NOVELDA

AN0001

HPD Studio Getting Started Guide

Revision G

14-Sep-2022

Summary

This document will explain how to use Novelda HPD Studio to evaluate the performance of Novelda's UWB sensor products.

Table of Contents

1 Abstract 4
2 Brief Step by Step Procedure 5
3 Difference between X4T24 and X4F103 6
4 Novelda HPD Studio - PC Application 7
4.1 Windows Installation procedure 7
4.2 macOS Installation Procedure 8
4.3 Raspberry Pi Installation Procedure 8
4.4 How to Run the Application 8
4.5 Command Line Interface (CLI) 12
4.6 Adding new .hpd files 15
4.7 Customizable Python Demos 16
4.8 Troubleshooting 17
5 References 18
6 Document History 19
7 Disclaimer 20

1 Abstract

The Novelda UWB sensor demo kit enables evaluation of Novelda´s UWB sensor products. The demo kit contains a radar module and a serial protocol interface module to connect the sensor to the computer.

Sensor output can be visualized in Novelda HPD Studio for Windows/MacOS and there is a command line option available for long-term logging and prototyping. The SW can run also run on Raspberry Pi 3/4.

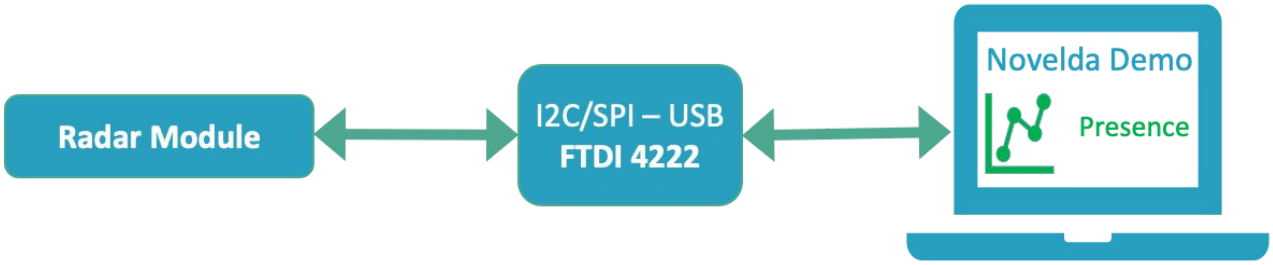


Figure 1 - X4 Demo Kit HW

2 Brief Step by Step Procedure

This section briefly outlines the steps to set up and run the Novelda UWB sensor demo. More details are found in the sections below.

1. Run the Novelda HPD Studio Windows installer.
    a. The installer will automatically install Novelda HPD Studio as well as required drivers.
    b. For other operating systems see more detailed sections further down.
2. Connect the FTDI USB dongle to the laptop with the provided USB cable.
3. Attach the sensor module in a suitable location.
    a. For occupancy / light control the sensor must be placed in the ceiling or inside a ceiling mounted light fixture.
    b. For proximity applications the sensor must be placed vertically on the wall or on the product the sensor is intended for.
    c. Keep in mind that the radar antenna (visible through the transparent plastic cover) must point towards the detection zone.
4. Run the demo application by double clicking Novelda HPD Studio shortcut on the desktop.

> The USB receptacle on the FTDI dongle may be a bit tight. If the demo GUI reports no sensor connected, verify that the USB-C contact is plugged all the way into the receptacle.

3 Difference between X4T24 and X4F103

The X4T24 module has a squared shape, while the X4F103 module has a rectangular shape, as shown in the next picture.

