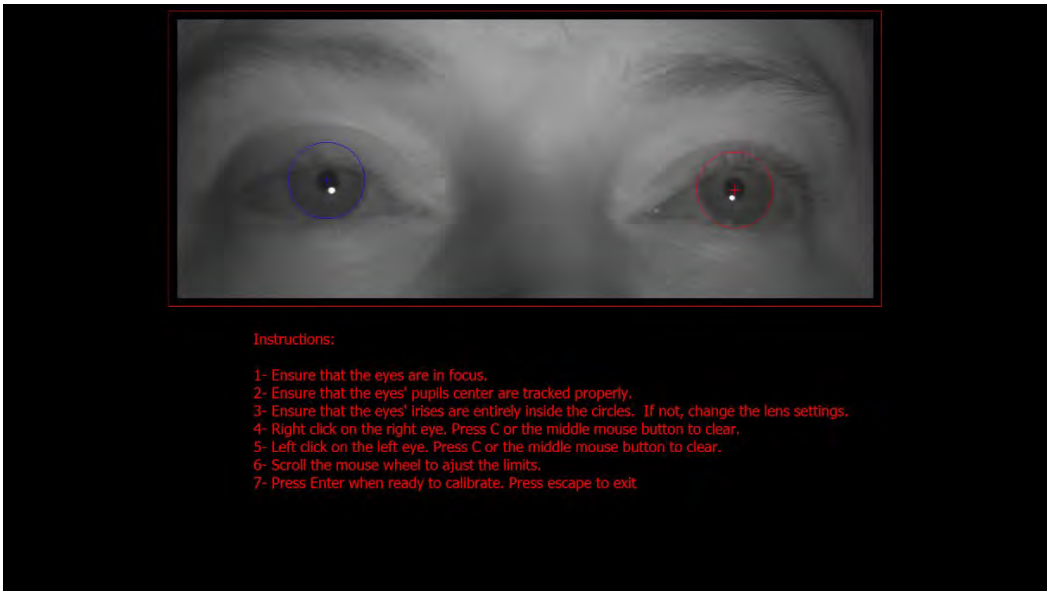


Figure 55 – Tabletop Calibration

9. Reposition the TRACKPixx camera (and InfraRed Illuminator for a TRACKPixx3 system) position to aim directly at the participant's eyes. For TRACKPixx3 systems, adjust the camera focus using the focus ring. The focus should be adjusted such that the corneal reflection is as sharp and as circular as possible.
10. **Setting Search Limits: points 4 and 5, which are optional**, allow you to define the search limits for each eye. This means that the system will not consider pupil positions occurring beyond these search limits.

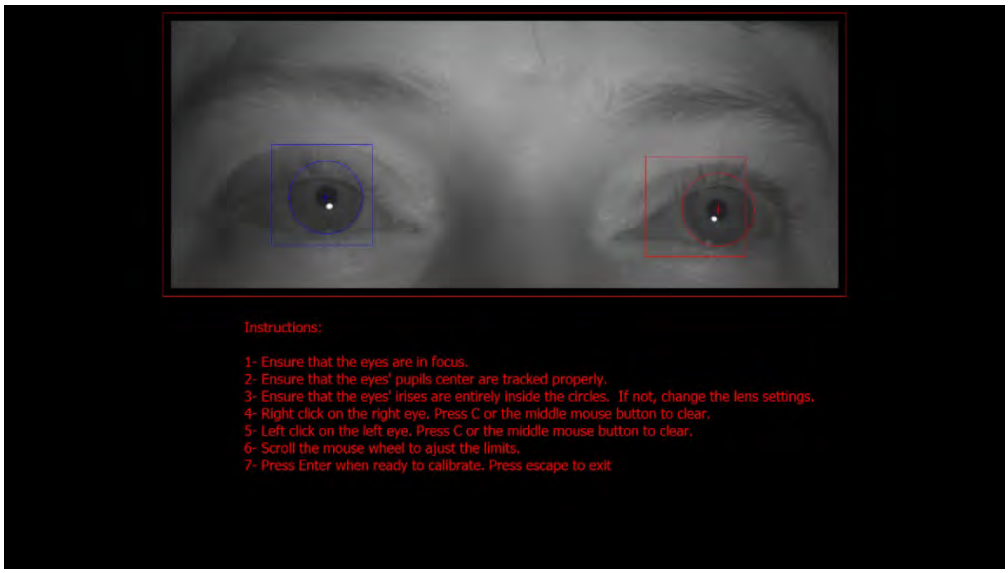


Figure 56 – Setting Search Limits

11. **Starting the calibration:** at point 7, you can press **Enter** to start the calibration. A succession of 13 blinking red/white targets will be shown one after the other on a black background. These are used to create the mapping between eye position/gaze position on the screen. Wait until the 13 targets have appeared on screen.
12. After the 13 targets have appeared on screen, the **Gaze follower** screen will appear which allows you to validate the calibration. It also verifies that the participant's gaze (represented by **two circles of different color (blue and red)**, one for each eye) corresponds either to the position of the mouse pointer, or to specific squares displayed on screen.

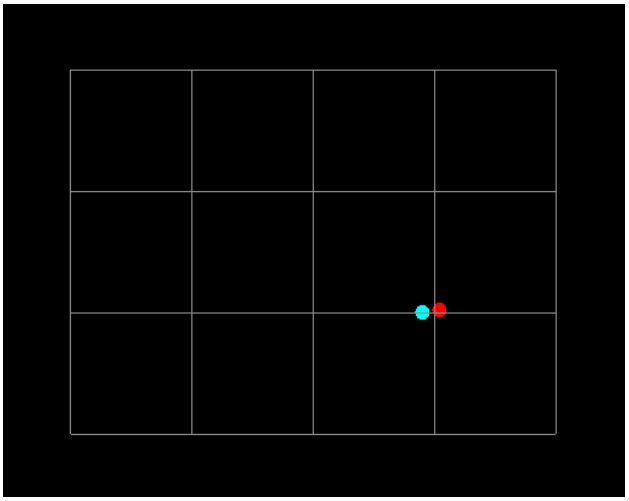


Figure 57 – Gaze Follower screen

The blue circle represents the LEFT SCREEN EYE, while the red circle represents the RIGHT SCREEN EYE.

As previously mentionned, note that the left or right screen eye may or may not correspond to the participant's left or right eye depending on your setup. A setup involving a mirror means that the screen eye corresponds to the participant's corresponding eye (for example, left screen eye corresponds to participant's left eye).

A setup without a mirror inverses this logic (left screen eye would correspond to participant's right eye).

Also, a change in lighting conditions, participant head placement or changing any other experimental condition may require a new calibration step.

13. Once the Gaze Follower validation is successfully completed, the calibration is considered complete and the experiment can begin.

Calibration tab

The Calibration tab contains the analog subsystem calibration options available for your currently connected VPixx products. In the case of the TRACKPixx, no calibration procedures are available; the calibration necessary to run an eye-tracking experiment is done through the *Demo* tab.

Diagnostic tab

The Diagnostic tab allows you to verify that all of your currently connected VPixx products are functioning correctly. If they are not, the information gathered here will be useful when you contact the VPixx support team.

In the case of the DATAPixx3, the available diagnostic tools are explained in the figure below.

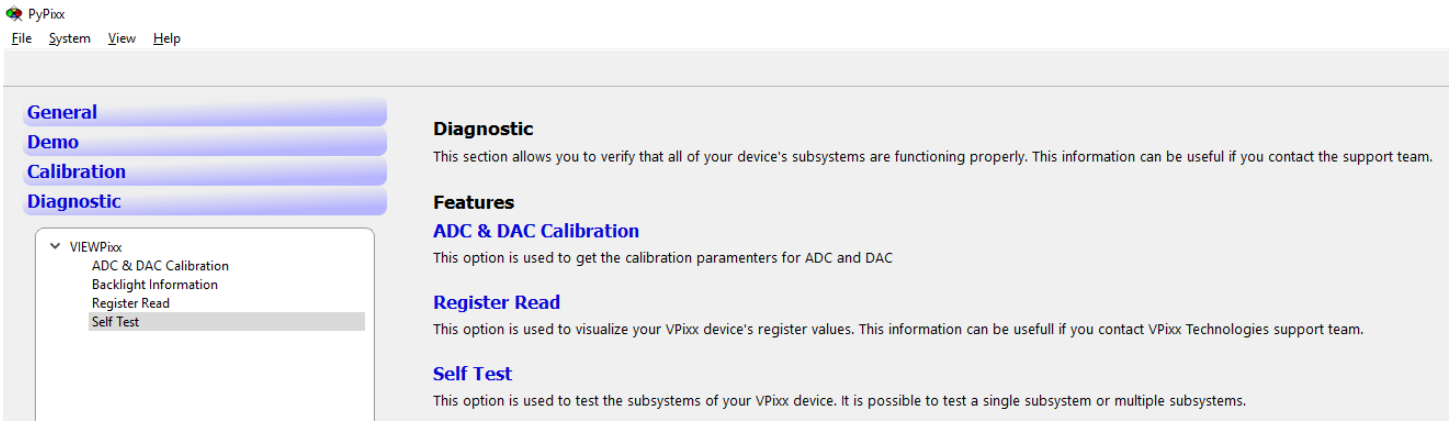


Figure 58 – Diagnostic tab

The *ADC & DAC Calibration* and *Register Read* tests allow you to update the specific diagnostic information and send it to the VPixx support team if required.

Clicking on *Self Test* allows you to select exactly which self-diagnostic test(s) you wish to run for your DATAPixx3. If you wish to run a complete test, activate all the checkboxes and click on the **Run** button. The green progress bar from the Selftest window displays the test's progress and when completed, the window will display the results as shown below. These may be relayed to the VPixx support team if requested.

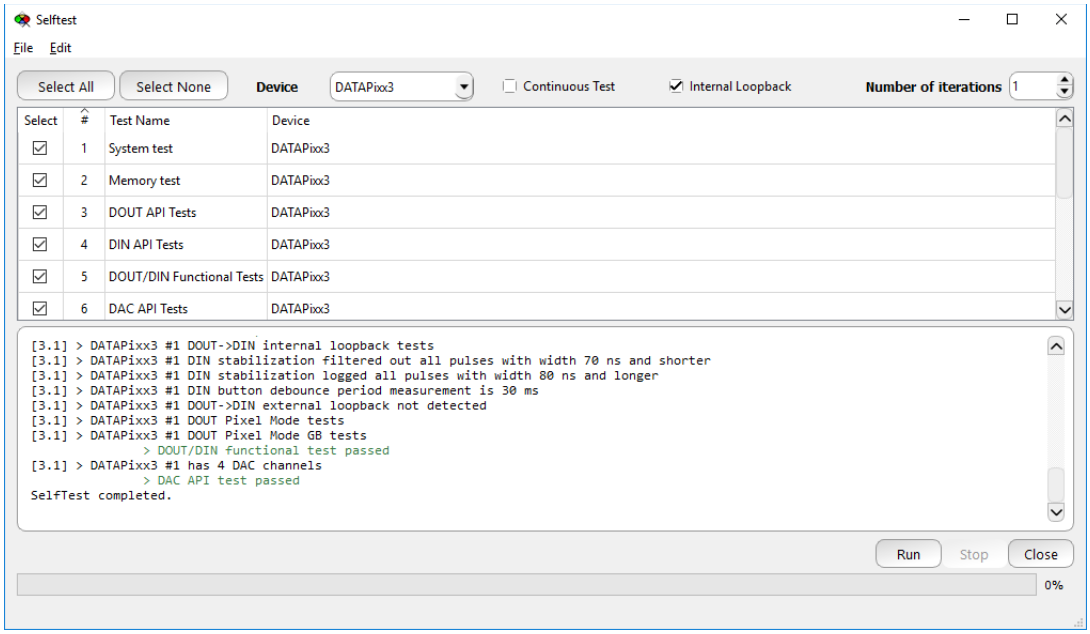


Figure 59 – DATAPixx3 Self Test

Similarly, for the TRACKPixx, the self test can be run in a similar fashion. Simply select the desired test(s) and click the **Run** button.

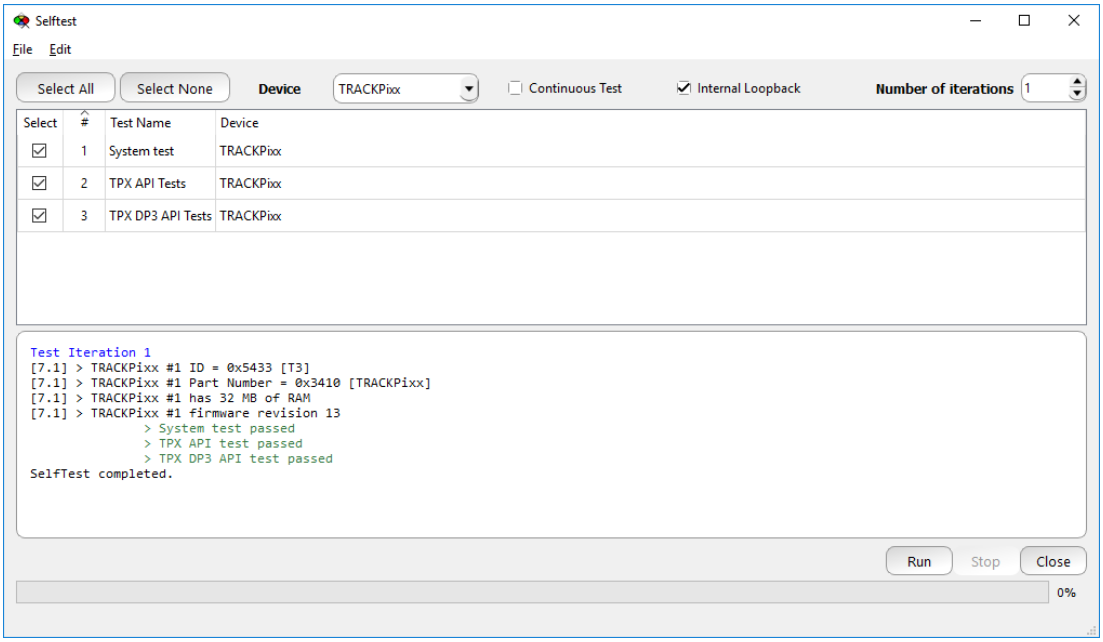


Figure 60 – TRACKPixx Self Test

Using Python to setup your TRACKPixx system

Use Python to setup your VPixx equipment in expectation of a future eye-tracking experiment. To do so, follow this procedure:

1. Verify that Python 3.7 is installed on the computer.

Most of the pypixxlib development has been done on Python 3.7.

2. Install the VPixx Python package using ***pip***. If you do not have ***pip*** installed, please follow these instructions:
    a. Download *get-pip.py*
    b. Run it through the command line: Python get-pip.py
    c. To enable the use of pip from the command line, ensure the Scripts subdirectory of your Python installation is available on the system variable PATH. This is not done automatically. Alternatively, navigate to the Python27/Scripts folder. Once there, simply install our package from the command line:

    pip install -U PATH_TO_FILE/pypixxlib.tar.gz

3. You are now ready to use Python and combine it with VPixx Technologies high performance systems.

In the pypixxlib package, ***Tracker.py*** is a Python module containing classes to manage the TRACKPixx. To be able to interact with the TRACKPixx, you need to import this module and use the *TRACKPixx3* class (if you are using a TRACKPixx3 product) or the *TRACKPixxMini* class if you are using a TRACKPixx /mini.

Once this class has been built, you can access different methods to setup your TRACKPixx3 properly. To access the documentation for each method, inspect the doc string of the tracker base class which is also included in the same module.

1. Import the Tracker module and build a TrackPixx3 object.
2. Call 'Open'. This is the first method to call whenever you wish to setup any VPixx device. This method awakens the TRACKPixx3 and powers up the LEDs of the infrared illuminator.
3. Call 'SetLedIntensity'. This method activates the infrared illuminator LEDs to an intensity equal to the *intensity* value you typed in. Allowed *intensity* values are 1-8. We recommend using intensity values higher than 6, but this may vary according to your specific testing environment and the TRACKPixx3 camera aperture.
4. Call 'SetIrisSize'. This method allows you to set the lens type to correctly track the participant's eye movements and estimate iris size. **THE IRIS SIZE VALUE IS OF CRITICAL IMPORTANCE!** Setting the wrong iris size will yield incorrect tracking data. Depending on the lens used in your TRACKPixx3 (either 25, 50 or 75 mm), you will need to set your iris size to a value determined in the next table.

Table 4 – Iris Size Values (2)

TRACKPixx3 lens / distance to participant	Iris size value (pixels)
25 mm / 60 cm	70
50 mm / 60 cm	140
75 mm / 150 cm	100

5. Call 'ShowOverlay'. This method displays the eye video images in the corner of the console monitor connected to the OUT2 Display Port output.
6. Call 'GetImagePtr'. This method returns a pointer to the camera image.
7. For other calls, you can also use the low level API and access the available API documentation on the VPixx website.

Performing a calibration in Python

Following the preliminary setup procedure described above, you can also use Python to calibrate your VPixx equipment. To do so, follow this procedure:

Calibrating your TRACKPixx3 system

Two methods exist to calibrate your TRACKPixx3 system using Python.

1. Use PyPixx software's integrated calibration tool to calibrate the TRACKPixx3 and save your calibration. You can then load the calibration using Python's 'LoadCalibration' method.
2. You can program the calibration directly in Python by using the appropriate method.
    a. Import the Tracker module and build a TrackPixx3 object.
    b. Refer to the section for information on how to setup the tracker.
    c. Define the 13 calibration points to be used. We recommend using points that are approximately equidistant from one another. The calibration points need to cover the region of interest used with the gaze follower for the experiment. The following figure shows a pattern that represents an example of a proper calibration point pattern.

Figure 61 – Calibration point pattern example

d. Display the first point on screen.
e. Wait a moment to allow the user to locate and fixate on the calibration point.
f. Call 'GetEyePositionDuringCalib' from the TRACKPixx3 object.
g. Repeat steps d-e-f for all 12 remaining points.
h. At the end of the process call the finishCalibration method to compute the calibration coefficients.
i. If the method returns a *True* value, it signifies that the calibration is a success.

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.

Calibrating your TRACKPixx /mini system

Two methods exist to calibrate your TRACKPixx /mini system using Python.

1. You can use PyPixx software's integrated calibration tool. The **remote mode** allows the participant to move his head within a specific range.
2. You can program the remote calibration directly in Python using the appropriate method.
    a. Import the Tracker module and build a TRACKPixxMini objet.
    b. Call 'initializeCalibration' from the TRACKPixxMini object to Initialize and get the calibration points.
    c. Display the first point on screen.
    d. Wait a moment to allow the user to locate and fixate on the calibration point.
    e. Call 'calibrateTarget' from the TRACKPixxMini object.
    f. Repeat steps d-f for all remaining points.
    g. At the end of the process call the 'finalizeCalibration' method to compute the calibration coefficients.
    h. If the method returns a *True* value, the calibration was a success.

Performing an eye-tracking experiment in Python

Following the proper setup and calibration of your system (see above), or, optionally, by loading a previously-saved calibration, you can now use Python to run and manage your eye-tracking experiment.

The following table lists the Python methods that may be required to run your experiment. These methods are included in the Tracker class and installed with pypixxlib.

Table 5 – Python methods

Method	Arguments [default]	Returns	Description
TRACKPixx /mini (remote calibration)			
calibrateBuiltInTarget	• *targetIndex (int): Index of the target to calibrate.*	None	Initializes the built-in calibration.
finalizeBuiltInCalibration	None	None	Calibrates a target with a built-in calibration.
initializeBuiltInCalibration	None	None	Initializes the built-in calibration process. The built-in calibration is also named the remote calibration. The function returns the target's positions to be calibrated.
TRACKPixx3			
clearDeviceCalibration	None	None	Clears the calibration from the tracker. The device is thus no longer calibrated.
Close	None	None	Closes a TRACKPixx device.
convertCoordSysToCartesian	• *inList (List): List of coordinates to convert.*	List: list of converted coordinates.	Converts custom coordinates to system coordinates.
convertCoordSysToCustom	• *inList (List): List of*	List: list of converted coordinates.	Converts system coordinates to custom coordinates.

Method	Arguments [default]	Returns	Description
	coordinates to convert.		
finishCalibration	None	None	Finishes the calibration on the device. This method is to be called when you are done displaying stimuli for a calibration. It will calculate the calibration parameters on the device and set the state to calibrated (which can be verified through *isDeviceCalibrated*
getEyePosition	None	List: list of gaze positions, such that first elements are screen_x_left_eye, screen_y_left_eye, screen_x_right_eye, screen_y_right_eye	Returns the current calibrated gaze position.
getEyePositionDuringCalib	• *x_screen (float): x-coordinate of the stimulus* • *y_screen (float): y-coordinate of the stimulus*	None	Tells the camera to immediately get eye position. This function should be called during calibration when there is a calibration fixation point displayed on the stimulus display.

Method	Arguments [default]	Returns	Description
getImagePtr	None	Integer pointing to image data.	Returns a pointer to the image data stored in RAM.
getLEDintensity	None	Integer of the intensity value(between 1 and 8)	Gets the LED intensity of the infrared LED connected to the camera tracker.
getTime	None	Float: time in seconds.	Gets the device time from DATAPixx since power up.
hideOverlay	None	None	Deactivates the DATAPixx3 Overlay to display the eye image and eye cursors on the console monitor.
isDeviceCalibrated	None	True when tracker has been successfully calibrated, false otherwise.	Returns the calibrated state of the tracker.
loadCalibration	None	None	Loads the saved calibration to the tracker.
Open	None	None	Opens a TRACKPixx device.
saveBufferedData	None	None	Saves data from last_read_address up to TPxGetReadAddr() into today's default file.
setIrisSize	• *lens (int): Diameter of the len in millimeter (25, 50,75)*	None	Sets the iris size depending on lens type and distance.

Method	Arguments [default]	Returns	Description
	• dist (int): Distance between the camera and the observer.		
setLEDintensity	• *led_intensity (int)*		Sets the LED intensity of the infrared LED connected to the camera tracker.
setUpDataRecording	None	None	Sets up the base address and buffer size for schedule recording of the TRACKPixx data.
showOverlay	None	None	Activates the DATAPixx3 Overlay to display the eye image and eye cursors on the console monitor.

Using MATLAB to setup your TRACKPixx3 system

Use MATLAB to setup your VPixx equipment in expectation of a future eye-tracking experiment. To do so, follow this procedure:

1. Call Datapixx ('Open'). This is the first function to call whenever you wish to setup any VPixx device.
2. Call Datapixx ('SetTPxAwake'). This function awakens the TRACKPixx3 and powers up the infrared LEDs of the Illuminator.
3. Call Datapixx ('SetLedIntensity', *intensity*). This function activates the infrared illuminator LEDs to an intensity equal to the *intensity* value you typed in. Allowed *intensity* values are 1-8. We recommend using intensity values higher than 6, but this may vary according to your specific testing environment and the TRACKPixx3 camera aperture.
4. Call Datapixx ('SetExpectedIrisSizeInPixels', *iris size*). This function allows you to set the *iris size* to correctly track the participant's eye movements. **THE IRIS SIZE VALUE IS OF CRITICAL IMPORTANCE!** Setting the wrong iris size will yield incorrect eye-tracking data. The normal range of *iris size* values should be between 50-150 pixels. Recommended values depend on your experimentation setup; refer to the following table.

Table 6 – Iris Size values

TRACKPixx3 lens / distance to participant	Iris size value (pixels)
25 mm / 60 cm	70
50 mm / 60 cm	140
75 mm / 150 cm	100

5. Call Datapixx ('RegWrRd') as required. This function writes the content in the local register cache on the computer to the register on the device, and reads back the current device state.

Performing a calibration in MATLAB

Following the preliminary setup step described above, you can also use MATLAB to calibrate your VPixx equipment.

Pupil Size Calibration

This script has a standalone version which is also implemented in the normal calibration testing script.

Standalone: *TPxPpSizeCal.m*

Complete: *TPxTrackpixx3CalibrationTesting.m* (which is a shortcut to call *TPxCalibrationTesting.m*).

The TPxPpSizeCal stand-alone script can be started with no argument. To call it from another script, you must provide the screen pointer as the first argument (usually called windowPtr).

The second argument is required if you have not set up the TRACKPixx3 in your current script.

Make sure you specify the camera lens and distance to the participant.

Follow the on-screen instructions. A fixation point of variable brightness will be displayed in the middle of the screen. This should cause a response in pupil size, which will be factored in during pupil calibration.

Once the calibration is finished, graphs will be created displaying pupil size according to brightness, which can help you verify that the calibration was successful.

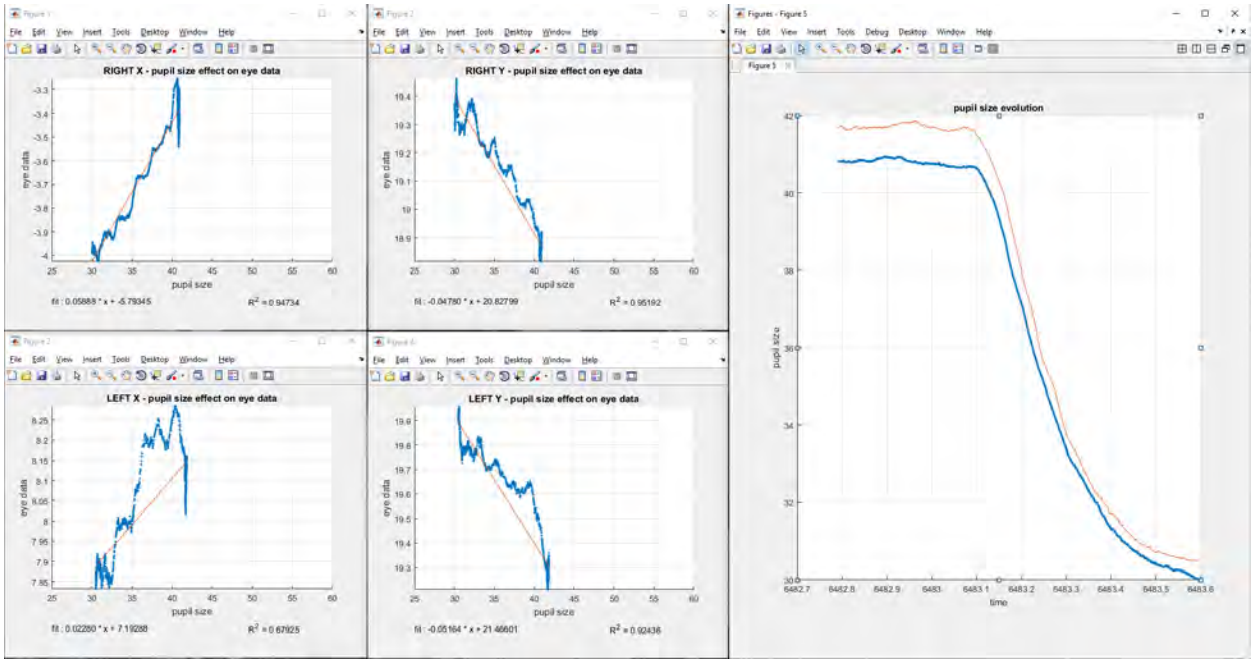


Figure 62 – Pupil Size Calibration graphs

Complete Calibration

Following pupil size calibration, the rest of the calibration procedure can be completed.

1. Define the 13 calibration points to be used. We recommend using points that are approximately equidistant from one another. Here is a pattern that represents an example of a proper calibration point pattern.

In MATLAB, the coordinate system origin is situated at the top left of the screen.

Note that the Datapixx ('ConvertCoordSysToCartesian') command can be used to convert the default MATLAB coordinate system to the VPixx coordinates which places the (0,0) origin at the center of the screen.

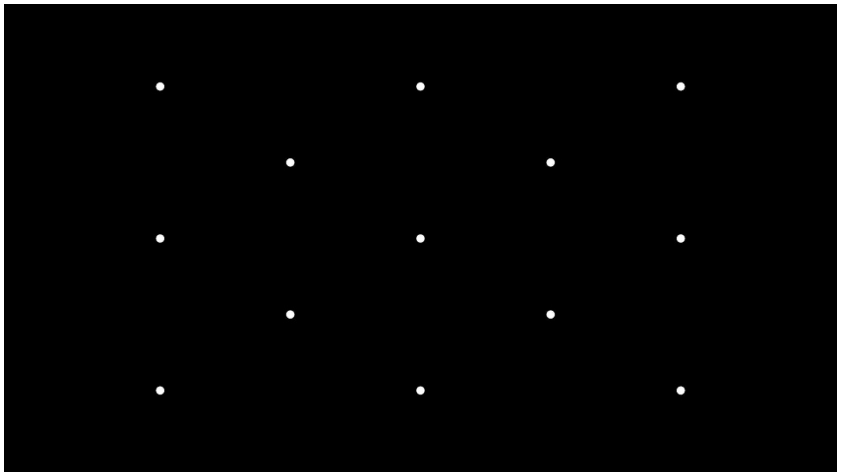


Figure 63 – Calibration point pattern example

2. Display the first point on screen. This is usually done by calling Screen('DrawDots') and Screen('Flip').
3. Wait a moment to allow the user to locate and fixate on the calibration point.
4. Update the local registry cache using Datapixx ('RegWrRd').
5. When ready, call Datapixx('GetEyeDuringCalibrationRaw', xScreen, yScreen). This function tells the tracker to immediately acquire eye data. xScreen and yScreen represent the x and y coordinates of the marker on screen. It then returns the raw coordinates that will be used for the calculations. Note that the coordinate system defining the xScreen and yScreen coordinates will determine the coordinate system for the calibration.
6. Repeat steps 2-5 for all 12 remaining points.

In order to sequence the calibration process, you can use Datapixx('GetTime'). This function returns double precision seconds since DATAPixx powerup.

The KbWaits function can also be used to gain added control over the calibration process.

7. Call Datapixx ('FinishCalibration'). This function makes the necessary calculations and ends the calibration process.
8. Call Datapixx ('ConvertCoordSysToCustom'). This function returns an array of 2 x n elements mapped from the DATAPixx to the user coordinate system.
9. Optionally, call Datapixx ('SaveCalibration') to save the calibration results into the TRACKPixx3 device.

Calibration Validation Script

The calibration validation script allows you to validate the quality of a calibration for the current setup. It will present the same points as the calibration and gather a large gaze dataset for every target. Once the points have finished being presented, the script will display three result screens.

Results 1 of 3: These are the average results for the left and right eyes (in screen pixels). If you wish to translate this into degrees of visual angle, you require the distance between the user and the screen as well as the screen size. With this

information, simply construct a triangle to find the total visual angle for 1920 pixels and convert the result for the obtained pixel values.

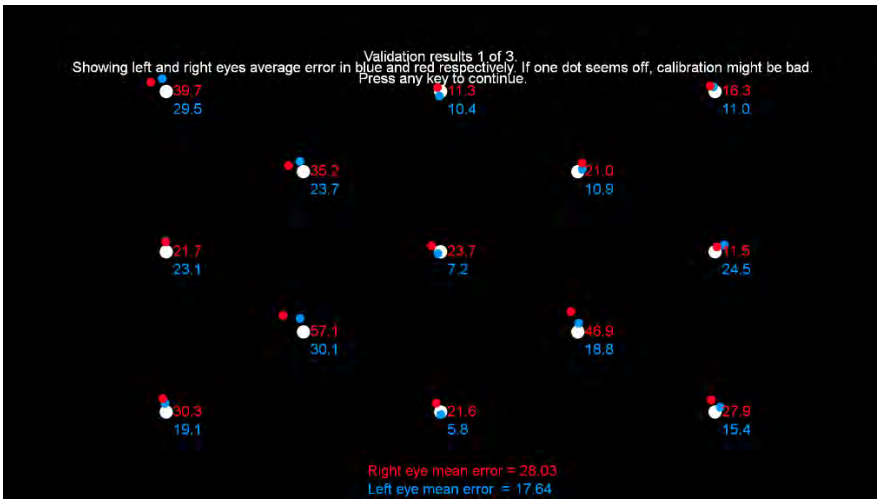


Figure 64 – Calibration Validation results 1 of 3

Results 2 of 3 and 3 of 3: These results present all the data gathered during the presentation of the validation points. A circle is centered on the average value. The radius of the circle is equal to two standard deviations for all the data gathered for this point. The smaller the circle, the more precise the data.

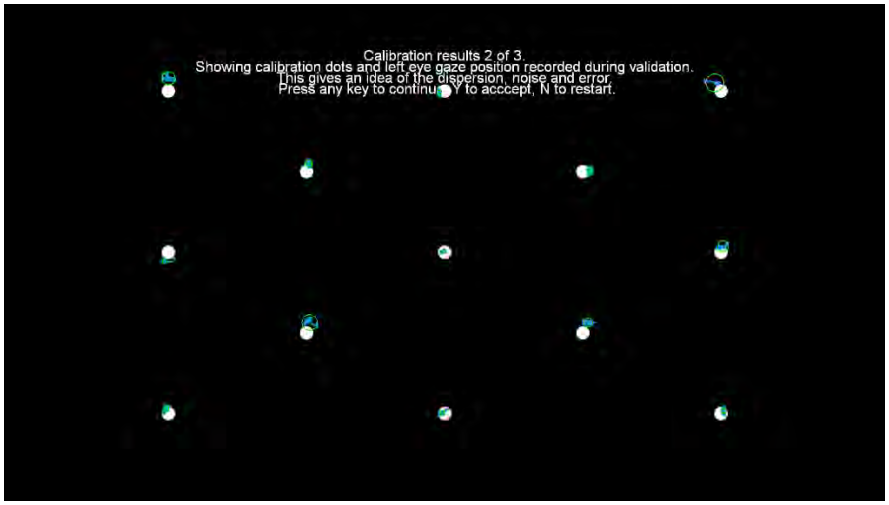


Figure 65 – Calibration Validation results 2 of 3

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 function which finishes the calibration and calculates the mapping between raw data and screen coordinates.

How to send eye positions over an analog interface in MATLAB

The DB25 analog I/O port of the DATAPixx3 allows you to pass X, Y gaze coordinates and pupil diameter to an external system as analog signals. This output is enabled for both DATAPixx3 'Lite' and 'Full' systems.

There are four dedicated digital-to-analog converter (DAC) pins on the analog I/O port. The digital eye position signal recorded by the tracker is converted to a +/- 10 volt analog signal, which is sent out on these pins. The pins can be configured to several types of outputs based on the researchers needs… [see my comments on the PyPixx section]

If you only select only one eye, you will get X, Y and Pupil Diameter on DAC0, 1 and 2, respectively. The Pupil size is in pixels (camera space pixels) and the values of pixel size 0 to 100 are mapped from 0 V to 10 V.

If both eyes are selected for the analog output, the data format is X, Y, X, Y (on DAC0, 1, 2 and 3).

Pin	Description	Pin	Description
1	ADC0	14	ADC1
2	ADC2	15	ADC3
3	ADC4	16	ADC5
4	ADC6	17	ADC7
5	ADC8	18	ADC9
6	ADC10	19	ADC11
7	ADC12	20	ADC13
8	ADC14	21	ADC15
9		22	REF
10		23	+5 V
11	DAC0	24	DAC1
12	DAC2	25	DAC3
13	GND	Shield *	

X eye position coordinates

Y eye position coordinates

TRACKPixx3 analog output control

There are 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, selecting 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 EnableTPxAnalogOut 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.

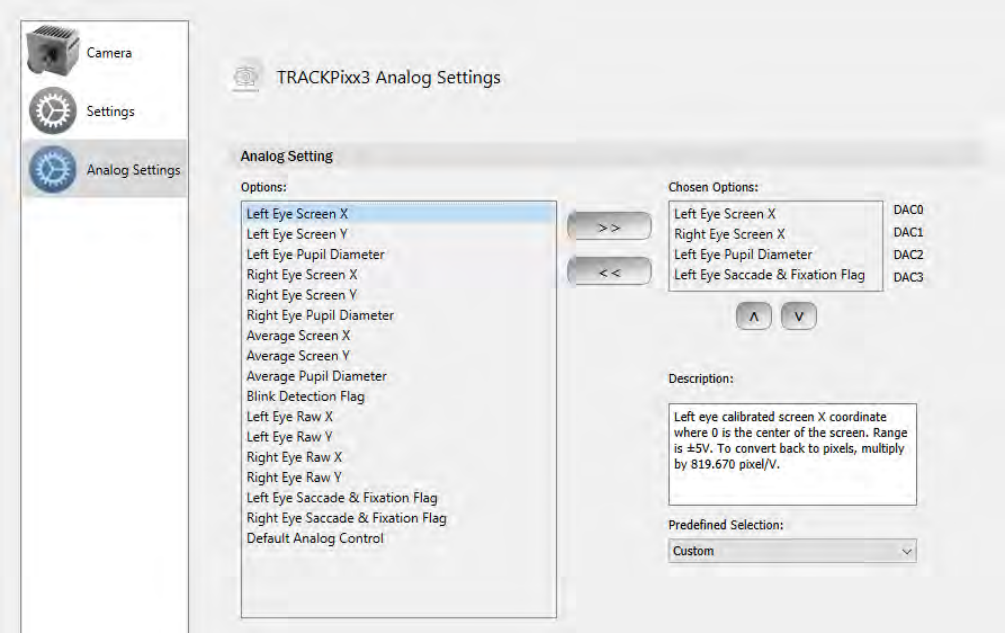


Figure - 66 Analog Settings

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.

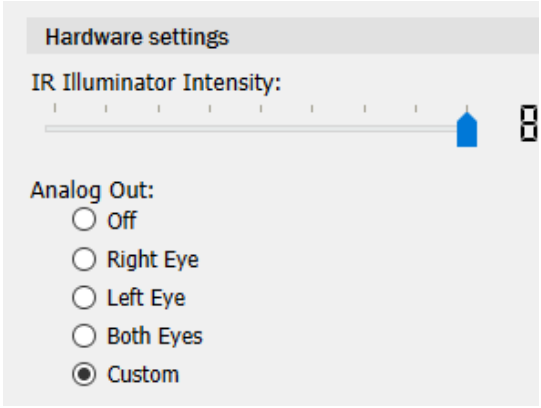


Figure 67 - Hardware Settings

Synchronizing your TRACKPixx3 data with other equipment

Once calibrated, there are two different ways to access participant eye position. First, the user can set up a tracker schedule which continuously records eye position data and saves it to a buffer in the device memory at a rate of 2000 samples per second. To read the contents of this schedule, the user can perform a device register write-read and access a user-specified number of recent frames. This strategy yields the most information but requires a few extra lines of code to implement. Each recorded sample of eye tracking data is a 1 x 20 array of values. The first item in the array is a timetag which indicates the time, in seconds, since the DATAPixx3 was last powered on.

Synchronisation of eye data with Digital Input and Output is automatic, as the tracker schedule records the current 24-bit input and output on each frame in columns 8 and 12, respectively. To synchronise the tracker schedule with other I/O, the timetags listed in the schedule can be compared to the DATAPixx3 timetags associated with other recordings or triggers. For more information on how to get timetags for specific events, refer to the section on “GetTime” and Markers. The second method of accessing eye position data is to simply call ‘GetEyePosition’. This does not require that a schedule be currently running. This method simply returns the calibrated screen position of the eyes, raw eye vectors and a timetag. This data is drawn from the last update of the local register (e.g., the last register write-read). This strategy is useful when continuous recording of eye position is not required, and only eye position is of interest to the user. As with the tracker schedule, timetags can be used to align other events marked on the DATAPixx3 clock with eye tracking data. To create triggers or initiate events based on real-time eye position, the user will need to create a looping function which repeatedly checks eye position for specific criteria (e.g., foveating a target) and sends out triggers or starts events when a specific criterion is met.

If the tracker schedule is running, it is possible to set automatic flags to detect saccades and fixations. The status of these flags can be rapidly checked using the commands “isSubjectFixating” and “isSubjectMakingSaccade” (currently only enabled in MATLAB).

Performing an eye-tracking experiment in MATLAB

Following the proper setup and calibration of your system (see above), or, optionally, by loading a previously-saved calibration, you can now use MATLAB to run and manage your eye-tracking experiment.

All the functions made available by VPixx are called using the following nomenclature:

Datapixx (‘functionname’)

The following table lists the MATLAB functions that may be required to run your experiment.

Table 7 – MATLAB Functions

Function	Arguments [default]	Returns	Description
ClearCalibration	None	None	This function removes any saved calibration in this device.
ConvertCoordSysToCartesian	• *source array* • *offsetX [-960]* • *scaleX [1]* • *offsetY [540]* • *scaleY [-1]*	Converted array	This function takes a 2xn array of positions (x,y) in the coordinate system specified by the offset and scale argument values and returns a converted 2xn array of positions in the VPixx coordinate system (origin at center of screen).
ConvertCoordSysToCustom	• *source array* • *offsetX [-960]* • *scaleX [1]* • *offsetY [540]* • *scaleY [-1]*	Converted array	This function takes a 2xn array of positions (x,y) in the VPixx coordinate system (origin at center of screen) and returns a converted 2xn array of positions in the coordinate system specified by the offset and scale argument values.
FinishCalibration	None	None	Finishes the calibration.
GetCRCoordinatesInPixels	None	• *Left_CR_X* • *Left_CR_Y* • *Right_CR_X* • *Right_CR_Y*	Gets the raw position of the Corneal Reflection (CR) in TRACKPixx space.
GetExpectedIrisSizeInPixels	None	None	Approximation of the iris size in pixels.

Function	Arguments [default]	Returns	Description
GetEyeDuringCalibration	• *xScreen* • *yScreen*	• *Left_CR_X* • *Left_CR_Y* • *Right_CR_X* • *Right_CR_Y*	This function triggers the acquisition of eye data NOW during calibration with the current marker coordinates.
GetEyeDuringCalibrationRaw	• xScreen • yScreen	• *xRawRight* • *yRawRight* • *xRawLeft* • *yRawLeft*	This function triggers the acquisition of eye data NOW during calibration with the current marker coordinates.
GetEyeImage	None	• *image*	This function gets a 2D matrix of the eye image.
GetEyePosition	None	• *xScreenRight* • *yScreenRight* • *xScreenLeft* • *yScreenLeft* • *xRawRight* • *yRawRight* • *xRawLeft* • *yRawLeft*	This function returns the calibrated eye position and raw data.
GetLedIntensity	None	None	Gets the LED intensity.
GetPupilCoordinatesInPixels	None	• *Left_X* • *Left_Y* • *Right_X* • *Right_Y*	Gets the raw position of the pupil in TRACKPixx space.
GetPupilSize	None	• *Left Major* • *Left Minor* • *Right Major* • *Right Minor*	Gets the Major and Minor axis length of the pupil for both eyes. If the returned value is low (approx.0), it indicates a detected

Function	Arguments [default]	Returns	Description
			blink during the measurement span.
GetPupilSizeSimple	None	• *0 if no pupil is found*	Gets the Major axis length of the pupil for one eye.
GetTPxStatus	None	*Refer to the **GetTPxStatus Return Descriptions** table* (Table 8*) for an explanation of what the GetTPxStatus function returns*	Returns a struct containing TRACKPixx input status information.
HideOverlay	None	None	Deactivates DATAPixx3 overlay to hide the eye images and cursors on the monitor.
isSubjectFixating	• *leftFixationFlag* • *rightFixationFlag*	*Returns 1 if the subject is currently fixating, 0 otherwise.*	The TRACKPixx3's schedule must be running. **This requires firmware 14 in the DATAPixx3 and 16 in the TRACKPixx3.**
isSubjectMakingSaccade	• *leftSaccadeFlag* • *rightSaccadeFlag*	*Returns 1 if the subject is currently making a saccade, 0 otherwise.*	The TRACKPixx3's schedule must be running. **This requires firmware 14 in the DATAPixx3 and 16 in the TRACKPixx3.**
LoadCalibration	• *shift_y*[1080]	None	This function verifies if there is a calibration in

Function	Arguments [default]	Returns	Description
			the tracker and if so, it loads it.
ReadTPxData	• *numFrames*	• bufferDAta • underflow • overflow	This function reads the number of samples specified by *numFrames* from the buffer of available data. To read all the available data, use the value **-1**. If the number of samples specified by *numFrames* is greater than the number of samples available in the buffer, the *underflow* flag will be set to **1** and all the data available in the buffer will be returned. If the buffer is filled and starts back to the beginning of the buffer, thus overwriting the first samples, the *overflow* flag is set to **1**. Data is returned in *bufferData* in a two-dimension array of samples having the following format:

Function	Arguments [default]	Returns	Description
			+----------+---------- --+----------+------- ---------+---------- +----------+--------- --------+---------- +----------+---------- ------+----------+--- ---------+----------+----- --------+ | TIMETAG | LEFT EYE X | LEFT EYE Y | LEFT EYE PUPIL DIAMETER| RIGHT EYE X | RIGHT EYE Y | RIGHT EYE PUPIL DIAMETER| DIGITAL INPUT | LEFT EYE BLINK DETECTION | RIGHT EYE BLINK DETECTION | DIGITAL OUTPUT | LEFT EYE FIXATION FLAG | RIGHT EYE FIXATION FLAG | LEFT EYE SACCADE FLAG | RIGHT EYE SACCADE FLAG | MESSAGE CODE | LEFT EYE RAW X | LEFT EYE RAW Y | RIGHT EYE RAW X | RIGHT EYE RAW Y | +----------+---------- --+----------+------- ---------+---------- +----------+--------- --------+---------- +----------+---------- ------+----------+---

Function	Arguments [default]	Returns	Description
			---------+-----------+----------+
			The sample is represented as an array of doubles. The first index of the array is the sample number. The second index is the element of the sample.
SaveCalibration	• *shift_y*[1080]	None	This function verifies if the system is currently calibrated and if so, saves the results in the device.
SetExpectedIrisSizeInPixels	• *IrisSize*		Usual values: 70 @ 60 cm with 25 mm lens 140 @ 60cm with 50mm lens 150 for MRI setup
SetFixationThresholds	• *maxSpeed* *minNumberOfConsecutiveSamples*	None	Set the fixation thresholds for the TRACKPixx schedule. The maximum speed is in pixels per seconds and the number of samples is at 2KHz. **This requires firmware 14 in the DATAPixx3 and 16 in the TRACKPixx3.**

Function	Arguments [default]	Returns	Description
SetSaccadeThresholds	• *minSpeed* *minNumberOfConsecutiveSamples*	None	Set the saccade thresholds for the TRACKPixx schedule. The minimum speed is in pixels per seconds and the number of samples is at 2KHz. **This requires firmware 14 in the DATAPixx3 and 16 in the TRACKPixx3.**
SetLedIntensity	• *SetLedIntensity* • *LedIntensity*	None	Changes the LED intensity from 0-8.
SetTPxAwake	None	None	Awakens the tracker and turns on the infrared illuminator.
SetTPxSleep	None	None	Puts the tracker to sleep and shuts down the infrared illuminator.
SetupTPxSchedule	• *bufferbaseAddress*[12e6] • *numberOfEyeData*[3600000]	None	This function sets up the base address and buffer size.
ShowOverlay	None	None	Activates DATAPixx3 overlay to display the eye images and cursors on the monitor.
StartTPxSchedule	None	None	Starts the recording of camera data in a buffer specified by the SetupTPxSchedule command.
StopTPxSchedule	None	None	Stops the recording.

Table 8 – GetTPxStatus return descriptions

Return value	Description
BufferBaseAddress	Acquisition of data buffer base address within DATAPixx.
BufferSize	Number of bytes in the acquisition data buffer.
CurrentReadAddr	Buffer address which will be read by the next streaming call to TPxGetData.
CurrentReadFrame	Buffer frame which will be read by the next call to TPxGetData.
CurrentWriteAddr	Buffer address into which will be written the next acquired TPx sample.
CurrentWriteFRame	Buffer frame into which will be written the next acquired TPx sample.
IsRecording	Acquisition is currently running.
IsLogTimetag	Acquisition of timetags in addition to eye-tracking data.
NewBufferFrames	CurrentWriteFrame-CurrentReadFrame
numBufferFrames	Total number of samples which fit in the acquisition data buffer.
NumStreamOverflows	Number of ReadMicrophoneBuffer streaming overflows which have occurred since SetTPxSchedule.
NumStreamUnderflows	Number of ReadMicrophoneBuffer streaming underflows which have occurred since SetTPxSchedule.

At the end of your experiment, call Datapixx ('Close'). This function closes the DATAPixx.

Refer to the VPixx-supplied MATLAB eye-tracking experiment example located here:

VPixx Software Tools\DatapixxToolbox_trunk\DatapixxToolbox\DatapixxDemos

The demo is called either TPxTrackpixxCalibrationTesting.m or TPxTrackpixxMiniCalibrationTesting.m depending on the type of TRACKPixx system you are using

Stimulus Presentation

PROPixx

VPixx Technologies' **PROPixx** system offers vision researchers a precise stimulus presentation package that may be central to the advancement of their research.