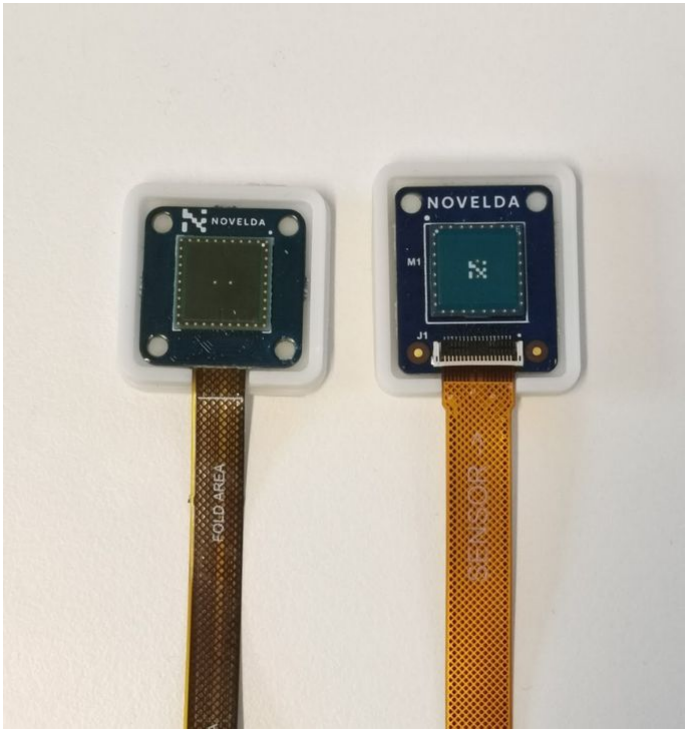
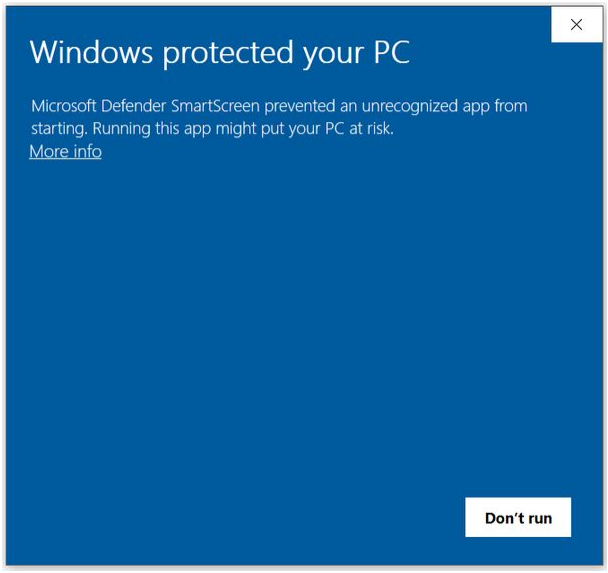


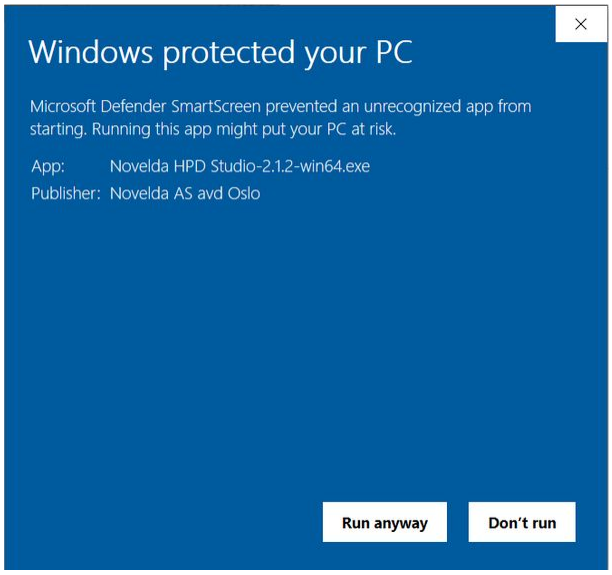
Figure 2 - Left: X4T24, Right: X4F103

4 Novelda HPD Studio - PC Application

4.1 Windows Installation procedure

- Run Novelda HPD studio installer
- The following window from Microsoft Defender SmartScreen will appear. To continue the installation, press “More info” and then “Run anyway”.

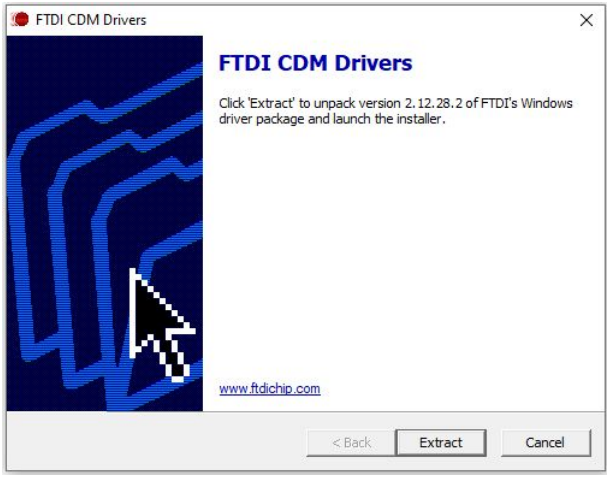


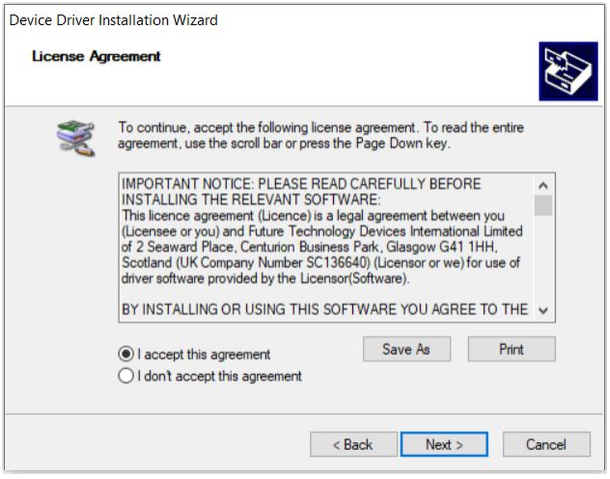


- Follow the instructions from the installer to install the application. In the first window the user can choose to install additional FTDI Drivers, as shown in the next image. This is necessary to run the demo. If the user already have FTDI drivers on their computer, this box can be unchecked.

Figure 3 - Install FTDI Drivers

If the user is downloading FTDI drivers, the following windows will appear after the download of the Novelda HPD Studio application has completed.





4.2 macOS Installation Procedure

- Download and extract the installation zip file.
- Open the folder in terminal.
- Run `xattr -d com.apple.quarantine Novelda\ HPD\ Studio.app`
- Change permissions using `chmod -R 755 "Novelda HPD Studio.app"`
- Double click on the application to run it.

4.3 Raspberry Pi Installation Procedure

- Download and extract the installation zip file.
- Open the folder in terminal.
- Change permissions using `sudo chmod +x "NoveldaHPDStudio-armhf.AppImage"`
- Run the application by `sudo ./NoveldaHPDStudio-armhf.AppImage`

4.4 How to Run the Application

The program communicates with the connected sensor through USB. When the program is launched, the user will have to select some settings to get started.

4.4.1 Procedure

- Select product
  Choose the algorithm to run. The available algorithms and the version will be listed when the GUI is started. The demo currently supports algorithms for the following products:

- Occupancy sensor - ceiling mounted, typically used for controlling luminaires.
- Proximity sensor - vertically mounted, typically inside products such as appliances etc. to control displays or similar.