Product overview

The PROPixx is a unique DLP LED projector designed to be the most flexible display possible for vision research. It features a native resolution of 1920 x 1080 and can be driven with refresh rates up to 500 Hz (RGB mode) or 1440 Hz (Greyscale mode) with deterministic timing.

The PROPixx uses high brightness LEDs as a light source, giving a larger color gamut and much longer lifetime than halogen light sources (60,000 hrs vs 2,000 hrs). The LEDs also support high bit depth and high frequency full color stimulation which would not be possible with a color-wheel/halogen architecture.

An important aspect of the PROPixx is that its **gamma (γ) is exactly 1**.

For stereo vision applications, our high-speed circular polarizer can project stereoscopic stimuli for passive polarizing glasses at up to 400 Hz.

The PROPixx comes equipped with an array of peripherals which often need to be synchronized to video during an experiment. These peripherals include a stereo audio stimulator, a button box port for precise reaction-time measurement, triggers for electrophysiology and eye-tracking equipment, and even a complete analog I/O subsystem. You can now successfully synchronize all of your I/Os to video refresh with microsecond precision.

The PROPixx is available with multiple projection lens options including short-throw lenses for CRT replacement applications, and long-throw lenses for MRI/MEG applications.

PROPixx Controller

Other than the projector itself, your PROPixx system also includes a controller for synchronized data acquisition. The PROPixx projector requires the PROPixx controller to function.

Video and I/O synchronization

The PROPixx displays video information with no processing delay, and the I/O subsystems have microsecond-precise synchronization with the incoming video vertical sync pulse. The system waits until it has received a complete video frame, then immediately presents all raster lines simultaneously (rather than being scanned from top to bottom) without delay.

For more information on I/O, refer to the I/O Monitoring and Generation section.

DVI Out

A second display can be connected to the DVI Out port. This secondary display can be used, for example, by a remote operator wishing to monitor the stimuli being presented to a participant.

LED light source

The PROPixx light engine uses 3 big chip LEDs that can deliver a large color gamut to the screens. The following CIE 1931 color space diagram shows the x-y color coordinates of the PROPixx LEDs. The peak wavelengths are Red: 623 nm, Green: 525 nm, Blue: 460 nm

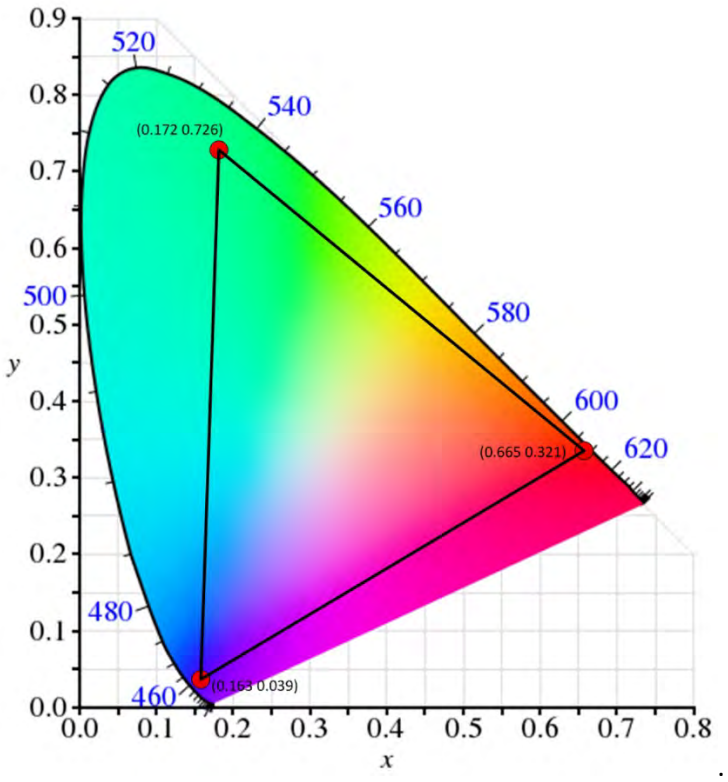


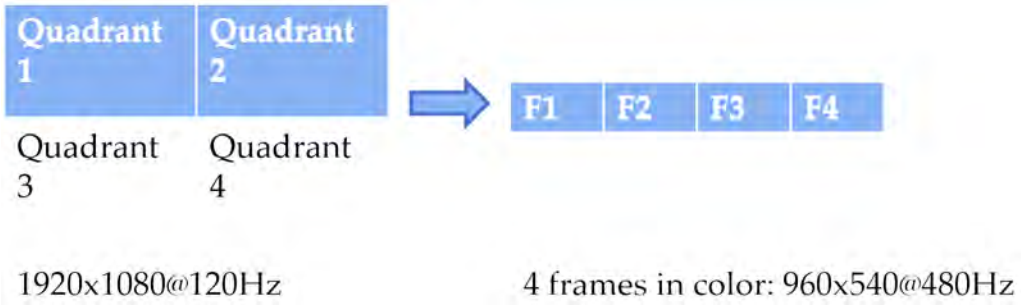
Figure 68 – CIE 1931 Color Space Diagram

High Refresh Rate

Some of the available sequencer modes for the PROPixx include high refresh rate (over 120 Hz) options. These are described below.

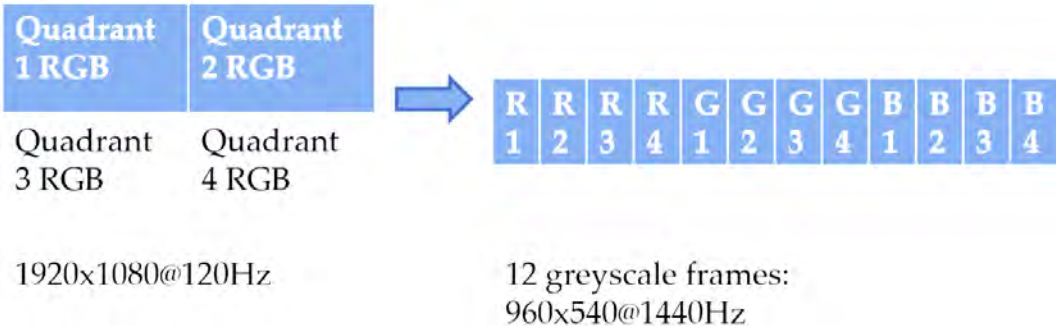
480 Hz (QUAD4X)

This mode uses a 1920x1080@120 Hz input to create 4 frames to display sequentially on the PROPixx.



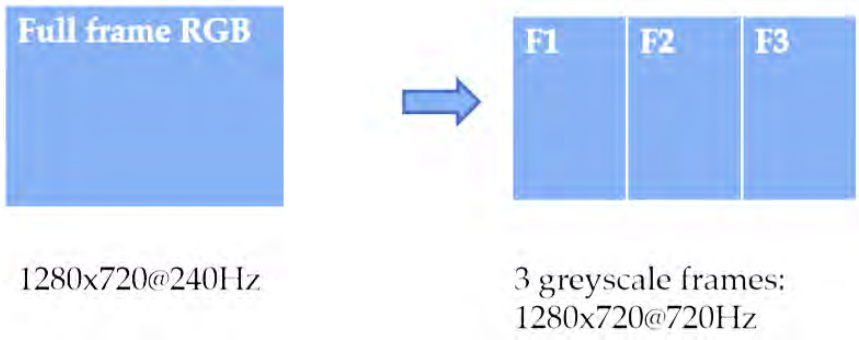
1440 Hz (QUAD12X)

This mode uses a 1920x1080@120 Hz input to create 12 frames to display sequentially on the PROPixx.



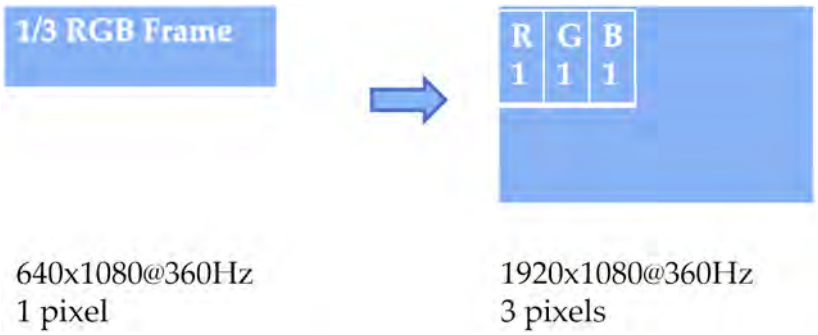
720 Hz (3D Compatible) (GREY720)

This mode uses a 1280x720@240 Hz input to create 3 frames to display sequentially on the PROPixx.



360 Hz (3D Compatible) (GREY3X)

This mode uses a 640x1080@360 Hz input to create 1 frame 1920x1080@360 Hz frame. Every frame has a black frame of equal length inserted in between.



RB3D (3D Only)

Red-Blue 3D mode (RB3D) is a grayscale 3D mode in which we create a 3D stimulus using blue color information for the left eye image and red color information for the right eye image. This mode can be run at any refresh rate under or equal to 120 Hz. Each frame will show the left eye image for 1.2 ms, right eye image for 2.4 ms, then the left eye image for another 1.2 ms. The rest of the frame is blank (these time values are only valid for 120 Hz video input).

Note that the following demos can be found by logging in to your MyVPixx account and navigating to https://vpixx.com/myvpixx/downloads-and-updates/

Powering up your PROPixx

Once all cables have been properly connected, you can power on your PROPixx. The power up procedure is as follows:

1. Toggle the controller's power switch to the ON position.
2. On the projector, press the ⏻ button once to switch the projector ON.

Pressing the projector's ⏻ button continually for 3 seconds or more will put it in SLEEP mode.

Status LED

Your PROPixx is equipped with two status LEDs that supply information on the energy or power status of your system. The information relayed by each LED, one blue and one red, is summarized in the table below.

Table 9 – Status LEDs

	ON	Blinking
BLUE LED	Awake Mode	LED Off mode
RED LED	Sleep Mode	Thermal Shutdown

Remote controller

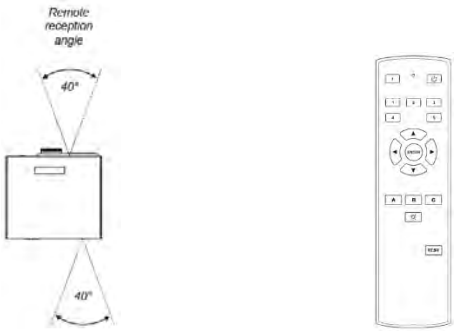


Figure 69 – Remote controller

Table 10 – Remote controller buttons

	Function	Description
	Power ON	Press the remote control's POWER ON button to switch the projector ON
	Power OFF	Press the remote control's POWER OFF button to switch the projector to SLEEP mode
	LED ON/OFF	Turn ON/OFF LED light source

Installation test pattern

To validate the installation and screen geometry, use your PROPixx projector in *Test mode*. To enter Test mode, simply press the **TEST** button followed by the **A** button on the remote control. You can do this same sequence for tests B and C.

When in Test mode, press the **TEST** button to return to Normal mode.

Configuring your PROPixx and PROPixx Controller using PyPixx

PyPixx is a Python application allowing a user to configure any VPixx Technologies device. It can also provide information on the devices, perform calibrations and calibration tests and run demos. PyPixx operates on Windows, Mac and Linux operating systems.

This section briefly presents the PyPixx operating environment. It introduces the environment's menus, tabs, toolbars and status bars. The figure below shows PyPixx's main window when the application is first opened (for purposes of this example, we connected two VPixx devices: a PROPixx projector and its accompanying PROPixx controller).

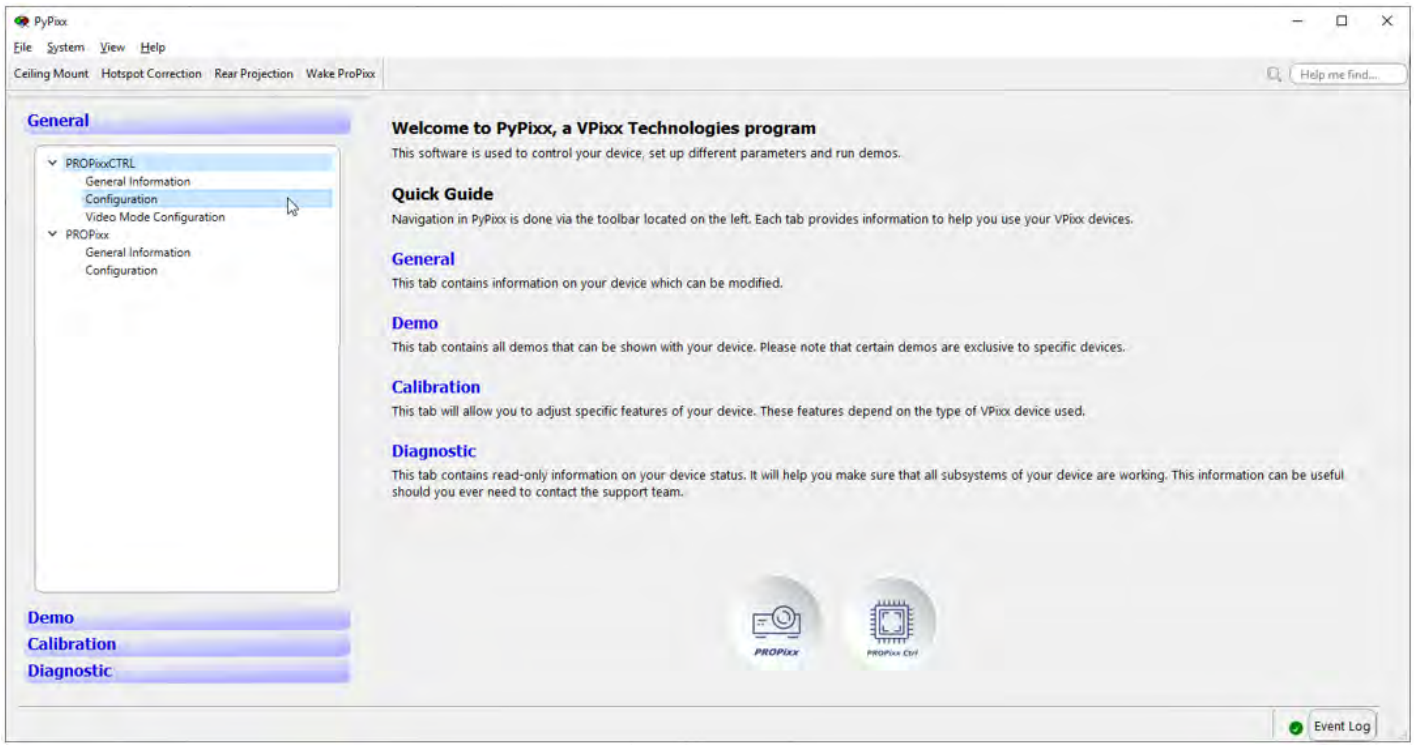


Figure 70 – PyPixx main window (connected PROPixx projector and controller)

Note that if your controller is disconnected or not powered, the PyPixx application will display the following information, letting you know that the controller must be properly connected and powered for you to use your PROPixx projector.

Figure 71 – PyPixx main window (improperly connected PROPixx controller)

The main window's left portion is divided into four tabs: *General, Demo, Calibration* and *Diagnostic*. Each tab displays the VPixx devices currently powered and connected to the computer.

- The *General* tab contains VPixx device information and allows you to configure certain device features.
- The *Demo* tab contains demos relevant to your device(s). Please note that certain demos are exclusive to specific devices.
- The *Calibration* tab allows you to recalibrate analog sub-systems in your device(s). These features depend on the type of VPixx device(s) used.
- The *Diagnostic* tab contains read-only information pertaining to your device status. It allows you to verify that all of your device subsystems function properly. This information can be useful if you contact the support team.

Central window

The central window provides information specific to the active device and the actions that can be performed on it. The figures below show PyPixx's Central window when two devices are detected (a PROPixx and PROPixx Controller).

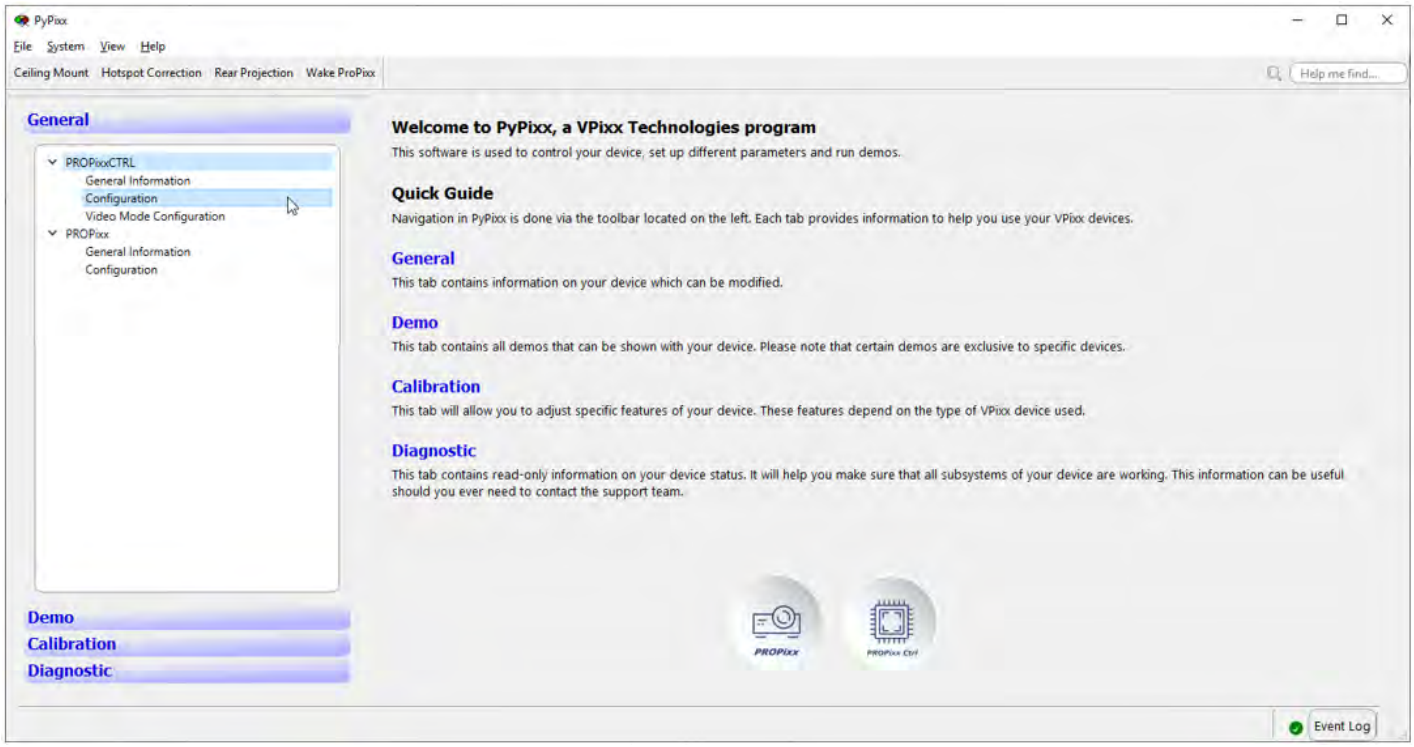


Figure 72 – Central Window (detected PROPixx and PROPixx Controller)

Note that depending on which device you are using and which tab you clicked on, the central window will display a different message.

Status bar

The status bar at the bottom of the main window provides additional information to the user as to the functioning of PyPixx and the devices currently detected by the application. The status bar will also warn the user when the device is busy processing instructions.

Figure 73 – PyPixx status bar

Should an error occur on a VPixx device or in PyPixx, information related to this error may be obtained by clicking on the *Event Log* button.

General tab

The General tab contains setup information for your PROPixx device.

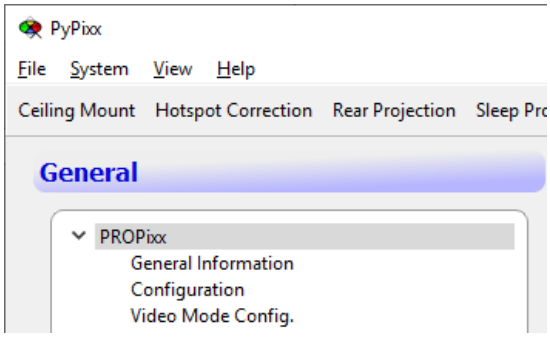


Figure 74 – General tab – PROPixx device

General Information

Clicking on **General Information** will open the PROPixx device's *General Information* window. In this window you will be able to see information concerning your PROPixx device such as its device ID, revision, firmware version, etc.

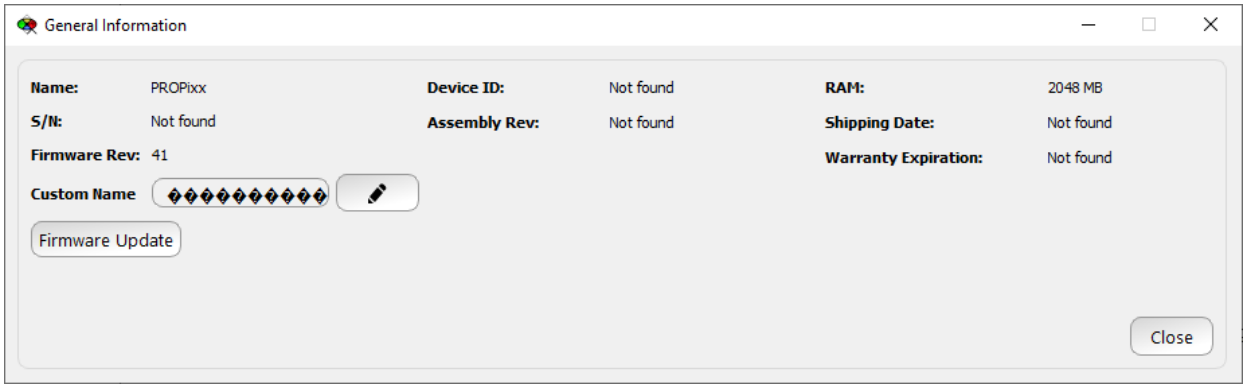


Figure 75 – General Information Window – PROPixx device

Clicking on the **Firmware Update** button will open the *Firmware Update* window allowing you to download the latest firmware update for your PROPixx.

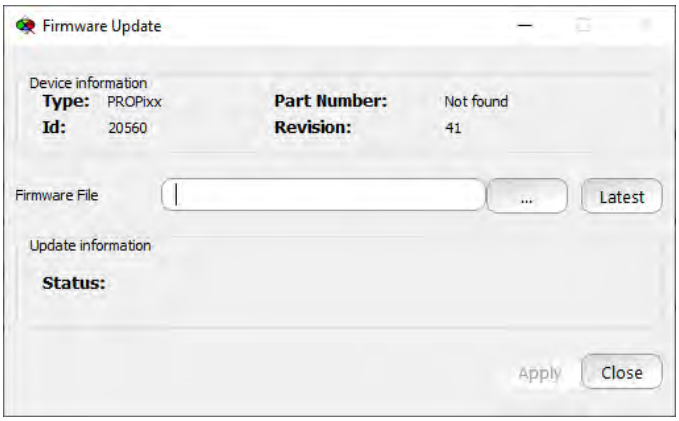


Figure 76 – Firmware Update Window – PROPixx device

From this window you can download the latest firmware file by clicking on the **Latest** button or choose a specific firmware file by clicking on the "…" button and selecting the desired file. Once this is done, click on the **Apply** button to save your changes.

Configuration - PROPixx

Clicking on **Configuration** for the PROPixx will open the *PROPixx Configuration* window.