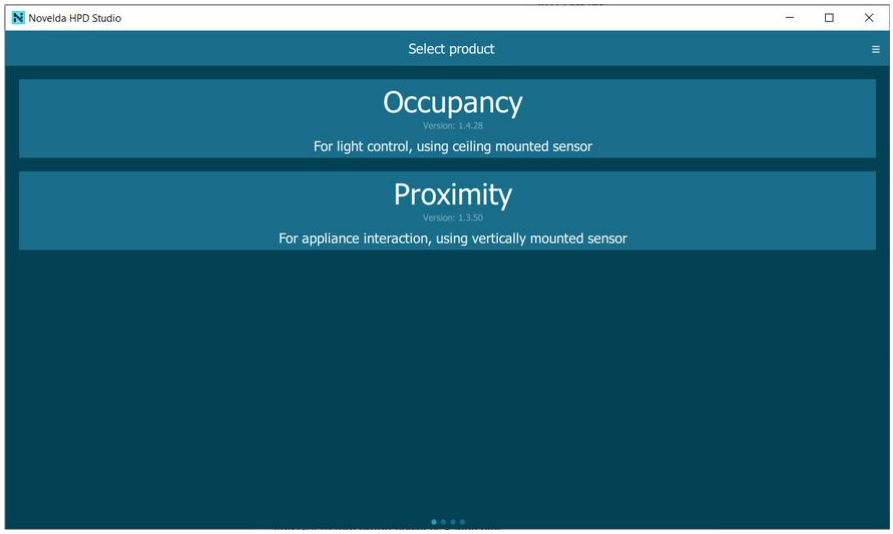


Figure 4 - Select Algorithm

- Select connected sensor
  The user should select the correct sensor from the menu with available sensors. See the previous chapter “Difference between X4T24 and X4F103” on how to differentiate between the modules.

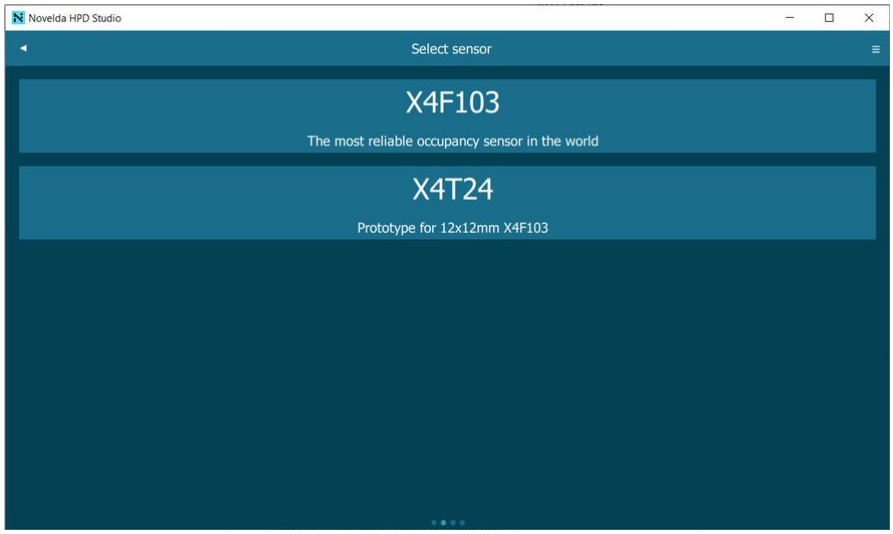


Figure 5 - Select Sensor

- Select range, sensitivity and timeout
  - The user selects a maximum detection range from the drop down menu. The maximum distance will vary based on which algorithm is being used.

- Keep in mind that the sensor can see through most walls and that static objects can act as mirrors. It is a good idea to set the max range as short as possible to avoid false detections and allow a more free choice of sensitivity settings in the next step.

Within the selected detection range all movements will be detected. Right outside the detection zone there is a transition area of approximately 20 cm. Within this area there is a decreasing probability of detection. This is due to the pulse width of the radar signal. Outside this transition zone, there is an absolute rejection of any movements.
The transition zone is illustrated in the figure below.