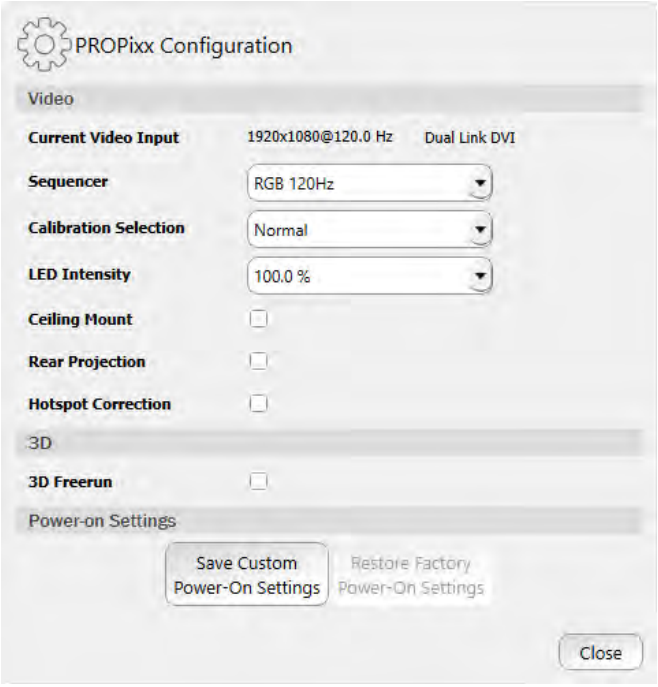


Figure 77 – PROPixx Configuration Window

Sequencer allows you to select which type of projection will be displayed. To generate images, the PROPixx will flash its LEDs for different time intervals. The total time where the LEDs are on and the mirrors pointed towards the screen gives the final intensity over a full period. The process handling this is called the *Sequencer*. Since many video modes and refresh rates are available, many sequencers are also available.

Figure 78 – Sequencer options

Refer to the **High Refresh Rate** *section for more information on sequencer modes*

Calibration Selection allows you to choose between either *normal, Custom RGB#1* or *Custom RGB#2*) according to your experimental needs.

The *LED Intensity* allows you to adjust the projector's LED intensity.

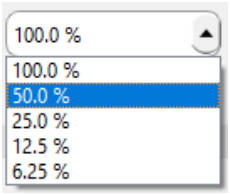


Figure 79 – Selecting LED intensity

Checking either the *Ceiling Mount* or *Rear Projection* boxes will flip the projected image. *Ceiling Mount* will flip it vertically, while the *Rear Projection* option will flip it horizontally.

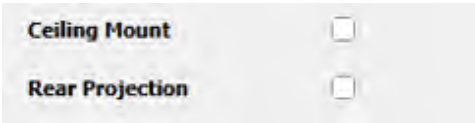


Figure 80 – Ceiling mount / Rear projection checkboxes

Checking the *3D Freerun* box will enable the polarizer to be driven regardless of the chosen mode.

Configuration – PROPixx Controller

Clicking on **Configuration** for the PROPixx Controller will open the *PROPixx Controller Configuration* window.

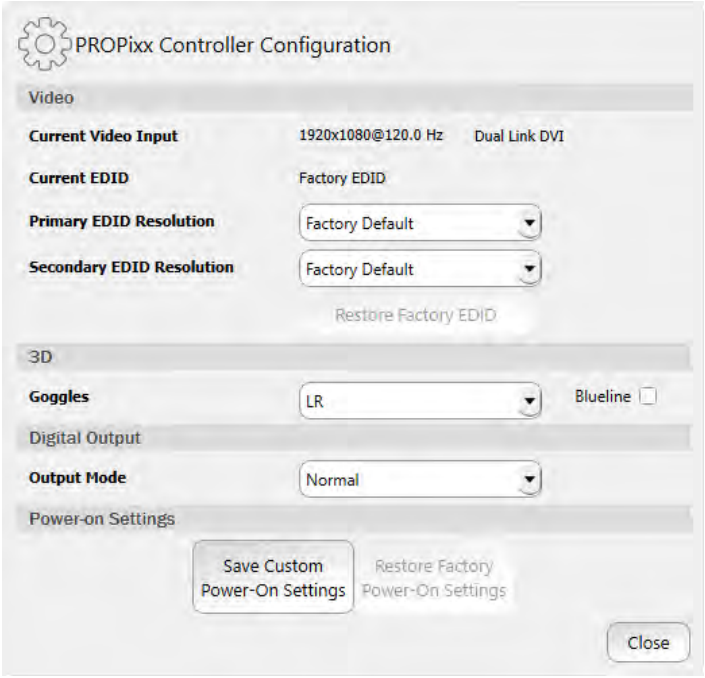


Figure 81 – PROPixx Controller Configuration Window

Both *Primary EDID Resolution* and *Secondary EDID Resolution* menus allow you 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.

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 (not PROPixx Controller) configuration screen

Select the proper setting in the *Goggles* menu if you use active glasses (3DPixx) with the controller. Check the *Blueline* box to enable 3D viewing using a VIEWPixx/DATAPixx2.

Output Mode allows you to select the resolution of the current video being sent to the controller.

Video Mode Config.

There are different video modes available when running the PROPixx at its native resolution of 1920x1080@120Hz.

From the **General** tab, clicking on **Video Mode Config.** will open the *Video Mode Configuration* window which allows you to select the video mode.

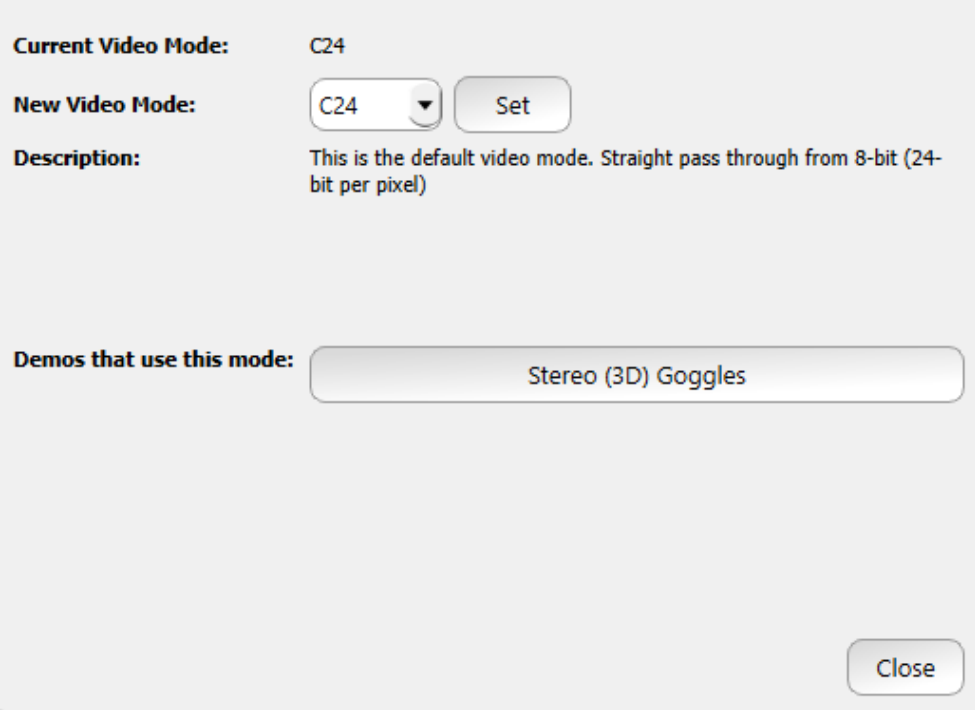


Figure 82 –Video Mode Config Window

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.

Clicking on the *New Video Mode* arrow will display the available video modes you can choose from.

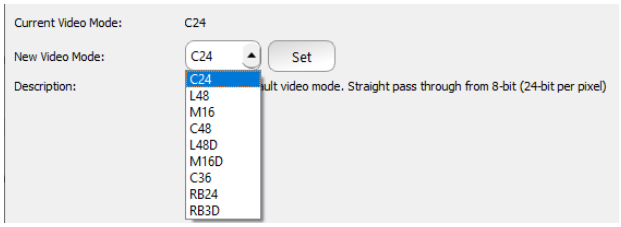
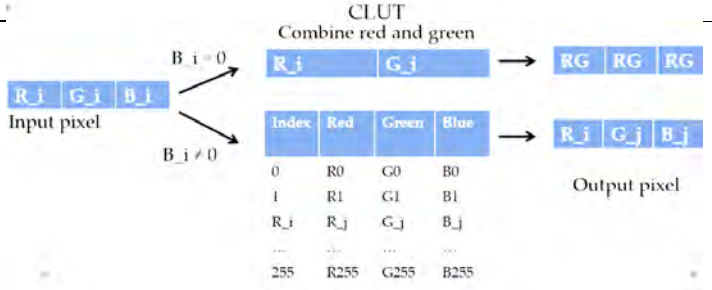
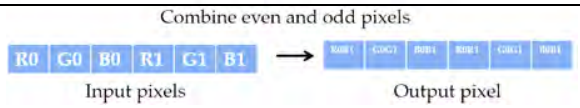


Figure 83 – New Video Mode selection

The following table gives a short description of the available video modes.

Table 11 – Video Mode descriptions

Mode	Description	Representation (if required)
C24	Default Mode.	
L48	L48 is a video mode for high bit-depth color. It allows for 256 16-bit colors using a CLUT*. You send your CLUT to the PROPixx and then you set the pixel's red color to the index for the color to be used.	
M16	M16 is a video mode for high bit-depth grayscale. It allows for 16-bit grayscale color as well as an overlay using a CLUT. The red and green values (8bits) are combined to create a 16-bit grayscale value. If the blue is non-zero it is used as an index to the CLUT.	
C48	C48 is a video mode for high bit-depth color. It sacrifices spatial resolution to gain bit-depth. It combines even and odd 8-bit pixels to create duplicate 16-bit pixels.	
L48D	L48D is a video mode for high bit-depth color. It allows for 256 16-bit colors using a CLUT. You send your CLUT to the PROPixx and then you use the pixel's red and green color to generate the index for the color to be used.	
M16D	M16 is a video mode for high bit-depth grayscale. It allows for 16-bit grayscale color as well as an overlay using a CLUT. The red, green and blue values are combined to create a 16-bit grayscale value.	

Mode	Description	Representation (if required)
C36D	C36D is a video mode for high bit-depth color. It sacrifices spatial resolution to gain bit-depth. It combines even and odd pixels to create duplicate 12-bit pixels (it ignores the lowest 4 bits of every color).	

(*): Color look-up tables (CLUTs) are a part of the video pipeline. You can have CLUTs for color correction, gamma correction, color removal, etc. A CLUT is usually a 256 by 3 matrix, where each column represents a color channel (red, green, blue) and each row represents an operation or a new color value for that color.

After choosing a new video mode, a short description will be displayed to help you decide if this mode is appropriate for your experiment.

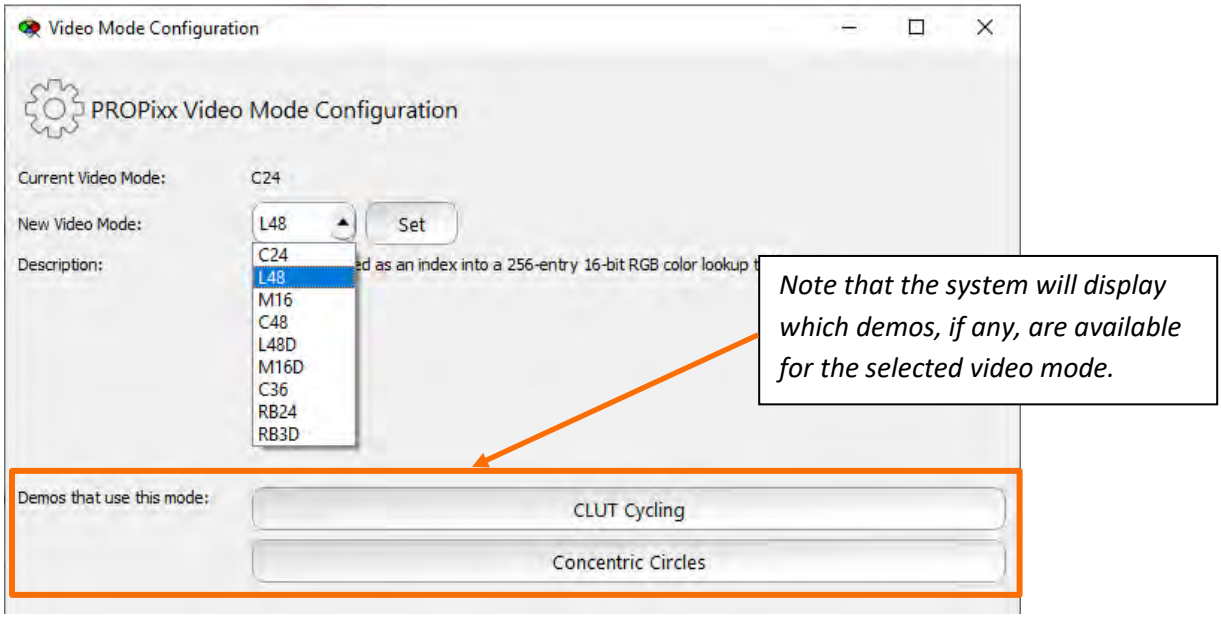


Figure 84 – Available demos (for L48 video mode)

Demo tab

PyPixx contains a multitude of PROPixx demos that may be used to quickly preview the functionality of your PROPixx device. To access the demos, click on the *Demo* tab. This will display the PyPixx demo window for your PROPixx. Simply select the demo you wish to run and follow the on-screen instructions.

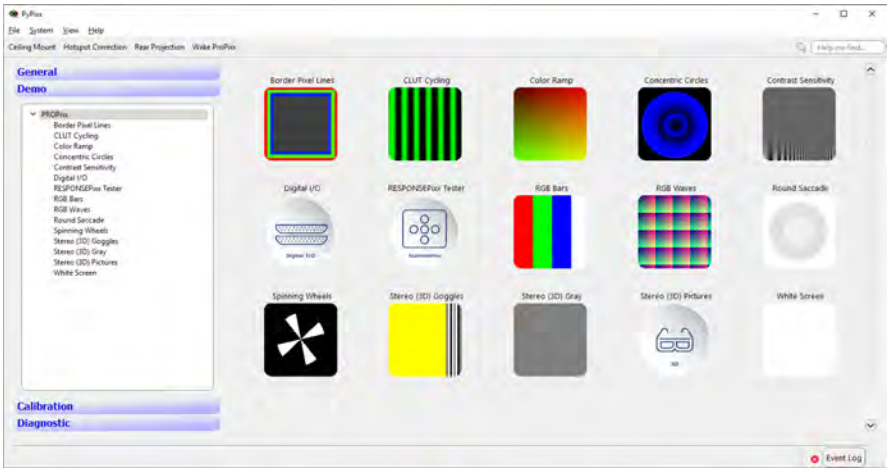


Figure 85 – Demo tab – PROPixx device

As previously mentioned, depending on your selected video mode, some demos may not be available.

Calibration tab

The Calibration tab contains the calibration options that are available for your currently connected VPixx products. In the case of the PROPixx, the available calibration procedure is *White Calibration* and *Hotspot Correction* and both procedures require an I1Display from X-Rite (refer to the

Calibrating your Stimulus Presentation Equipment chapter for more information).

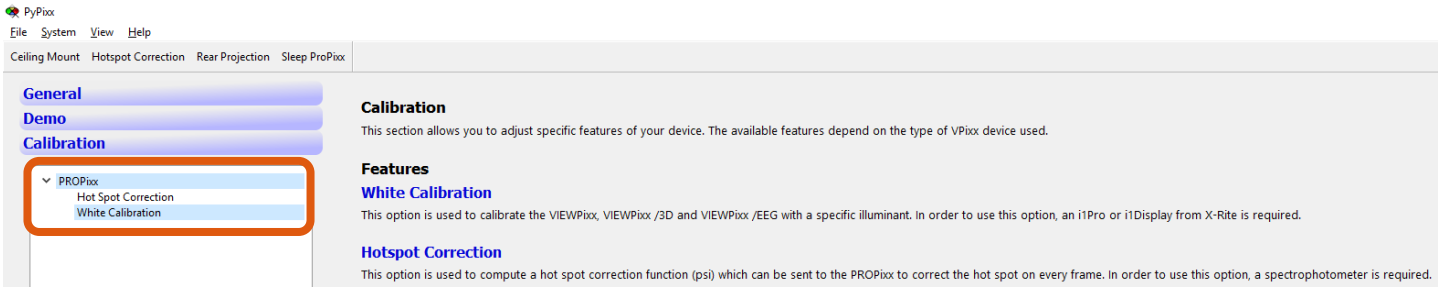


Figure 86 – Calibration tab – PROPixx device

Click on *White Calibration* to open the *PROPixx Calibration* window.

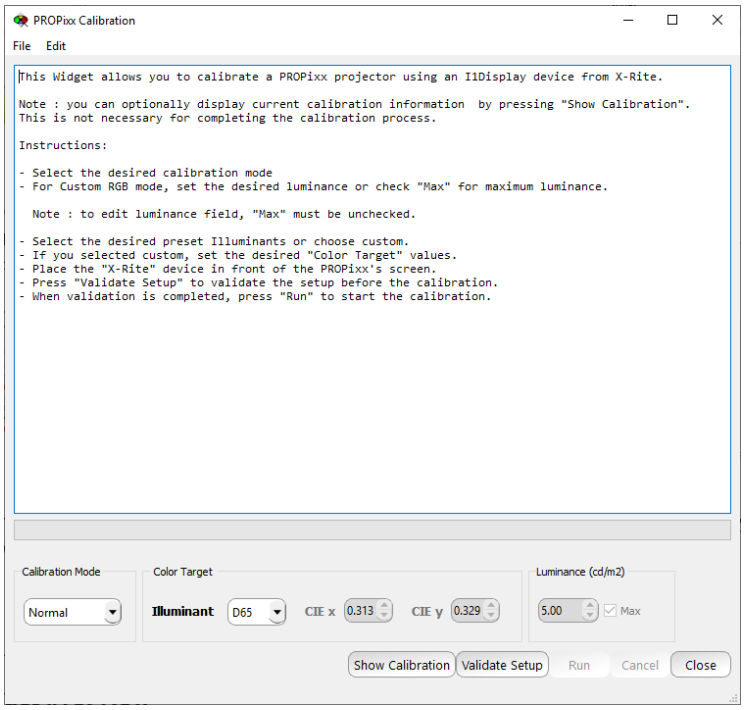


Figure 87 –PROPixx Calibration window

From this window, you can access and adjust all of the PROPixx's white calibration parameters. Simply follow the on-screen instructions to complete the white calibration.

Clicking on *Hot Spot Correction* will cause the PROPixx projector to display a set of on-screen instructions for you to follow in order to perform the hot spot correction.

Diagnostic tab

The Diagnostic tab allows you to verify that all of your currently connected VPixx products are functioning correctly. If they are not, the information gathered here will be useful when you contact the VPixx support team.

In the case of the PROPixx, the available diagnostic tools are explained in the figure below.

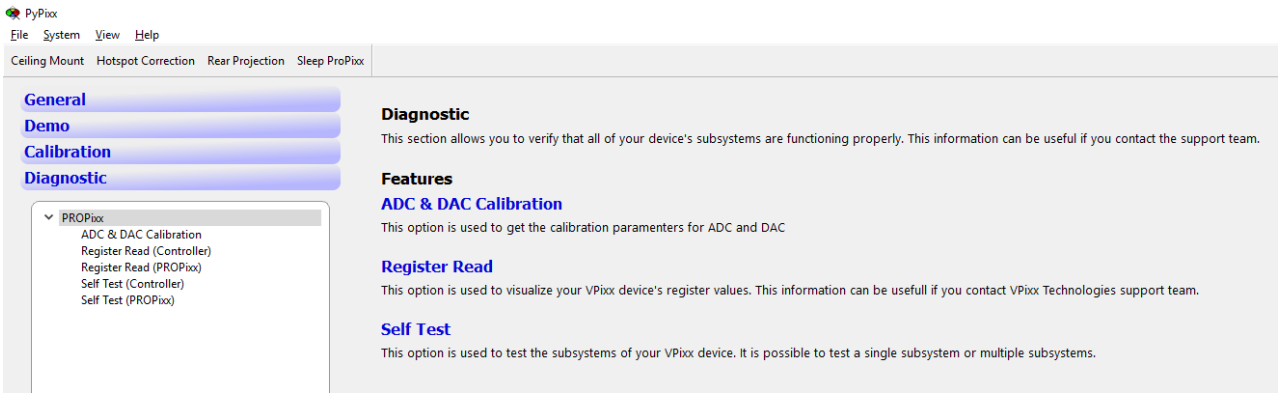


Figure 88 – Diagnostic tab – PROPixx device

The *ADC & DAC Calibration* and *Register Read* tests allow you to update the specific diagnostic information and send it to the VPixx support team if required.

Clicking on one of the *Self Test* options (either for the PROPixx projector or controller) allows you to select exactly which self-diagnostic test(s) you wish to run for your PROPixx. If you wish to run a complete test, activate all the checkboxes and click on the **Run** button. The green progress bar from the Selftest window displays the test's progress and when completed, the window will display the results as shown below. These may be relayed to the VPixx support team if requested.

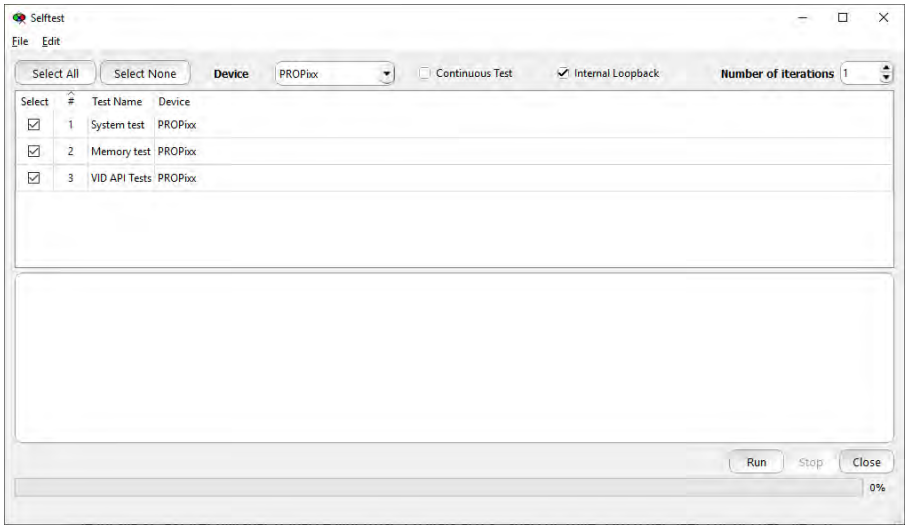


Figure 89 – PROPixx Self Test

VIEWPixx

Product overview

The VIEWPixx is a complete display toolbox which has been conceived specifically to replace CRTs in vision science labs. The VIEWPixx features high-performance industrial LCD glass and a panel controller which has been custom designed to support vision research. Our innovative LED backlight design eliminates the long warmup delay required by CRTs and

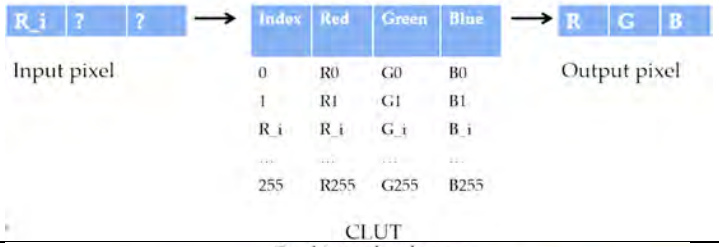
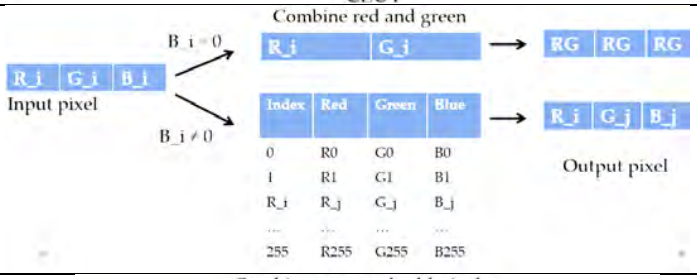
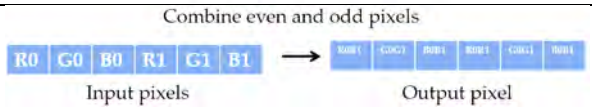
other LCD backlight technologies. In addition, the VIEWPixx includes an array of peripherals which often need to be synchronized to video during an experiment, including a stereo audio stimulator, a button box port for precise reaction-time measurement, triggers for electrophysiology equipment and a complete analog I/O subsystem. Since the video controller and peripheral control were implemented on the same circuit board, you can successfully synchronize all of your participant I/Os to video refresh with microsecond precision.

As with all VPixx Technologies products, the VIEWPixx is field upgradable. If you require a new feature in order to follow some exciting new direction in your research, we can develop the functionality in our labs, then promptly email you an update for your VIEWPixx.

Video Mode

The following table gives a short description of the available video modes for the VIEWPixx.

Table 12 – VIEWPixx Video Mode descriptions

Mode	Description	Representation (if required)
C24	Default Mode.	
L48	L48 is a video mode for high bit-depth color. It allows for 256 16-bit colors using a CLUT*.	
M16	M16 is a video mode for high bit-depth grayscale. It allows for 16-bit grayscale color as well as an overlay using a CLUT. The red and green value (8bits) are combined to create a 16-bit grayscale value. If the blue is non zero it is used as an index to the CLUT.	
C48	C48 is a video mode for high bit-depth color. It sacrifices spatial resolution to gain bit-depth. It combines even and odd 8-bit pixels to create duplicate 16-bit pixels.	
L48D	L48D is a video mode for high bit-depth color. It allows for 256 16-bit colors using a CLUT.	
M16D	M16 is a video mode for high bit-depth grayscale. It allows for 16-bit grayscale color as well as an overlay using a CLUT. The red, green and blue values are combined to create a 16-bit grayscale value.	
C36D	C36D is a video mode for high bit-depth color. It sacrifices spatial resolution to gain bit-depth. It combines even and odd pixels to create duplicate 12-bit pixels (it ignores the lowest 4 bits of every color).	

(*): Color look-up tables (CLUTs) are a part of the video pipeline. You can have CLUTs for color correction, gamma correction, color removal, etc. A CLUT is usually a 256 by 3 matrix, where each column represents a color channel (red, green, blue) and each row represents an operation or a new color value for that color.

After choosing a new video mode, a short description will be displayed to help you decide if this mode is appropriate for your experiment.

VIEWPixx image processing and I/O synchronization

The VIEWPixx has two ways of treating incoming video. For timing-critical research, the incoming video should have a resolution of 1920 x 1200 (1920 x 1080 for VIEWPixx /3D and VIEWPixx /EEG) and a refresh rate of 100-120 Hz. Under these conditions, the VIEWPixx displays the video with no processing delay and the I/O subsystems have microsecond-precise synchronization with the incoming video. If the resolution drops below 1920x1080 or the refresh rate drops below 100 Hz, then the VIEWPixx operates as a normal LCD panel. Incoming video is scanned into an internal frame buffer, then rescanned out to the display. This mode should only be used when not running timing-critical experiments. **In these cases, the VIEWPixx warns the user that it is in "resampling" mode by flashing a small red square in the top-left corner of the display**.

For more information on I/O, refer to the I/O Monitoring and Generation section.

DVI Out

A second display can be connected to the DVI Out port. This secondary display could be used by a remote operator to monitor the stimuli being presented to a participant. Output resolution is a copy or an upscale of the input resolution. Therefore, the output resolution is always 1920 x 1080 @ 120 Hz.

This feature is only available for VIEWPixx and VIEWPixx /3D. Only specific LCD models work with the VIEWPixx as a console output

Backlight

The VIEWPixx backlight uses a matrix of RGB LEDs. RGB LEDs can deliver a large color gamut to screens. When using three separate LEDs (additive color) the backlight can produce a color spectrum that closely matches the color filters in the LCD pixels themselves.

The following CIE 1931 color space diagram shows the x-y color coordinates of the VIEWPixx LEDs. The peak wavelengths are Red: 630 nm, Green: 527 nm, Blue: 470 nm.

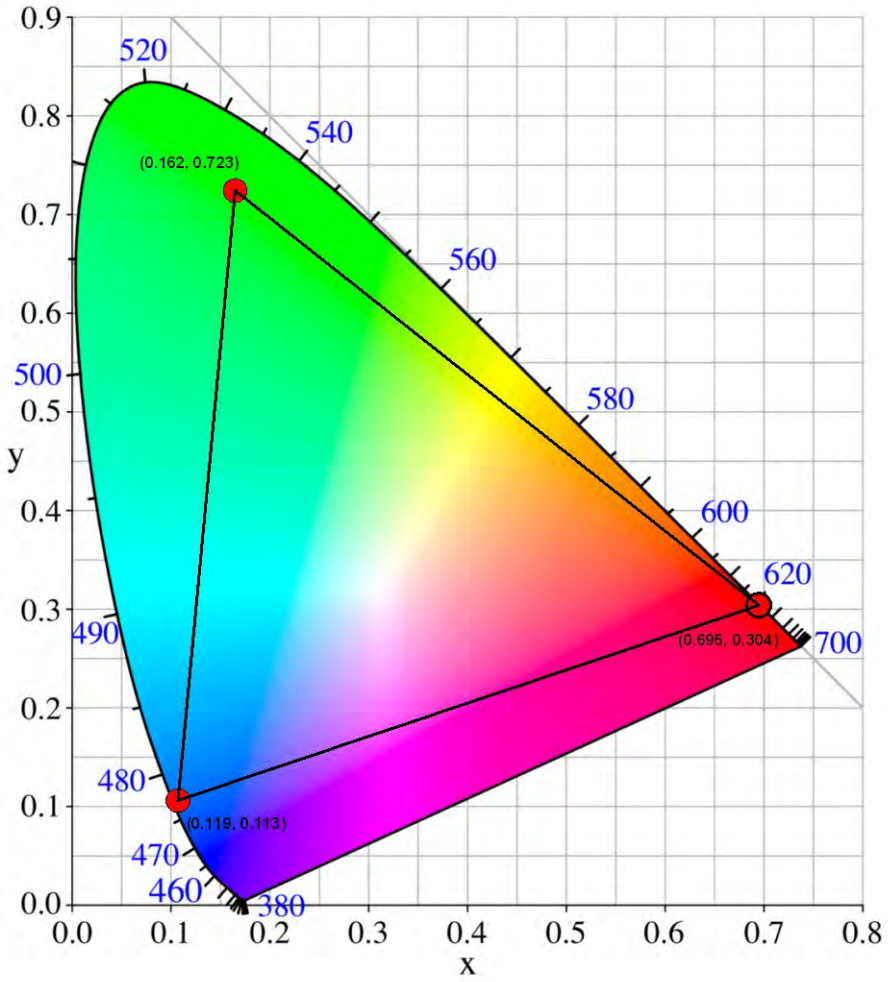


Figure 90 – CIE 1931 Color Space Diagram

Luminance and white point

The VIEWPixx is factory calibrated to 250 cd/m^2 for standard backlight mode and 100 cd/m^2 when scanning backlight mode is active. In both cases, the white point is factory set to D65, with a native gamma very close to 2.2.

If the backlight mode is changed (standard mode vs. scanning backlight mode) VPixx Technologies recommends a transition time of 20 minutes between your experiments.

You can change the backlight intensity with the *bli* command of vputil. By reducing the intensity for low luminance stimuli, you maintain the high-bit-depth video for the entire luminance range of your stimulus. The bli command is also supported with low-level ANSI C API, MATLAB/Octave and Python libraries.

*For more information on the bli command, please refer to the **ANNEX – VPUTIL SOFTWARE** section.*

LED configuration

The following diagram shows the LED matrix configuration of the VIEWPixx. A total of 1344 LEDs are distributed into 56 cells to obtain excellent uniformity on the display.

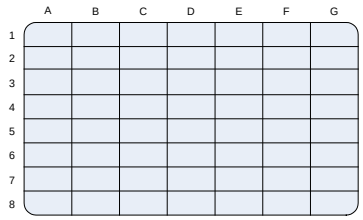


Figure 91 – LED Configuration

Standard backlight mode

When the standard backlight mode is enabled, all LEDs are constantly illuminated and the display can generate up to 250 cd/m^2. **By default, the VIEWPixx powers up in standard backlight mode** while the VIEWPixx /EEG and VIEWPixx /3D both power up in **scanning backlight mode**.

Scanning backlight mode

If your software enables the scanning backlight, the VIEWPixx will have display timing characteristics that resemble those of a CRT. In this mode, the backlight LEDs are powered only within a narrow horizontal bar, and the bar is scanned from the top of the display to the bottom. The scanning is synchronized to the LCD pixel updates, effectively hiding much of the liquid crystal rise/fall time. The scanning backlight can be used to improve the display characteristics of dynamic visual stimuli. In this mode, the display can generate up to 100 cd/m^2.

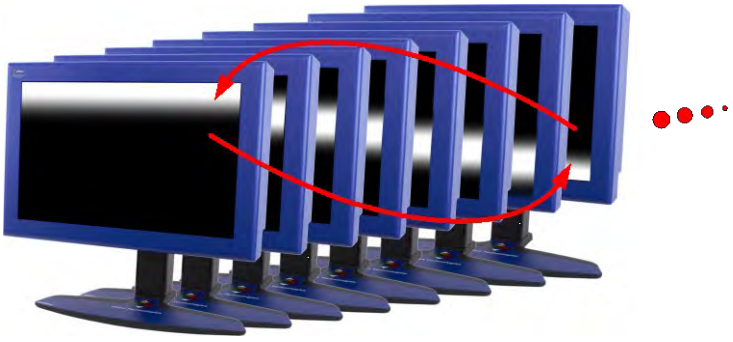


Figure 92 – Scanning backlight mode (visual example)

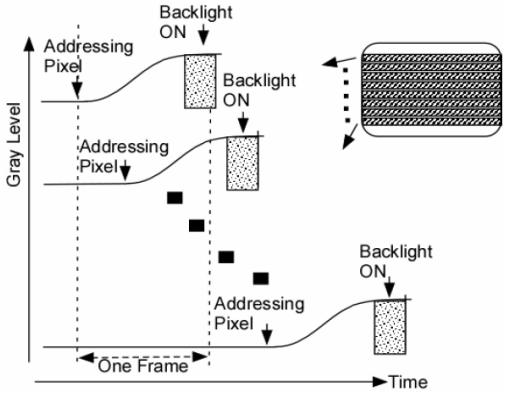


Figure 93 – Scanning backlight mode (Time graph)

Note that the following demos can be found by logging in to your MyVPixx account and navigating to https://vpixx.com/myvpixx/downloads-and-updates/

Configuring your VIEWPixx using PyPixx

PyPixx is a Python application allowing a user to configure any VPixx Technologies device. It can also provide information on the devices, perform calibrations and calibration tests and run demos. PyPixx operates on Windows, Mac and Linux operating systems.

This section briefly presents the PyPixx operating environment. It introduces the environment's menus, tabs, toolbars and status bars. Figure 35 below shows PyPixx's main window when the application is first opened (for purposes of this example, we connected one VPixx device: a VIEWPixx).

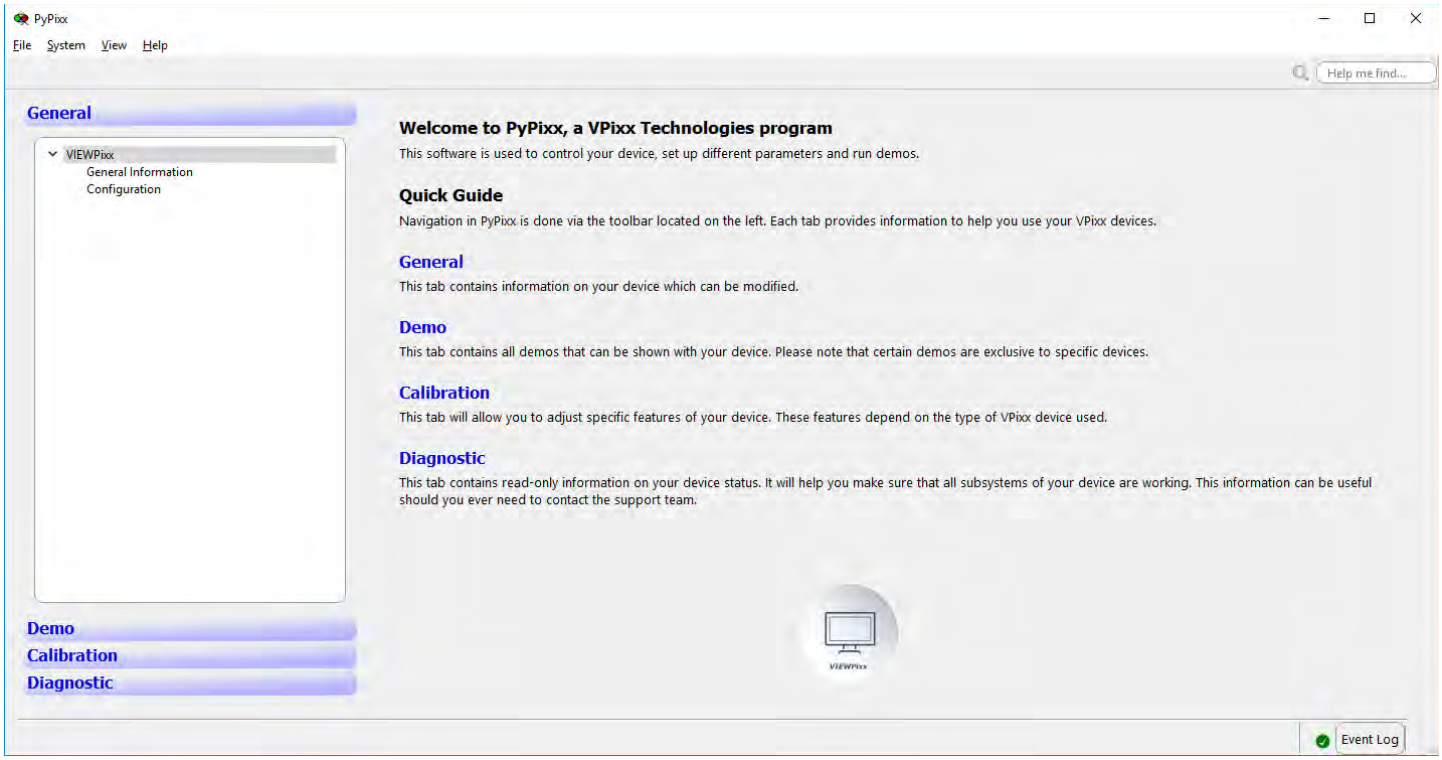


Figure 94 – PyPixx main window (connected VIEWPixx)

The main window's left portion is divided into four tabs: *General, Demo, Calibration* and *Diagnostic*. Each tab displays the powered VPixx devices currently connected to the computer.

- The *General* tab contains VPixx device information and allows you to configure certain device features.
- The *Demo* tab contains demos relevant to your device(s). Please note that certain demos are exclusive to specific devices.
- The *Calibration* tab allows you to recalibrate analog sub-systems in your device(s). These features depend on the type of VPixx device(s) used.
- The *Diagnostic* tab contains read-only information pertaining to your device status. It allows you to verify that all of your device subsystems function properly. This information can be useful if you contact the support team.

Central window

The central window provides information specific to the active device and the actions that can be performed on it. The figures below show PyPixx's Central window when only one device is detected (in this case, a VIEWPixx).

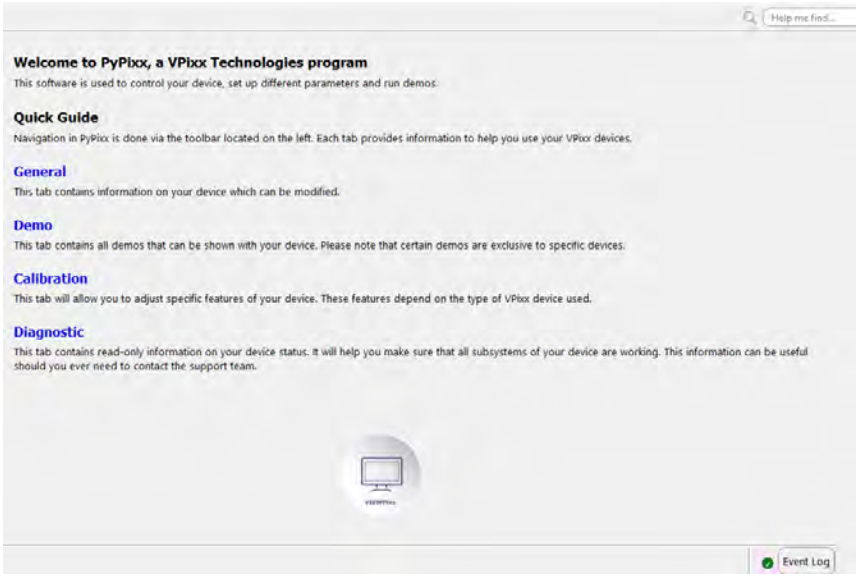


Figure 95 – Central Window (detected VIEWPixx)

Note that depending on which device you are using and which tab you clicked on, the central window will display a different message.

Status bar

The status bar at the bottom of the main window provides additional information to the user as to the functioning of PyPixx and the devices currently detected by the application. The status bar will also warn the user when the device is busy processing instructions.

Figure 96 – PyPixx status bar

Should an error occur on a VPixx device or in PyPixx, information related to this error may be obtained by clicking on the *Event Log* button.

General tab

The General tab contains setup information for your VIEWPixx device.

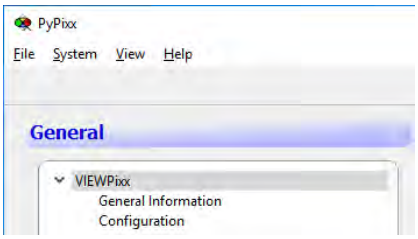


Figure 97 – General tab – VIEWPixx device

General Information

Clicking on **General Information** will open the VIEWPixx device's *General Information* window. In this window you will be able to see information concerning your VIEWPixx device such as its device ID, revision, firmware version, etc.

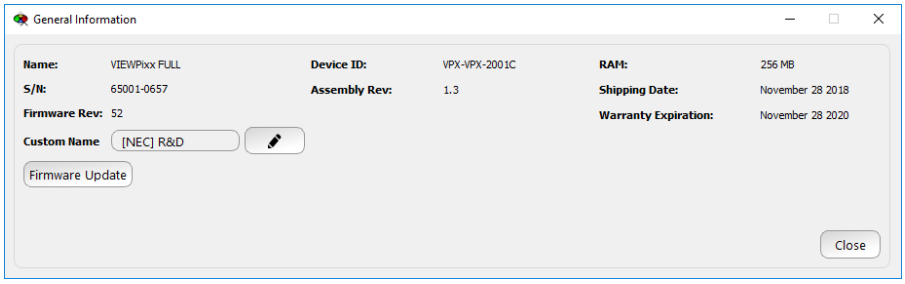


Figure 98 – General Information Window – VIEWPixx device

Clicking on the **Firmware Update** button will open the *Firmware Update* window allowing you to download the latest firmware update for your VIEWPixx.

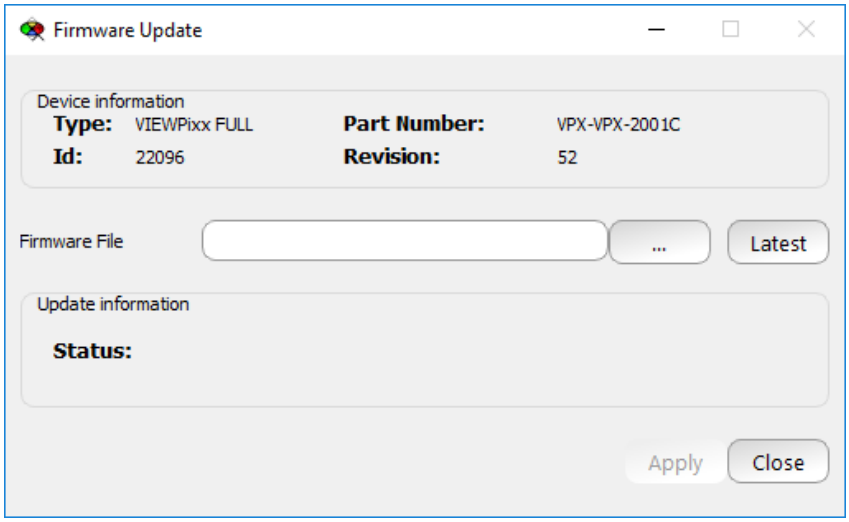


Figure 99 – Firmware Update Window – VIEWPixx device

From this window you can download the latest firmware file by clicking on the **Latest** button or choose a specific firmware file by clicking on the "…" button and selecting the desired file. Once this is done, click on the **Apply** button to save your changes.

Configuration

Clicking on **Configuration** will open the *VIEWPixx /3D Configuration* window, which applies to both VIEWPixx and VIEWPixx /3D.

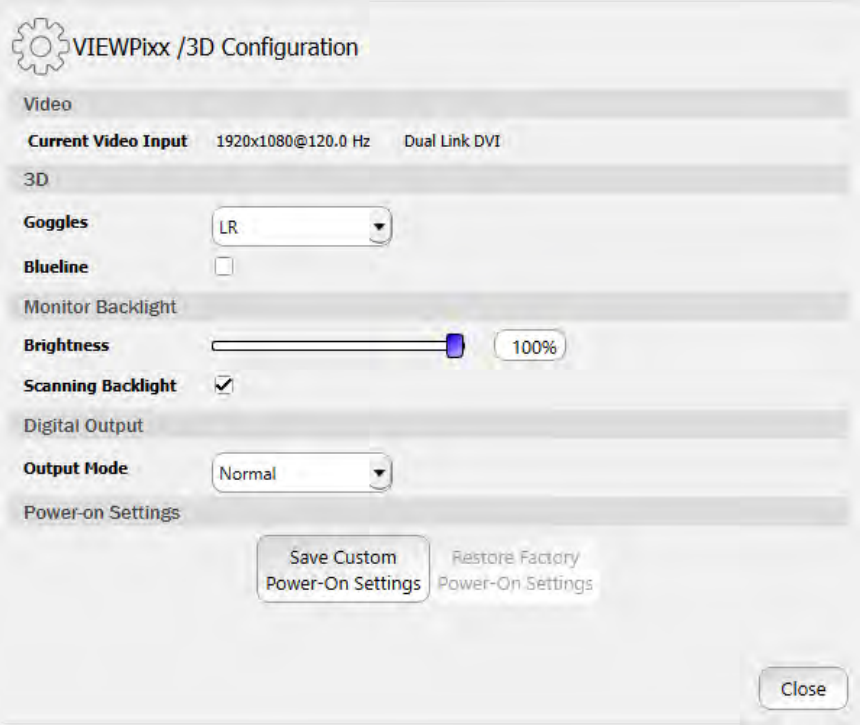


Figure 100 – VIEWPixx Configuration Window

You can select the *3D* configuration (either Blueline or specific goggle brands) according to your experimental needs.

The *Monitor Backlight* allows you to adjust your screen brightness from 0-100% and activate or deactivate *Scanning Backlight*. Enabling *Scanning Backlight* means the VIEWPixx or VIEWPixx /3D will have display timing characteristics that resemble those of a CRT.

Scanning backlight is ALWAYS enabled for VIEWPixx /EEG. It is also the default mode for the VIEWPixx /3D but may be deactivated.

In Scanning Backlight mode, the backlight LEDs are powered only within a narrow horizontal bar, and the bar is scanned from the top of the display to the bottom. The scanning is synchronized to the LCD pixel updates, effectively hiding much of the liquid crystal rise/fall time. The scanning backlight can be used to improve the display characteristics of dynamic visual stimuli. In this mode, the display can generate up to 100 cd/m^2.

use of the scanning backlight can drastically reduce pixel response time, which in turn eliminates the blurring effect of moving stimuli and reduces "ghosting" with 3D stimuli.

Figure 101 demonstrates pixel response times for a 120 Hz alternating black/white signal. This is an example of a typical consumer LCD. It shows a pixel rise time of 5 ms and a fall time of 1.25 ms.

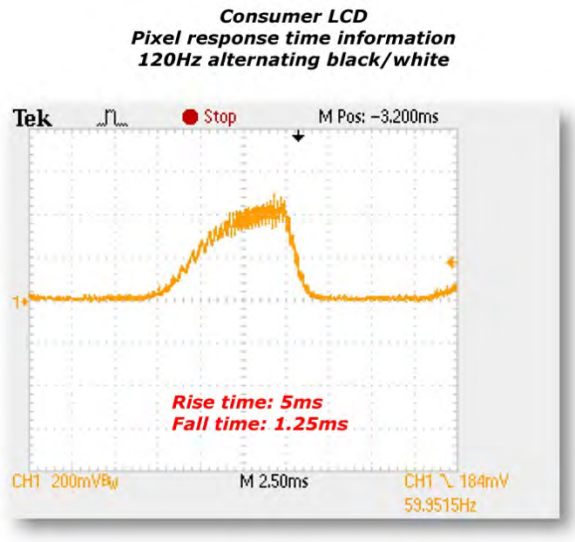


Figure 101 – Typical consumer LCD pixel response time

Conversely, for the same signal, use of a scanning backlight shows a pixel rise time of 1 ms and a fall time of 1 ms, as shown on Figure 102 below.