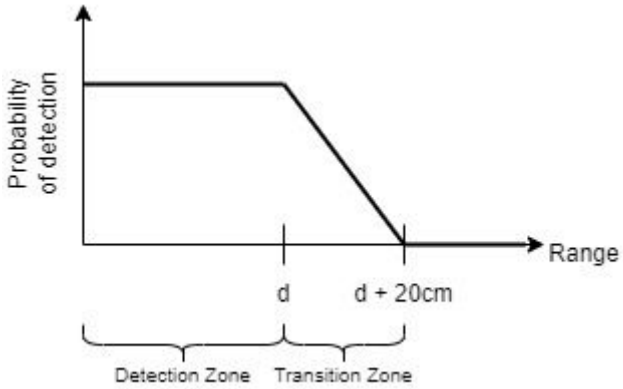


Figure 6 - Detection And Transition Zone

- Sensitivity and Timeout:
  - All the algorithms offer a selection of sensitivity settings. These must be chosen carefully to balance the desired detection probability against the probability of false triggers from unwanted targets in the zone. The optimal setting depends on the sensors environment (the amount of clutter in the detection zone) and the application specific consequences of missed detections and false triggers. Key points to bear in mind is that a longer hold time allows the choice of a less sensitive setting, which in turn will reduce the number of false detections.
  - The timeout decides how long the sensor reports “DETECTION” after the last actual detection. The current signal processing algorithms rely on a certain hold time for robust operation. This hold time is user-configurable within a given range for each algorithm, but Novelda recommends using at least the following:
    - Occupancy: 60 seconds
    - Proximity: 15 seconds

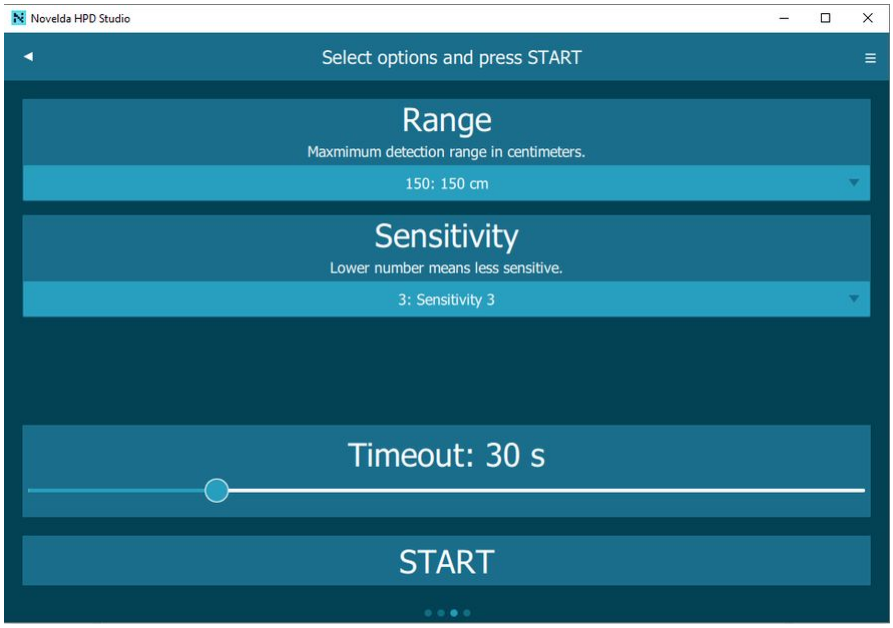


Figure 7 - Select Options

- Start
  When the correct settings are selected, press the “Start” button.
- Change settings
  It is possible to go back to change these options at any time.

When the program starts running, a new page will show the sensor output and the history. The current state (detection/no detection), the timeout countdown and the selected settings are visible on the top of the page. The first plot shows the logic history with the detections (+timeout) marked in green. The sensor history is plotted below that, and shows each detection made. How easy it is to trigger a detection will be affected by the chosen sensitivity.