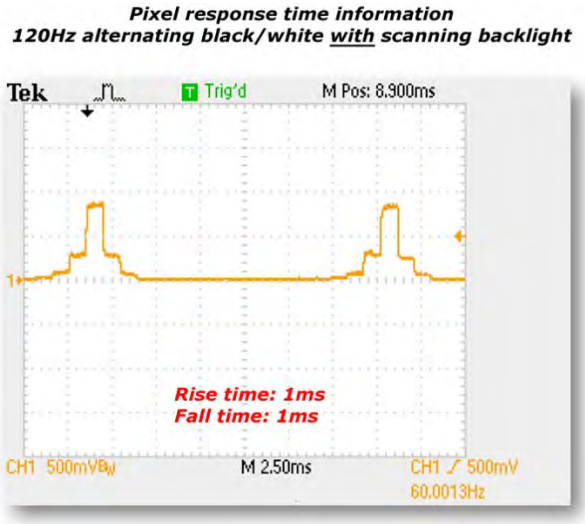


Figure 102 – VIEWPixx pixel response time (with scanning backlight)

Finally, use of a scanning backlight for a 120 Hz constant white signal shows a pixel rise time of 1 ms and a fall time of 1 ms, as shown on Figure 103 below.

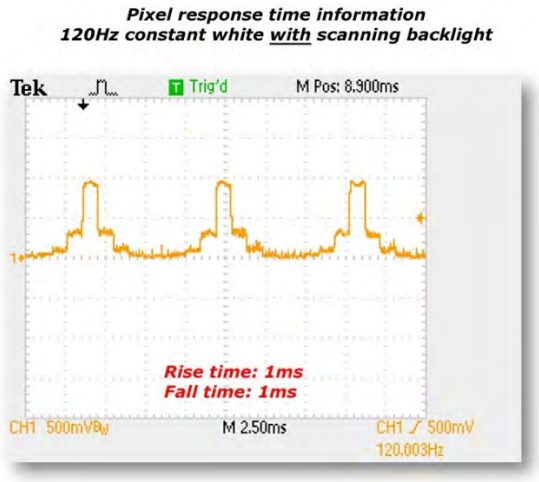


Figure 103 – VIEWPixx pixel response time (constant white signal)

The *Output Mode* allows you to select which TTL digital output mode (either *normal* or *Pixel Mode*) you wish to use. The VIEWPixx contains 24 TTL inputs and 24 TTL outputs. The Inputs are pulled up, so simple response boxes can be read with no additional hardware. TTL input transitions are timetagged enabling response-time calculation with microsecond precision. Similarly, TTL outputs can be scheduled to transition at a specified point within a video frame, enabling simple triggering of external testing hardware. *Normal Mode* is used when the digital output interface is controlled by the low-level ANSI C API, MATLAB/Octave and Python libraries through the USB port, while *Pixel Mode* can be enabled if you use software like E-Prime or Presentation to control the interface.

When *Pixel Mode* is enabled for the 24 TTL outputs, the levels of the 24 TTL outputs are controlled by the 8-bit RGB components of the display's top-left pixel. This strategy is simple to program using any stimulus generation software, and guarantees perfect synchronization between video and external hardware. The picture on the following page shows the delay between digital output and stabilization of the first top-left pixel with the scanning backlight. This delay is constant and is always 6 ms after the digital output rising edge.

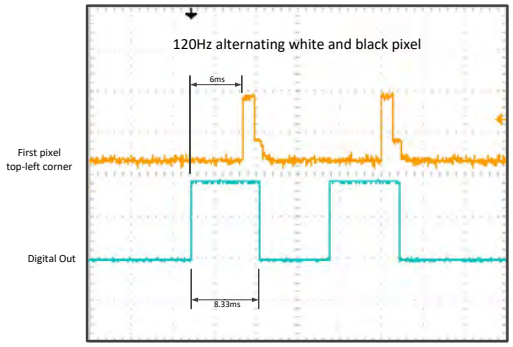


Figure 104 – Pixel Mode

The **Save Custom Power-On Settings** button allows you to save the current VIEWPixx configuration. This configuration will be used the next time the device is powered on. It is always possible to return to the factory settings at a later date by clicking on the **Restore Factory Power-On Settings** button.

Demo tab

PyPixx contains a multitude of VIEWPixx demos that may be used to quickly preview the features of your VIEWPixx device. To access the demos, click on the *Demo* tab. This will display the PyPixx demo window for your VIEWPixx. Simply select the demo you wish to run and follow the on-screen instructions.

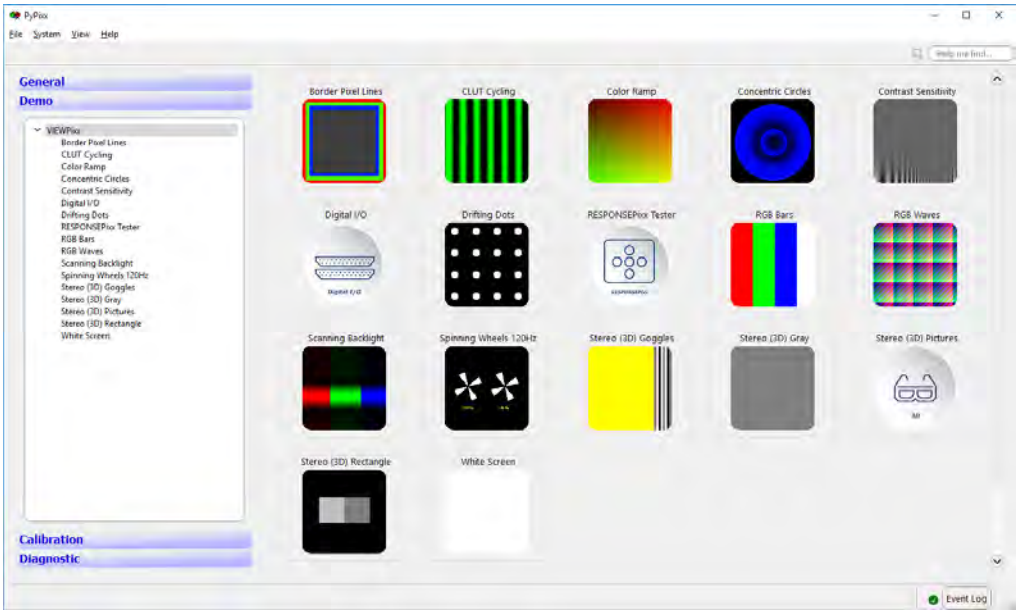


Figure 105 – Demo tab – VIEWPixx device

Calibration tab

The Calibration tab contains the calibration options that are available for your currently connected VPixx products. In the case of the VIEWPixx, the available calibration procedure is *White Calibration* and it requires either an I1Pro or I1Display from X-Rite (refer to the

Calibrating your Stimulus Presentation Equipment chapter for more information).

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

Simply navigate to the *Calibration* tab and select White Calibration:

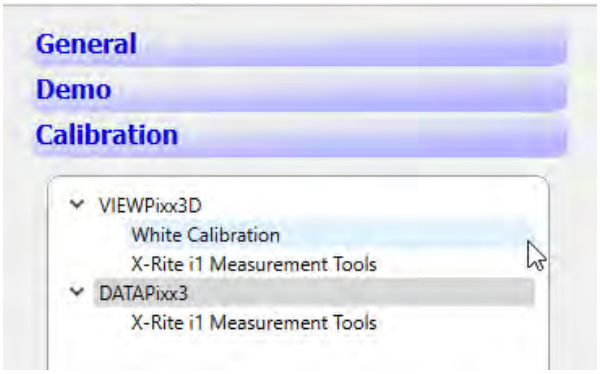


Figure 106 White Calibration

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 so 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.

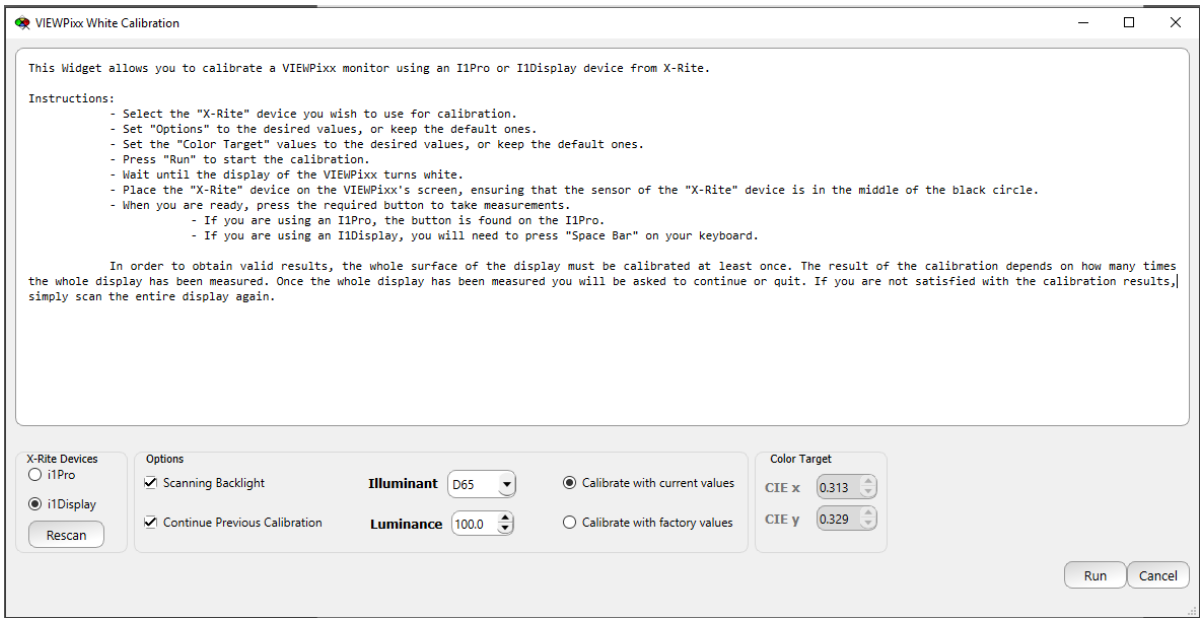


Figure 107 White Calibration

If no I1Pro or I1Display is detected, the window will display the *No X-Rite devices detected. Please make sure that the device is connected and press update* message.

Diagnostic tab

The Diagnostic tab allows you to verify that all of your currently connected VPixx products are functioning correctly. If they are not, the information gathered here will be useful when you contact the VPixx support team.

In the case of the VIEWPixx, the available diagnostic tools are explained in the figure below.

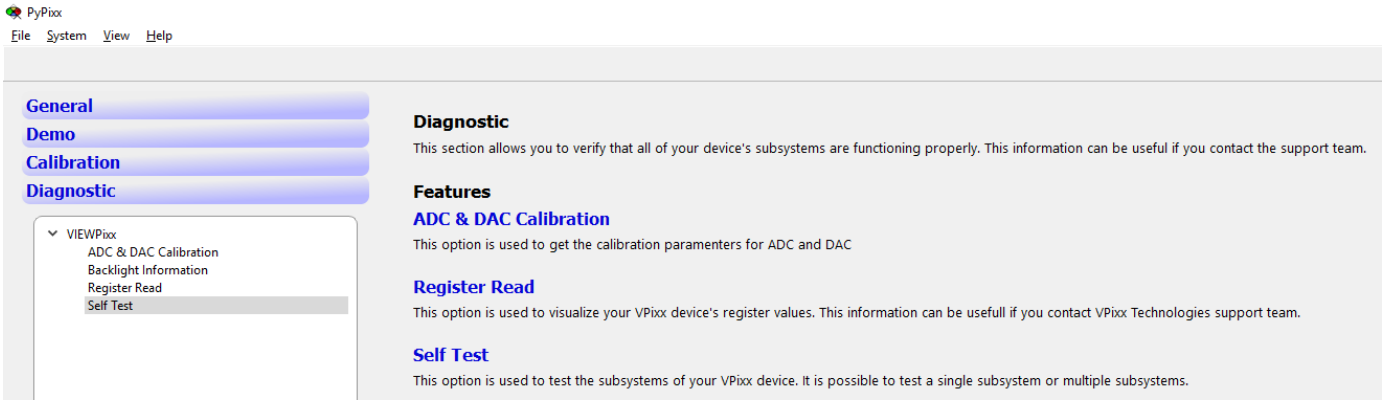


Figure 108 – Diagnostic tab – VIEWPixx device

The *ADC&DAC Calibration*, *Backlight Information* and *Register Read* tests allow you to update the specific diagnostic information and send it to the VPixx support team if required.

Clicking on *Self Test* allows you to select exactly which self-diagnostic test(s) you wish to run for your VIEWPixx. If you wish to run a complete test, activate all the checkboxes and click on the **Run** button. The green progress bar from the Selftest window displays the test's progress and when completed, the window will display the results as shown below. These may be relayed to the VPixx support team if requested.

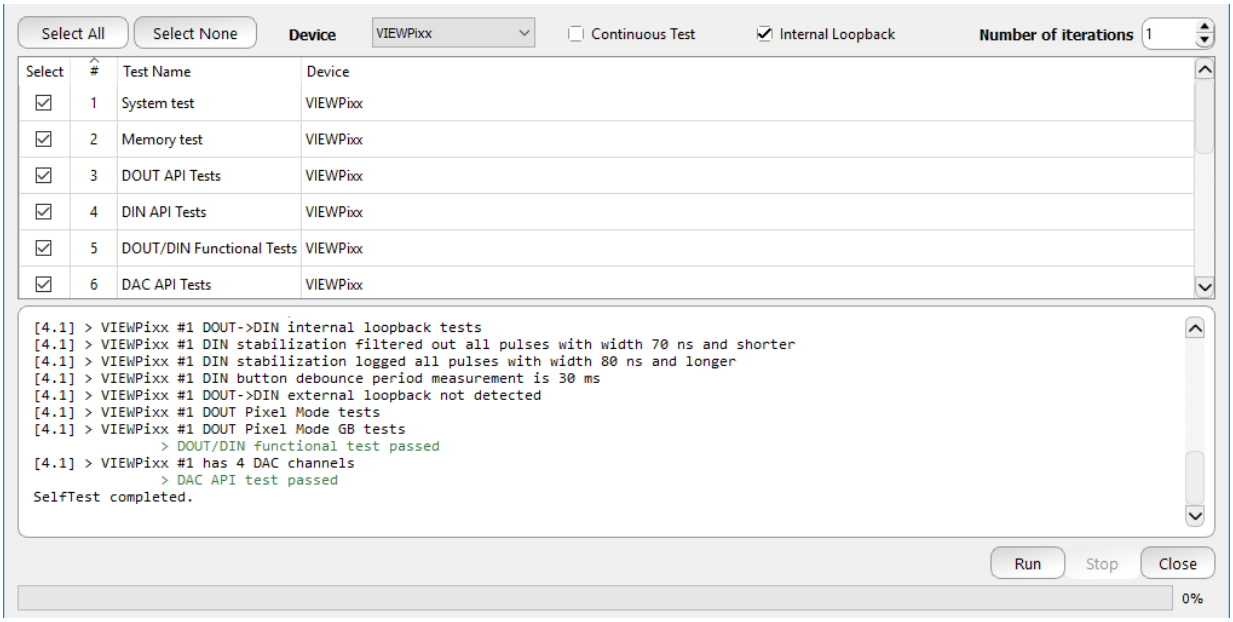


Figure 109 – VIEWPixx Self Test

Calibrating your Stimulus Presentation Equipment

The PROPixx and VIEWPixx products require periodic calibration to ensure their performance is optimal. This chapter describes the tools and procedures VPixx Technologies recommends to calibrate your VPixx stimulus presentation equipment.

Calibration procedure for VIEWPixx series using X-Rite i1Pro

The X-Rite i1Pro is a true spectrophotometer, featuring greater accuracy than tri-stimulus photometers which are based on color filters. A new innovative Swiss-engineered design makes the i1Pro much more affordable than previous generations of spectrophotometers.

The i1Pro is a contact spectrophotometer and includes hardware for holding the sensor flat against the region of the display being calibrated. The i1Pro connects to your computer through a USB interface, allowing your own software to initiate automated measurements and acquire data. The i1Pro can return CIE Lxy data for photometry measurements and can also return raw spectral data between 380 nm and 730 nm at 10 nm intervals for radiometry applications. The i1Pro includes a calibration base allowing the photometer to be automatically recalibrated before each use.

Figure 110 – X-Rite i1Pro spectrophotometer

Display calibration also allows you to change the white point and maximum luminance of the VIEWPixx series. The vputil software tool makes use of the X-Rite i1Pro spectrophotometer to run a simple calibration procedure using 56 sample points across the display.

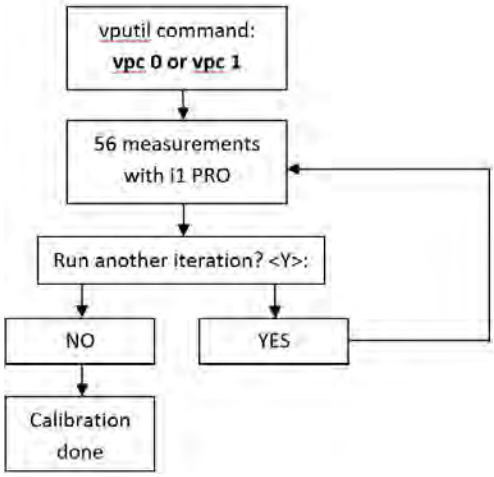
For best calibration results, please note the following points:

1- Make sure the room temperature is stable during calibration. For best accuracy, calibrate the VIEWPixx in the same location where it will be used for testing.
2- Mount the VIEWPixx onto its stand when calibrating or testing. Placing the VIEWPixx bezel directly on a table will impede cooling airflow and impact calibration accuracy.
3- VIEWPixx calibration should be done in a dark room. Other light sources (such as the host computer display) should not be allowed to reflect off of the VIEWPixx display during calibration.
4- For best measurement accuracy, keep the X-Rite i1Pro as stable and flat on the VIEWPixx screen as possible when taking measurements.
5- The VIEWPixx should be powered up a minimum of 30 minutes before the start of the calibration or the start of testing.

	For more information on using the X-Rite i1 Pro*, please refer to the* ***X-Rite i1 Pro Spectrophotometer user manual****.*

Calibration block diagram

vputil command definition : **vpc 0**: Backlight in standard mode. **vpc 1**: Scanning backlight mode



Calibration steps

1- Turn the host computer on and plug the i1Pro into a USB port. Place the i1Pro onto the calibration tile included in the kit. Make sure all drivers are installed and the i1Pro is detected and ready for use.
2- Connect the VIEWPixx to a USB port on the host computer and connect the video OUT of the host to the Video IN port of the VIEWPixx.
3- Power on the VIEWPixx.
4- Start the vputil application by clicking on its desktop shortcut. The vputil menu will appear:

Figure 111 – vputil menu

5- Type "3" to initiate the Calibration commands, then type the command vpc 0 (for standard backlight mode) or vpc 1 (for scanning backlight mode). The VIEWPixx internally maintains separate calibration tables for the scanning and standard (non-scanning) backlight modes. The scanning/standard backlight modes default to a target luminance of 100/250 candelas per square meter, respectively. White point defaults to D65 (CIE x,y = 0.313,0.329). **If desired, the user can explicitly set the luminance and white point calibration to target different values**. The quick command: vpc 0 is equivalent to the detailed command: vpc 0 250 0.313 0.329 where vpc = VIEWPixx calibration, 0 = standard backlight mode, 250 = luminance in cd/m^2, and 0.313 0.329 are the CIE chromaticity coordinates. Similarly, the quick command: vpc 1 is equivalent to the detailed command: vpc 1 100 0.313 0.329 where 1 = scanning backlight mode, and 100 = luminance in cd/m^2.

6- Before calibration begins, the user may be asked to put the i1Pro on the calibration tile provided in the kit and hit <<ENTER>>.

	Note that the calibration tile has a serial number, which must match the serial number of the i1Pro spectrophotometer.

Figure 112 – Calibration tile

7- The spectrophotometer measurements now begin. A square with an inscribed circle will appear on a white background. Place the rounded measurement head of the i1Pro onto the circle and push the side button of the

i1Pro to take a measurement. Maintain contact between the i1Pro and the VIEWPixx glass, **applying the least pressure possible** (just enough to ensure that the i1Pro is flat on the screen surface). The square and circle are redrawn with double lines while the i1Pro takes its luminance/chromaticity reading. It is important not to move the i1Pro during a reading, since this can result in incorrect data and therefore increase the number of calibration iterations required to attain the desired uniformity. After each measurement, the square and circle target will move to the next measurement spot. Try to lift and move the i1Pro slowly to avoid scratching the LCD glass. After taking 56 measurement points, you will see the calibration results (an example is shown below) on the host computer:

```
Calibrating 'SingleEmission' ...done.
    Iteration 1
    (49,43,43)(43,35,36)(43,33,34)(40,32,33)(42,33,33)(42,34,35)(48,40,40)
    (34,29,31)(27,23,24)(26,21,21)(28,22,24)(28,23,25)(27,24,25)(35,29,30)
    (32,27,30)(27,23,26)(27,22,23)(30,25,28)(29,23,24)(28,23,25)(32,28,30)
    (31,28,29)(27,22,25)(27,22,24)(27,22,24)(27,22,25)(29,24,25)(33,28,32)
    (31,26,29)(26,21,24)(26,22,24)(27,22,23)(27,22,24)(28,24,25)(30,26,29)
    (31,28,30)(26,24,25)(27,23,25)(30,26,29)(26,23,23)(27,23,27)(31,28,31)
    (34,30,31)(27,22,24)(26,22,24)(26,23,24)(27,22,24)(26,23,25)(33,29,31)
    (42,36,39)(34,30,32)(39,33,36)(33,29,31)(35,29,31)(38,32,35)(42,37,42)

    (286.3,0.297,0.300)(278.5,0.292,0.298)(264.4,0.291,0.299)(259.3,0.290,0.298)(262.5,0.290,0.300)(266.3,0.289,0.300)(277.1,0.293,0.301)
    (276.1,0.295,0.303)(263.3,0.291,0.303)(252.2,0.288,0.302)(248.9,0.286,0.300)(253.5,0.285,0.302)(254.8,0.286,0.304)(264.4,0.290,0.308)
    (275.5,0.295,0.3                                                                                                  )(255.2,0.283,0.308)(257.3,0.285,0.309)(264.5,0.286,0.310)
    (266.2,0.294,0.3                                                                                                  )(245.5,0.288,0.307)(247.7,0
    (261.7,0.296,0.3                                                                                                  )(244.1,0.288,0.310)(245.8,0
    (263.4,0.295,0.3                                                                                                  )(246.1,0.293,0.311)(248.3,0
    (277.3,0.298,0.310)(267.1,0.298,0.311)(260.3,0.295,0.310)(256.4,0.295,0.313)(254.4,0.294,0.312)(259.5,0
    (283.3,0.297,0.305)(278.6,0.296,0.305)(277.1,0.295,0.304)(270.0,0.294,0.307)(269.3,0.293,0.308)(270.2,0

    L: range=243.3-286.3, uniformity(min/max)= 85.0%
    x: range=0.283-0.298
    y: range=0.298-0.313
```

Backlight LED current settings saved in VIEWPixx

Quick view of Lxy range.

Lxy CIE coordinates measured for 56 sample points

8- When the *Run another iteration? <Y>:* question appears, type <Y + ENTER> to continue with another calibration iteration, or <N + ENTER> if the results satisfy the calibration requirements. Each calibration iteration will improve the luminance/chromaticity uniformity, with a luminance uniformity exceeding 98% generally being attained within 5 iterations. Here is an example showing 98.5% uniformity after 6 calibration iterations targeting D65 at 250 cd/m^2:

```
    Iteration 6
    (33,34,29)(30,26,20)(38,30,25)(35,31,24)(35,29,24)(37,30,23)(31,30,25)
    (24,23,19)(26,23,20)(27,22,18)(29,24,19)(26,21,18)(28,25,19)(28,23,23)
    (24,21,24)(25,18,21)(24,18,17)(34,30,24)(29,15,17)(27,22,21)(27,20,22)
    (25,21,20)(24,22,21)(28,23,19)(29,24,21)(26,22,21)(28,25,19)(27,22,20)
    (25,22,23)(26,22,19)(27,24,20)(28,21,21)(29,21,21)(27,25,22)(28,24,23)
    (28,23,22)(25,26,23)(26,20,21)(32,32,27)(25,21,21)(29,25,22)(25,23,24)
    (26,22,22)(24,21,21)(24,21,18)(27,22,22)(28,23,22)(23,21,20)(28,24,23)
    (33,29,27)(29,26,23)(29,25,22)(29,26,24)(31,24,25)(33,29,23)(33,30,26)

    (249.4,0.312,0.328)(250.3,0.313,0.329)(250.9,0.313,0.329)(251.5,0.313,0.329)(251.9,0.314,0.329)(251.7,0.314,0.329)(250.9,0.313,0.329)
    (250.9,0.314,0.329)(252.1,0.313,0.329)(250.8,0.313,0.329)(250.6,0.313,0.329)(250.7,0.313,0.329)(250.5,0.314,0.330)(250.9,0.314,0.330)
    (250.5,0.313,0.330)(250.4,0.313,0.329)(250.8,0.313,0.329)(252.5,0.313,0.329)(251.9,0.313,0.329)(250.8,0.313,0.329)(250.6,0.313,0.329)
    (250.1,0.314,0.329)(249.4,0.314,0.329)(250.7,0.313,0.330)(251.2,0.313,0.329)(251.5,0.313,0.330)(250.7,0.313,0.329)(250.6,0.313,0.328)
    (249.9,0.315,0.330)(248.9,0.314,0.330)(251.7,0.314,0.330)(251.3,0.313,0.329)(251.7,0.313,0.329)(252.0,0.313,0.329)(250.4,0.313,0.328)
    (250.3,0.313,0.329)(250.0,0.313,0.329)(249.4,0.314,0.329)(250.0,0.312,0.329)(250.7,0.313,0.329)(251.1,0.313,0.329)(249.8,0.313,0.329)
    (249.7,0.313,0.329)(250.2,0.313,0.329)(250.4,0.313,0.330)(250.3,0.313,0.330)(251.5,0.313,0.330)(251.2,0.313,0.329)(251.2,0.313,0.329)
    (252.2,0.313,0.329)(249.7,0.313,0.329)(248.8,0.313,0.329)(249.1,0.313,0.329)(249.3,0.313,0.329)(251.9,0.312,0.329)(252.0,0.313,0.329)
```

250 – 1% = 247.5 cd/m^2
250 + 1% = 252.5 cd/m^2

```
L: range=248.8-252.5, uniformity(min/max)= 98.5%
x: range=0.312-0.315
y: range=0.328-0.330
Run another iteration? <Y>: n
```

When calibration results are satisfactory, type <N + ENTER> for the *Run another iteration? <Y>:* question and the calibration values will be stored into VIEWPixx memory. Every time the VIEWPixx is turned on, the new calibrated LED currents will be loaded, resulting in the same luminance/chromaticity uniformity as seen during calibration. VIEWPixx calibration should be run every time the operating environment (eg: display placement/airflow, temperature) changes appreciably.

Calibration procedure for PROPixx using X-Rite i1Display Pro

The i1Display Pro colorimeter features an advanced, high-end optical system with custom-designed filters. It provides a near perfect match to the color perception of the human visual system, delivering superior color measurement results. The i1Display Pro supports all modern display technologies, including LED backlight and wide gamut displays. It is spectrally calibrated, making it fully field upgradeable to support future display technologies.

Figure 113 – X-Rite i1Display Pro colorimeter

Calibration steps

1- Turn the host computer on and plug the i1Display Pro into a USB port.

	*For more information on using the* X-Rite i1Display Pro*, please refer to the* ***X-Rite i1Display Pro user manual****.*

2- Place the i1Display Pro as described in the ***i1Display Pro user manual*** (*Projector measurement* section).
3- Connect the PROPixx to the host computer.
4- Power on the PROPixx and the controller.
5- Start the vputil application by clicking on its desktop shortcut. The vputil menu will appear.

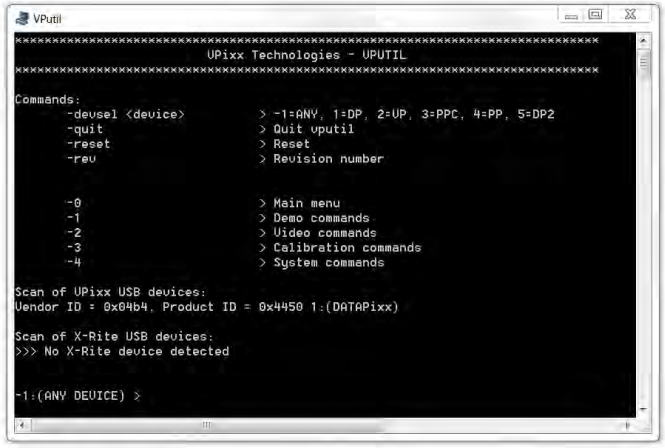


Figure 114 – vputil menu

6- Type “3” to initiate the Calibration commands, then type the command ppc.
The ppc command is set for a white point calibration to D65 (CIE x,y = 0.313,0.329)

7- The automated calibration now begins. It takes around 30 minutes to calibrate the PROPixx projector.

I1Pro and I1Display Widget (PyPixx)

The i1Pro and i1Display widget can 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 PyPixx’ *System* menu:

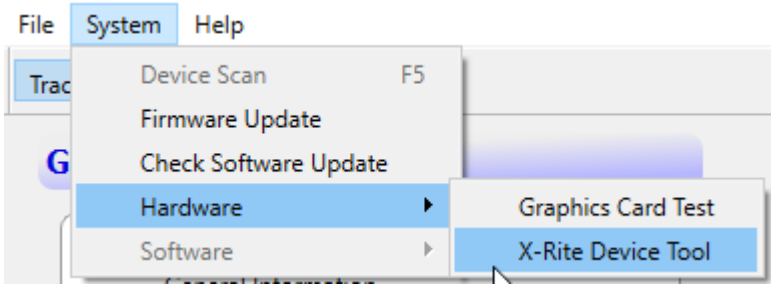


Figure 115 - X-Rite Device Tool

Depending on the X-Rite device, yfou can adjust relevant settings as well as visualize and store recorded data (as a spectrum graph or as a position in a CIExyY graph):

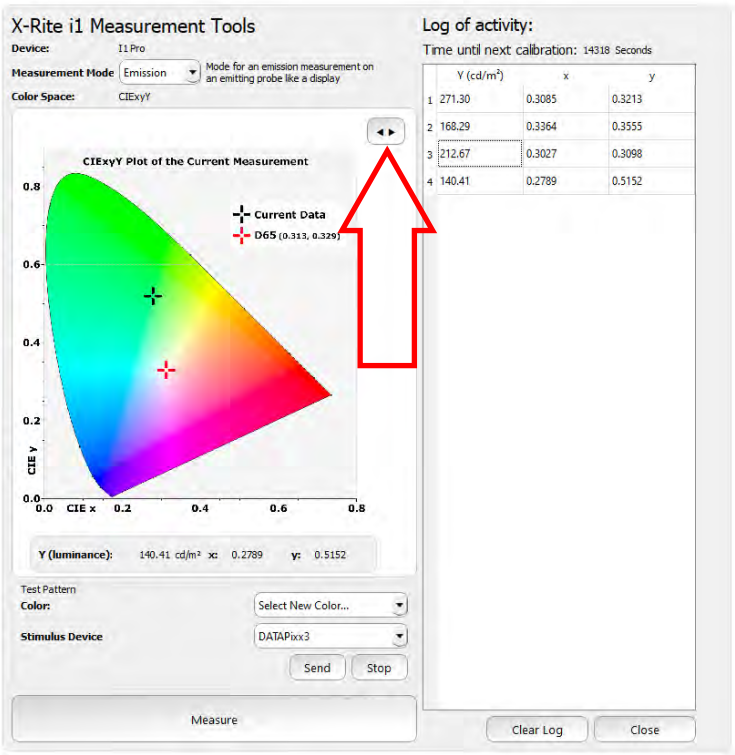


Figure 116 - default UI for i1 Toolset used with an i1-Pro device and button to toggle between pectrum and chromaticity charts

The ‘copy’ feature allows users to easily copy the spectrum data for use in their preferred analysis program:

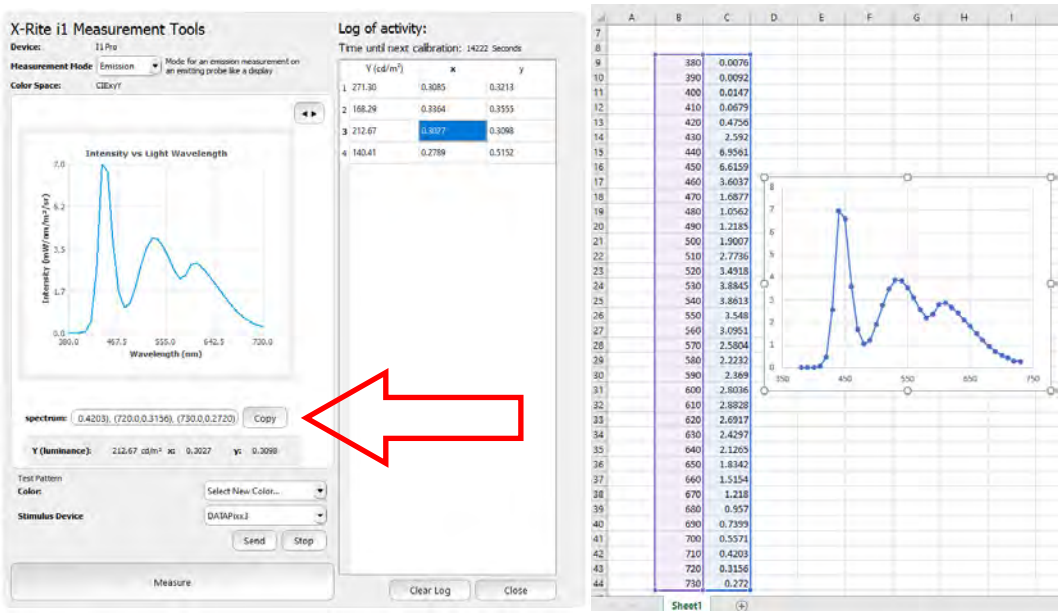


Figure 117 - Left: UI when looking at the spectrum and featuring the 'copy' button. Right: copied data pasted and plotted as-is by excel

You can also have a specific color displayed by any of your VPixx devices when taking readings with your x-rite device.

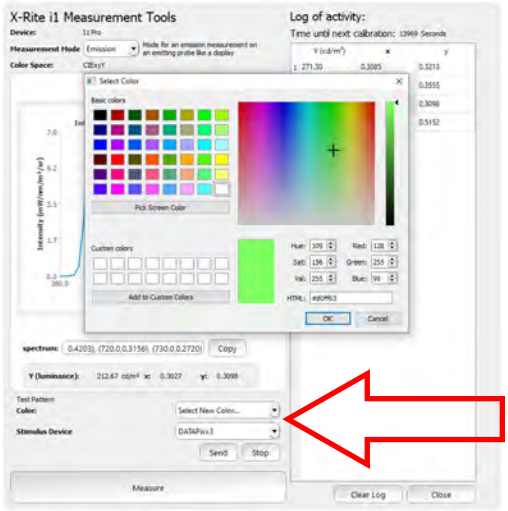


Figure 118 - color selection pop up featuring the combo boxes used to access it and to select which device to send that color to.

Transporting & storing your equipment

Following initial equipment unpacking and setup, your system should be moved only when necessary, such as when your experiment requires a different equipment setup. When not in use, cover the system with a soft protective sheet in order to protect the equipment from dust accumulation.

Stimulus Presentation Experiments

This chapter gives you the necessary information on how to rapidly get up and running with your VPixx Stimulus Presentation product(s) such as the VIEWPixx and PROPixx.

It is divided according to your specific product and research needs.

Before setting up your first stimulus presentation session, CAREFULLY READ the safety information contained in the Warnings and Safety Information section AND the information contained in the Installation Guide(s) relating to your product(s).

High Bit Depth

Color look-up tables (CLUTs) are a part of the video pipeline. You can have CLUTs for color correction, gamma correction, color removal, etc. A CLUT is usually a 256 by 3 matrix, where each column represents a color channel (red, green, blue) and each row represents an operation or a new color value for that color. A non-modifying CLUT is defined as follows in MATLAB: `linspace(0, 1, 256)' * [1, 1, 1]`. Color components in MATLAB are usually represented as a double going from 0 (zero intensity) to 1 (full intensity). Colors usually have an 8-bit representation, and therefore can vary from 0 to 1 in 256 steps.

In this demo, we use CLUTs to define a transparency color and to have different colors on the console display and the main display.

The transparency color is useful if you want to display a message on the console display but have it invisible to the participant. Different colors on the two displays can be useful if you want different shades or a different gamma on the two displays.

As stated before, colors are usually represented using 8 bits of information. Our devices are capable of displaying 10-12 bits of colors, however a video signal can only send 24 bits (8 per color) of information. To overcome this limitation, we have to manipulate the data and interpret it differently in our device.

In mode M16, we calculate a 16-bit shade of gray and send that information using the 8 bits of red and 8 bits of green. The VPixx device combines it to display a 10-12 bit closest value if the blue component is zero. If the blue component of the pixel is not zero, then it is used as an index to a CLUT to display the pixel as an overlay. That overlay will have the color defined in the CLUT.

There are two look-up tables on our device, one for the main device and one for the console display. This means you can have different color overlays on the console and on the main device.

It is also possible to define a transparency color to the device. If the CLUT table has that color at the given blue index, there will not be an overlay, the pixel will be drawn as if the blue value had been zero (using the gray color provided by the combined Red and Green).

Note that the following demos can be found by logging in to your MyVPixx account and navigating to https://vpixx.com/myvpixx/downloads-and-updates/

3D

The blue line is a single line at the bottom of the display which our device uses to determine if the current frame is a left-eye or right-eye frame and to synchronize our 3D goggles. On a PROPixx projector, there is a polarizing filter for passive 3D and on the VIEWPixx and VIEWPixx /3D, there is an infra-red emitter to synchronize the active 3D glasses.

Gaze Contingent Display

VPixx devices are very effective for gaze contingent paradigms. For example, the analog inputs can be used to acquire eye-tracking data for faster modification of the current stimulus. In the PROPixx , it is possible to transmit eye-tracking information directly and use it to control the display. The VIEWPixx /3D's fast response time allows for a very good stimulus display for gaze contingent applications.

Note that the following demos can be found by logging in to your MyVPixx account and navigating to https://vpixx.com/myvpixx/downloads-and-updates/

I/O Monitoring and Generation

Digital OUT (Triggers)

Most VPixx products have 24 digital outputs (Dout) that can be used to synchronize different lab equipment with your video signal.

You must first create a buffer for the desired triggers. It can be simple [1 0] or a complicated waveform [1 0 1 1 0 1]. This information must be filled in for all the channels that will output.

To determine the values that must be sent, think of the channels as binary states. This example gives 0b0000 0000 0100 0111. We convert this to decimal, 2^0+2^1+2^2+2^6 = 71, therefore our trigger is set to value [71 0 0 0].

One thing to keep in mind is that the physical pin number is not equal to the DOut channel number.

Pin	Description	Pin	Description
1	Digital Out 0	14	Digital Out 1
2	Digital Out 2	15	Digital Out 3
3	Digital Out 4	16	Digital Out 5
4	Digital Out 6	17	Digital Out 7
5	Digital Out 8	18	Digital Out 9
6	Digital Out 10	19	Digital Out 11
7	Digital Out 12	20	Digital Out 13
8	Digital Out 14	21	Digital Out 15
9	Digital Out 16	22	Digital Out 17
10	Digital Out 18	23	Digital Out 19
11	Digital Out 20	24	Digital Out 21
12	Digital Out 22	25	Digital Out 23
13	GND	Shield *	

Figure 119 – Digital Out (trigger) pin descriptions

Digital outputs can also be set using **Pixel mode**. In this mode, the 24 digital outputs will use the 24-bit RGB value of the top left pixel sent to the PROPixx controller. Refer to the **Pixel Mode Theory** section for more information on pixel mode.

Desired D_{out} to trigger	Value to add	Color and number	Desired physical pin to trigger
0	1	Red 0	Pin 1
1	2	Red 1	Pin 14
2	4	Red 2	Pin 2
3	8	Red 3	Pin 15
4	16	Red 4	Pin 3
5	32	Red 5	Pin 16
6	64	Red 6	Pin 4
7	128	Red 7	Pin 17
8	1	Green 0	Pin 5
9	2	Green 1	Pin 18
10	4	Green 2	Pin 6
11	8	Green 3	Pin 19
12	16	Green 4	Pin 7
13	32	Green 5	Pin 20
14	64	Green 6	Pin 8
16	1	Blue 0	Pin 9
17	2	Blue 1	Pin 22
18	4	Blue 2	Pin 10
19	8	Blue 3	Pin 23
20	16	Blue 4	Pin 11
21	32	Blue 5	Pin 24
22	64	Blue 6	Pin 12
23	128	Blue 7	Pin 25

Figure 120 – Digital Out pin descriptions (pixel mode)

Pixel Mode Theory

A video signal contains all the pixels for every frame. The resolution determines the number of pixels: for example, 1920 x 1080 means 1920*1080 pixels on your screen. Each of these pixels contains the color information it will show on screen.

Colors are described using the RGB (Red-Green-Blue) format. For every video frame, each pixel has access to 24 bits of color information (8 bits per color). This means, for example, that red can take values from 0 to 255 and consequently that 256 different shades of red are possible. Red 0 signifies no red, while red 255 signifies 100% red color. All of the three colors (RGB) have 8 bits, which creates 16 777 216 different possible colors. RGB colors are usually represented as (R,G,B), where R, G and B are the respective color values [0-255].

A color is usually described with 8 bits of information. 8-bit information can be seen as $R_7R_6R_5R_4R_3R_2R_1R_0$, which is a binary representation of the color (in this example, R for Red). Every bit (R_i) can take the value 0 or 1, and the final value will be a sum:

$$Red = \sum_{i=0}^{i=7} 2^i * R_i$$

The color red 255 would be represented as 0b11111111, whereas red 16 would be 0b00010000. Since 255 represents every bit at a value of 1, it is the maximum value for an 8-bit number.

Analog output interface

All analog subsystems contain 4 DAC (Digital-to-Analog-Converter) channels, with 16-bit precision and ±10 V output swing. The maximum update rate is 1 MSPS and all 4 channels update simultaneously. The DAC update rate can be specified in samples per second, samples per video frame, or nanoseconds per sample. Waveform onset can be synchronized to video refresh with microsecond precision.

Analog outputs are NOT available if you have the LITE version of your VPixx product. If you require this functionality, please send an email to sales@vpixx.com and request a Lite to Full I/O system upgrade

Analog input interface

VPixx products include 16 ADC (Analog-to-Digital Converter) channels, with 16-bit precision and ±10 V input range. The maximum sampling rate is 200 kSPS and all 16 channels are simultaneously sampled for evoked potential and other applications sensitive to the sampling phase. The ADC sample rate can be specified in samples per second, samples per video frame, or nanoseconds per sample. Sampling can be synchronized to video refresh with microsecond precision.

Analog inputs are NOT available if you have the LITE version of your VPixx product. If you require this functionality, please send an email to sales@vpixx.com and request a Lite to Full I/O system upgrade

Digital interface

VPixx products include 24 TTL inputs and 24 TTL outputs. Inputs are pulled up, so simple response boxes can be read with no additional hardware. TTL input transitions are timetagged to enable response time calculation with microsecond precision. Similarly, TTL outputs can be scheduled to transition at a specified point within a video frame, to enable simple triggering of external testing hardware.

New digital output on digital input mode

The digital output on digital input mode allows you to send triggers whenever there is an input registered on a VPixx device using 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.

Audio interface

VPixx products include a stereo audio CODEC which can bias and convert a stereo microphone input. The CODEC can also drive a 96 kSPS audio stimulus directly into headphones. Stimulus phase between audio-left, audio-right, and video refresh can be controlled with microsecond precision for cross-modal research.

The audio interfaces are NOT available if you have the LITE version of your VPixx product. If you require this functionality, please send an email to sales@vpixx.com and request a Lite to Full I/O system upgrade

Digital IN (RESPONSEPixx)

The Digital input (Din) subsystem is used to record triggers, button presses, etc. A log (instead of a schedule) is used for the digital inputs so that only changes are recorded.

Time (ms)	Channels (0 to 5)
0	[0,0,0,0,0,0]
2.52	[1,0,0,0,0,0]
5.52	[0,0,0,0,0,0]
…	…

In this example, we can see a button press (or trigger) on channel 1 at 2.52 ms that lasts for 3 ms.

Analog IN (ADC)

Analog inputs can be used to record voltages from a variety of devices that have a range of possible outputs, such as an eye tracker. There are 16 ADC channels on the PROPixx controller. When recording analogue data, you must select a rate at which to acquire data, the ADC in the PROPixx controller can acquire at up to 200 kHz. When data is recorded, it can be accessed in a buffer. This buffer will have the data as well as timetags.

Analog OUT (DAC)

Analog outputs can be used to send voltages to a variety of devices that have a range of possible states or that require power to function, such as a reward system. There are four DAC channels on all systems. The DAC have ranges of -10V to 10V. The buffer data for the DAC must have values for all four channels separated. An argument can also specify to which channel the buffer should be sent.

Audio IN

There are two audio inputs on the PROPixx controller depending on the kind of device used to record audio: audio in and mic in. The mic in is for an unpowered source, while the audio in is for a powered line-level source. A gain parameter can be added to the mic in to control the volume of the recorded data.

Audio OUT

All systems can be used to play perfectly synchronized audio using its own codec. Since the audio output uses a codec, it must always be initialized before it can be used. Again, a schedule will play a buffer of audio data. Any audio stimuli must be converted to the buffer format (data and rate) in order to be sent to the PROPixx.

MATLAB & PYTHON DEMOS

Demos can be found by logging in to your MyVPixx account and navigating to https://vpixx.com/myvpixx/downloads-and-updates/

Running an Experiment using E-Prime

Using E-Prime to access your VPixx Devices

To access your VPixx devices from E-Prime, follow this procedure.

1. Open MATLAB as an administrator.

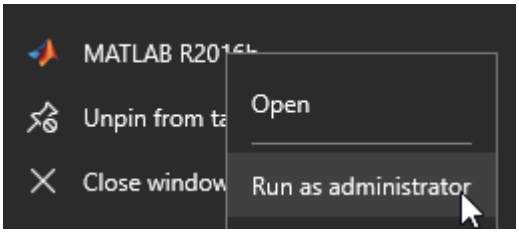


Figure 121 – Running MATLAB as an administrator

2. Once in MATLAB, run the 'regmatlabserver' command.
3. Open E-Studio.
4. Open the experiment which requires communication with a VPixx device.

You must have an *InLine* E-Object in your experiment session.

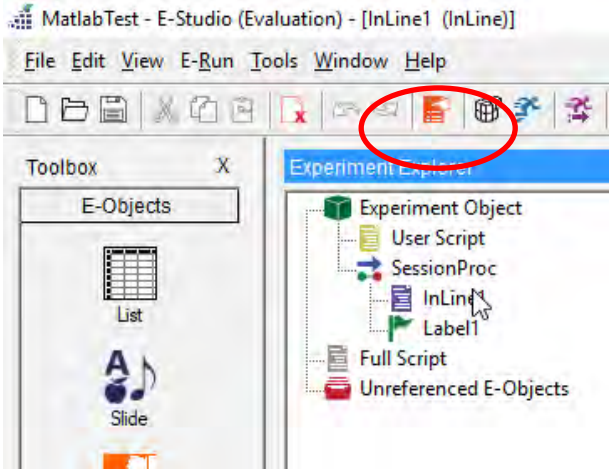


Figure 122 – InLine E-Object

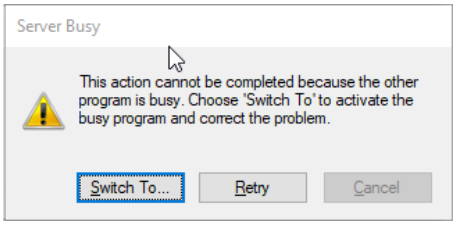
5. In order to communicate with MATLAB, the following code block must be placed at the beginning of the file: *Dim MatLab As Object*

   After all variable declarations, it is possible to create a MATLAB object:
   Set MatLab = CreateObject("Matlab.Application")

In order to pass commands to MATLAB, it is necessary to call the *Execute* function on the MATLAB object.

6. For an experiment involving VPixx devices, it is generally necessary to perform a Datapixx('Open') in E-Studio by issuing *MatLab.Execute("Datapixx('open')").*
   Data to and from MATLAB is exchanged in the form of Strings. For example, in order to get the device time, the following commands can be used:
   Dim Result As String
   Result = MatLab.Execute("Datapixx('RegWrRd')")
   Result = MatLab.Execute("Datapixx('GetTime')")
   The result will now contain the device time in String form.

   While a MATLAB object is created, an amount of lag may be noticed in the application. If this is the case, the following pop-up message will be displayed.



7. Once MATLAB is up and running, the user can press on *Retry* and the experiment will execute.

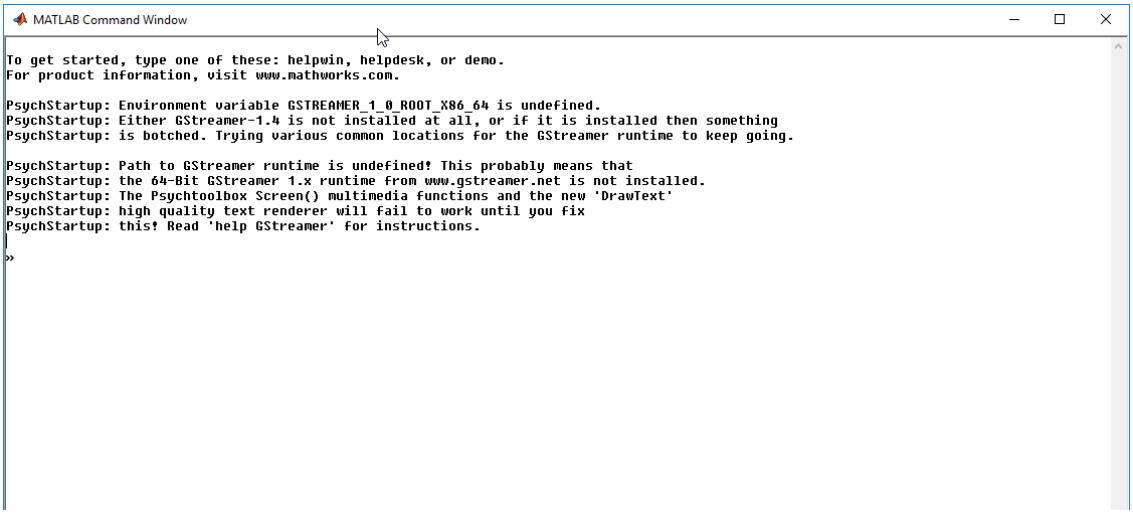


Figure 123 – MATLAB window when the program is ready to accept commands from E-Prime

Recording Eye-Tracking Data

In order to record eye-tracking data while using E-Prime to display stimuli, you must first calibrate the TRACKPixx system. This is done within the PyPixx application.

1. Open *PyPixx*
2. Click on the *Demos* tab
3. Select your TRACKPixx device
4. Set up the appropriate settings (lens, IR Illuminator Intensity, etc.)
5. Click on the ***Tabletop Calibrations*** or ***MRI Calibrations*** button appropriate to your system
6. Run the calibration and save the results
7. Start Recording using the ***Start Recording*** button. This will save a CSV file and update it every 15 seconds with the eye tracking data
8. Start *E-Prime* and run your paradigm

Sending Triggers

To send triggers using E-Prime you must enable a mode called *Pixel Mode*. In this mode, you control the 24 digital outputs of the DATAPixx3 using the color of the first pixel (top-left) of every video frame. This allows any software to control the digital output and it can also replace a photodiode. Instead of showing a big black to white square, a single pixel will do exactly the same thing with digital precision.
Pixel Mode can be enabled via MATLAB or PyPixx and must be enabled prior to the start of your experiment.

Pixel color controls the 24 digital outputs through simple mapping. The pixel's color is represented by three 8-bit numbers (red, green, blue). Those 24 bits are then applied as 0 (OFF) or 1 (ON) to the digital output. Red represents the lowest possible bit, green the middle bits and blue are the highest bits.

Sending Trigger Out when a response is recorded on a RESPONSEPixx MRI

Whenever there is a digital input trigger, it is possible to send a specific waveform on the digital output. This allows you to trigger external equipment such as a parallel port and receive the waveform in E-Prime while still being able to use the great synchronization of our device in a program which does not allow for direct control.

You will need to run a script within MATLAB in order for this mode to be usable in E-Prime.

Please follow the instructions found on this page to enable this mode:
http://www.vpixx.com/manuals/psychtoolbox/html/Demo13.html

Running an experiment using Low Level C API

VPixx Technologies provides a library of functions allowing for the control and management of VPixx devices.

The libraries are detailed on the vpixx website. Simply create an account, login and access these libraries by navigating to www.vpixx.com/manuals/libdpx/html/libdpx.html

ANNEX - VPUTIL SOFTWARE

The vputil software can be run from the main VPixx Technologies directory under the “Software Tools\vputil\bin” folder. This utility allows the user to control some of the VPixx equipment features. It can also be used for generating built-in test patterns from the equipment.

1. Toggle the power switch to the ON position

2. Run the vputil application

Figure 124 – vputil application

The vputil tool automatically detects all VPixx Technologies hardware connected to the computer. The main menu of the application can be displayed by using the Help command.

devsel command

This command selects a device to control if multiple VPixx hardware devices are installed on this computer. Table 13 shows all available devsel command options.

Table 13 – Devsel options

Command	Description
devsel -1	vputil automatically decides which VPixx device to control (default mode)
devsel dpx	Select DATAPixx
devsel vpx	Select VIEWPixx, VIEWPixx /3D or VIEWPixx /EEG
devsel ppc	Select PROPixx Controller
devsel ppx	Select PROPixx Projector
devsel dp2	Select DATAPixx2
Devsel tpx	Select TRACKPixx3
Devsel dp3	Select DATAPixx3
Devsel IOhub	Select I/O Hub

Rev command

This command shows the compatibility between the currently installed vputil and the VPixx equipment firmware revision. This command also provides the equipment's part number and serial number.

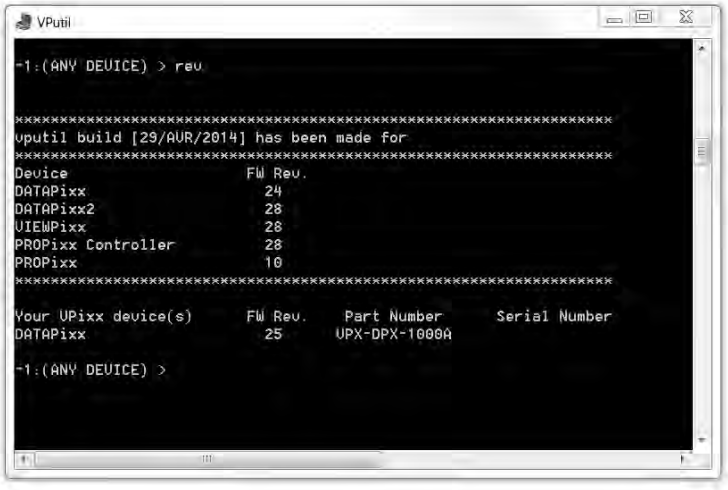


Figure 125 – vputil application (rev command)

0 - command

By typing "0" in the command line, the main menu appears.

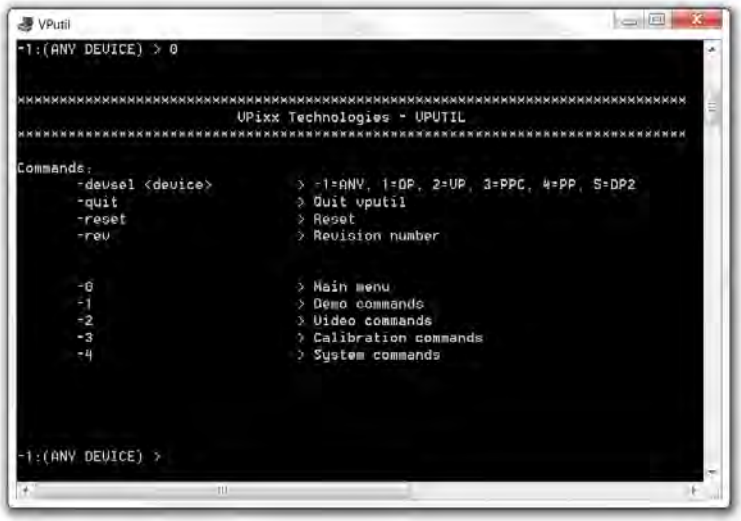


Figure 126 – vputil application (main menu)

1 - Demo commands

This command displays all demos available with the current VPixx hardware. Table 14 shows all available Demo command options.

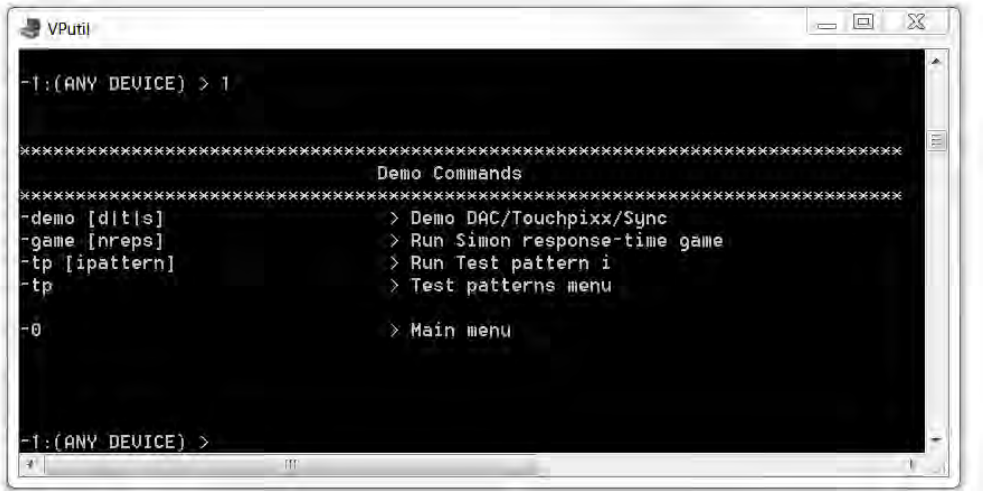


Figure 127 – vputil application (Demo commands)

Demo [d|t|s] : Demo DAC, TOUCHPixx, Sync

Table 14 – Demo command options

Mode	Description
Demo DAC	Analog signal +/- 1V is generated on Analog I/O connector [DAC0 pin 11].
Demo TOUCHPixx	When a TOUCHPixx is used, running this demo will test the hardware.
Demo Sync	The following outputs will be synchronized with the vertical sync pulse. • Pulse on Digital Out [D0 pin #1] • Analog Pulse +/- 1V on DAC0 [DAC0 pin #11] • Sine Wave on AUDIO Out and Internal speaker

game [nreps] : Simon game

Runs Simon game (response-time game). A RESPONSEPixx response box must be connected in the "Digital Input" DB-25 connector of the device. The number of repetitions can be specified with the game command.

tp : Test patterns menu

Table 15 – Test Patterns menu

TP	Argument	Description
0		Return to default video mode (DVI Input mode)
1		RGB waves
2	bitDepth	M16 video mode contrast sensitivity chart with bit depth <bitDepth range: 1-12>
3	bitDepth	C48 video mode color ramp with bit depth <bitDepth range: 1-12>
4	R G B	RGB bars, <R G B range 0-1>
5		VIEWPixx Scanning backlight
6		Stereo goggles
10		Border pixel rings
11		RGB waves in software
12	Lum	All white for calibration <Lum range: 0-1>
13		VIEWPixx CLUT tests
4		Rise/Fall time display
15		Looming
19	#pics	3D Pictures Slideshow <#pic range: 1-6>
23	1	Drifting dots (1 for SBL VIEWPixx Button Box Demo)
27	s/q	Start/Quit PROPixx 480Hz Spinning wheels demo

Additionnal information

tp 19 : 3D Pictures Slideshow

Note that the 3DPixx IR Emitter is specially designed to be used with VPixx Technologies equipment and is not compatible with non VPixx Technologies consumer 3D equipment.

tp 27: PROPixx 480Hz Spinning Wheel demo

Unlike other test patterns, you have to type “tp 27 q” command to quit this mode.

2 - Video commands

This displays the video commands available with the current VPixx hardware.

Figure 128 – vputil application (Video commands)

edid : Custom resolution

Set custom video timing and resolution by modifying the “Extended display identification data” of the VPixx device.

Table 16 – edid command description

Vpixx Device	Description and limitation
DATAPixx & DATAPixx2	Users can write two custom video timings to EDID in flash.
PROPixx & PROPixx Controller	Users can select two of the following resolution and video timing presets to EDID. • 1920 x 1080 @ 60 Hz • 1920 x 1080 @ 120 Hz • 1280 x 1024 @ 60 Hz • 1280 x 1024 @ 120 Hz • 1280 x 1024 @ 180 Hz • 1280 x 720 @ 60 Hz • 1280 x 720 @ 120 Hz • 1280 x 720 @ 240 Hz • 1024 x 768 @ 60 Hz • 1024 x 768 @ 120 Hz • 1024 x 768 @ 240 Hz
VIEWPixx /3D	Users can select the following resolution and video timing preset to EDID. For more information, please see the **Top-Bottom 3D format** section below**.** • 1920 x 2322 @ 60 Hz
VIEWPixx & VIEWPixx /EEG	Function not available.

Top-Bottom 3D Format

In order to present 3D video on a VIEWPixx /3D, you may use the top-bottom 3D format. Instead of sending a 1920 x 1080 @ 120Hz signal, the graphic card must send 1920 x 2232 @ 60 Hz signal to the monitor where:

- Lines 0 to 1079 Left Eye image
- Lines 1080 to 1242 Non visible active blanking
- Lines 1080 to 1242 Right Eye image

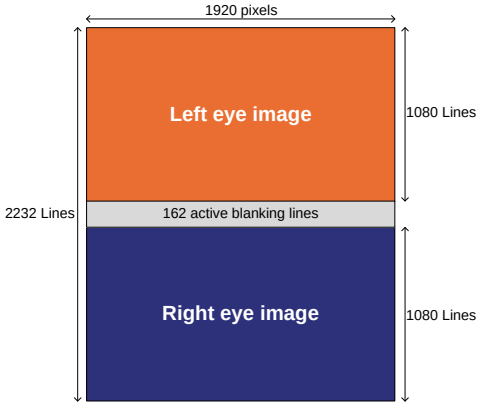


Figure 129 – Top-Bottom 3D format

Frame packing sends the left and right eye images to the monitor simultaneously, stacked on top of one another with a small space between them. Essentially, the source sends one giant double-height frame with active blanking lines instead of two smaller frames. This signal is transmitted at 60Hz, the VIEWPixx then separates the two images and displays them sequentially at 120 Hz.

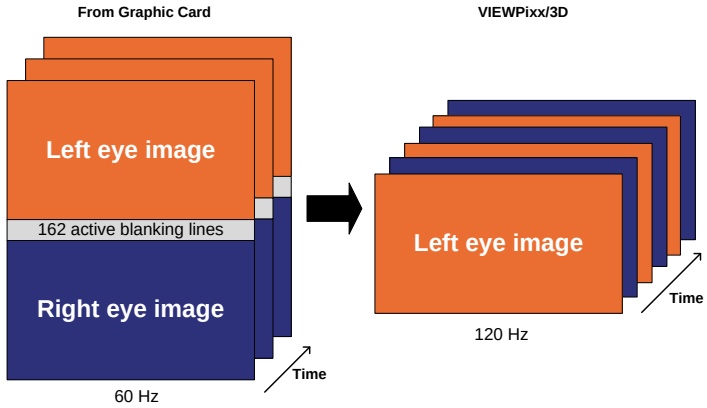


Figure 130 – Top-Bottom 3D format timing

vline <npixels> : Video Line buffer

Reads red, green and blue values of “npixels” from the VPixx device line buffer. This can only return pixel values from the first line.

vmode <mode> : Video mode

Sets video color translation mode. (Not available for DATAPixx2)

Table 17 – Vmode description

Mode	Description
C24	Straight pass-through from DVI 8-bit (or HDMI "deep" 10/12-bit) RGB to VGA 8/10/12-bit RGB
L48	DVI RED[7:0] is used as an index into a 256-entry 16-bit RGB color lookup table
M16	DVI RED[7:0] & GREEN[7:0] concatenate into a VGA 16-bit value sent to all three RGB components. Also implements a CLUT overlay which is indexed by a non-zero blue component
C48	Even/Odd pixel RED/GREEN/BLUE[7:0] concatenate to generate 16-bit RGB components at half the horizontal resolution

vsplit <on|off|auto> : Video Split screen mode

(DATAPixx and DATAPixx2 only) video output #1 shows left half of the video image, video output #2 shows right half of the video image. In automatic mode, DATAPixx will automatically split video across the two video outputs if the horizontal resolution is at least twice the vertical resolution (default mode).

vstereo <on|off|auto> : Vertical stereo mode

(DATAPixx only) the top/bottom halves of the input image are output in two sequential video frames. VESA L/R output is set to 1 when the first frame (left eye) is displayed, and set to 0 when the second frame (right eye) is displayed. In automatic mode, Vertical stereo is enabled automatically when the vertical resolution is greater than the horizontal resolution (default mode).

3 - Calibration commands

Calibration commands shows the commands available with the «i1Pro spectrophotometer» or the «i1Display Pro» colorimeter.

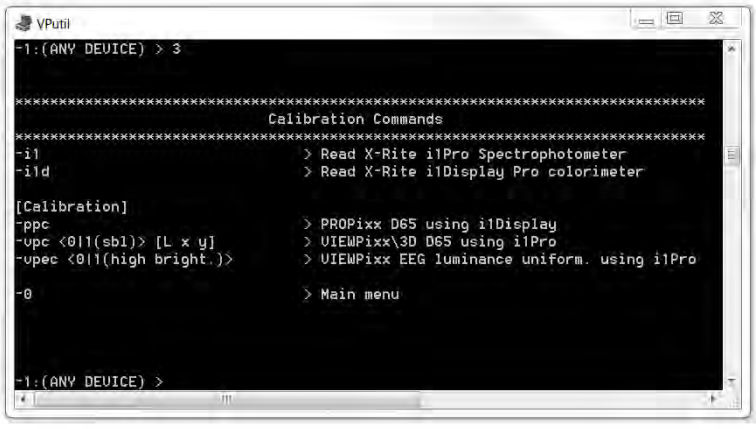


Figure 131 – vputil application (Calibration commands)

I1 : i1Pro Read

Reads L(x,y) luminance and chromaticity values using the i1Pro spectrophotometer. This command may require a quick calibration.

Figure 132 – X-Rite i1Pro spectrophotometer

I1d : i1Display Read

Reads L(x,y) luminance and chromaticity values using the i1Display Pro colorimeter.

Figure 133 – X-Rite i1Display Pro colorimeter

*The **I1d** command can be used only with the PROPixx projector. For use with VIEWPixx series monitors, please call VPixx Technologies or email* support@vpixx.com

ppc : PROPixx D65 calibration (X-Rite i1Display Pro device)

This command allows the user to perform a D65 calibration on the PROPixx projector. The calibration process takes approximately 30 minutes. Refer to the PROPixx calibration user manual for more information.

vpc : VIEWPixx D65 calibration (X-Rite i1Pro device)

Table 18 – VIEWPixx D65 calibration description

Command	Description
vpc 0	D65 calibration for standard backlight mode
vpc 1	D65 calibration for scanning backlight mode
vpc 0 [L x y]	calibration with a custom luminance and chromaticity for standard backlight mode
vpc 1 [L x y]	Calibration with a custom luminance and chromaticity for scanning backlight mode

4 - System commands

This command allows a user to have low level control of the equipment.

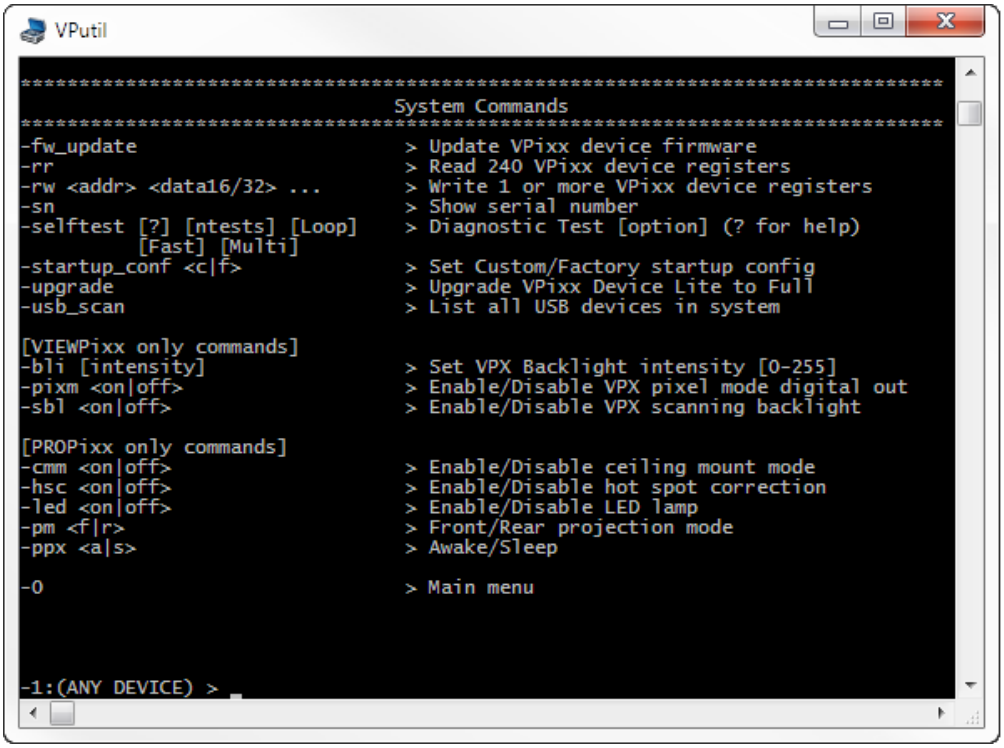


Figure 134 – vputil application (System commands)

fw_update : Update VPixx device firmware

Updates the VPixx device firmware to the latest version.

rr : Read register

Reads the 240 register values of the VPixx device. If multiple devices are installed, use the devsel command before using the read register command.

rw <add> <data> : Register write

Writes into one of the 240 registers of your VPixx device. This function is not intended to be used without assistance. Before using this command, please call VPixx Technologies or send an email to support@vpixx.com.

sn : Serial Number

Shows the serial numbers of connected VPixx Technologies devices.

selftest : Diagnostic test

Automated test of VPixx device functionality. Used for hardware problem diagnostics.

startup_conf <c|f> : Custom startup configuration

Table 19 – Custom startup configuration description

Command	Description
startup_conf c	Save current register values or "configuration" as default value for the next power up.
startup_conf f	Set default register values back to the factory configuration.

upgrade : Upgrade VPixx Device

Upgrades VPixx Device I/O subsystem from Lite to Full (adding the Stereo audio stimulator, 16 ADC channels and 4 DAC channels). Before using this command, please call VPixx Technologies or send an email to support@vpixx.com.

usb_scan : USB Scan

Lists all VPixx USB devices connected to the system.

bli <on|off>: Backlight Intensity

(VIEWPixx family only) Sets VIEWPixx backlight intensity (range 0-255).

pixm <on|off>: Digital Output Pixel Mode

(VIEWPixx and VIEWPixx/3D only) Enables/Disables digital output pixel mode. 24 digital output triggers for synchronizing external data acquisition systems such as EEG and eye trackers with microsecond precision.

The triggers can also be used to initiate external stimulators such as TMS. The levels of the 24 TTL outputs are controlled by the 8-bit RGB components of the display's top-left pixel. This strategy is simple to program with any stimulus generation software and guarantees perfect synchronization between video and external hardware.

sbl <on|off>: Scanning Backlight

(VIEWPixx and VIEWPixx/3D only) Enables/Disables scanning backlight.

cmm <on|off>: Ceiling Mount Mode

(PROPixx only) Enables/Disables Ceiling mount mode.

hsc <on|off>: Hot Spot Correction

(PROPixx only) Enables/Disables Hot spot correction.

led <on|off>: LED lamp

(PROPixx only) Enables/Disables LED lamp (light source).

pm <f|r>: Projection Mode

(PROPixx only) Sets front or rear projection mode.

ppx <a|s>: PROPixx awake/sleep

(PROPixx only) Sets the projector in awake or sleep mode.

VPixx Technologies Inc.
630 Clairevue West suite 301
Saint-Bruno, Qc
Canada, J3V 6B4

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