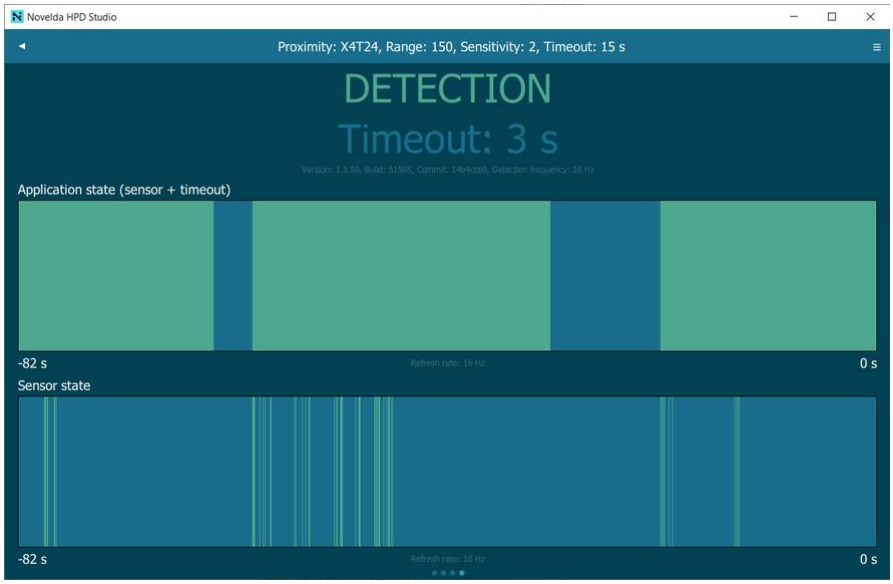


Figure 8 - Sensor Output

4.4.2 Power vs. Range plot

The power vs. range plot can be activated in the settings menu in the top right corner.

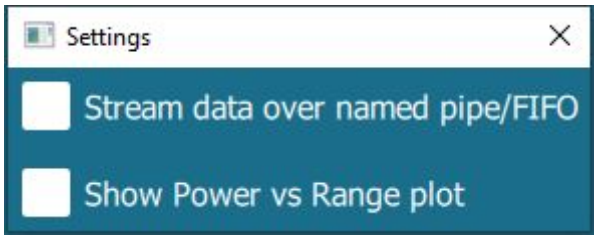


Figure 9 - Settings Menu

When this box is checked the following plot will show up under the existing plots. It plots the signal power at different ranges, and the dotted line shows the detection threshold.

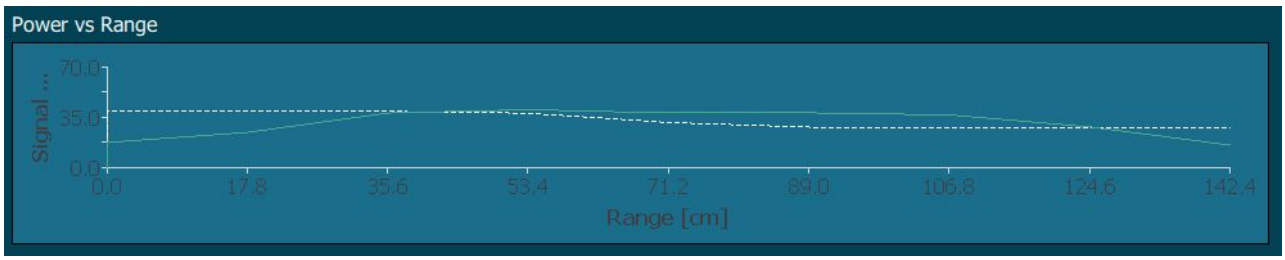


Figure 10 - Power Vs Range Plot

4.5 Command Line Interface (CLI)

The Novelda HPD Studio also has a command line interface which will print the sensor data to the standard output in a semicolon separated format. The format of the output is: timestamp; distance; presence (0/1);

The CLI includes the same options for range, sensitivity and timeout as in the GUI. To print all options for CLI run the “Novelda HPD Studio.exe” (“NoveldaHPDStudio-armhf.AppImage” for RPi) followed by -h or --help in the command line.

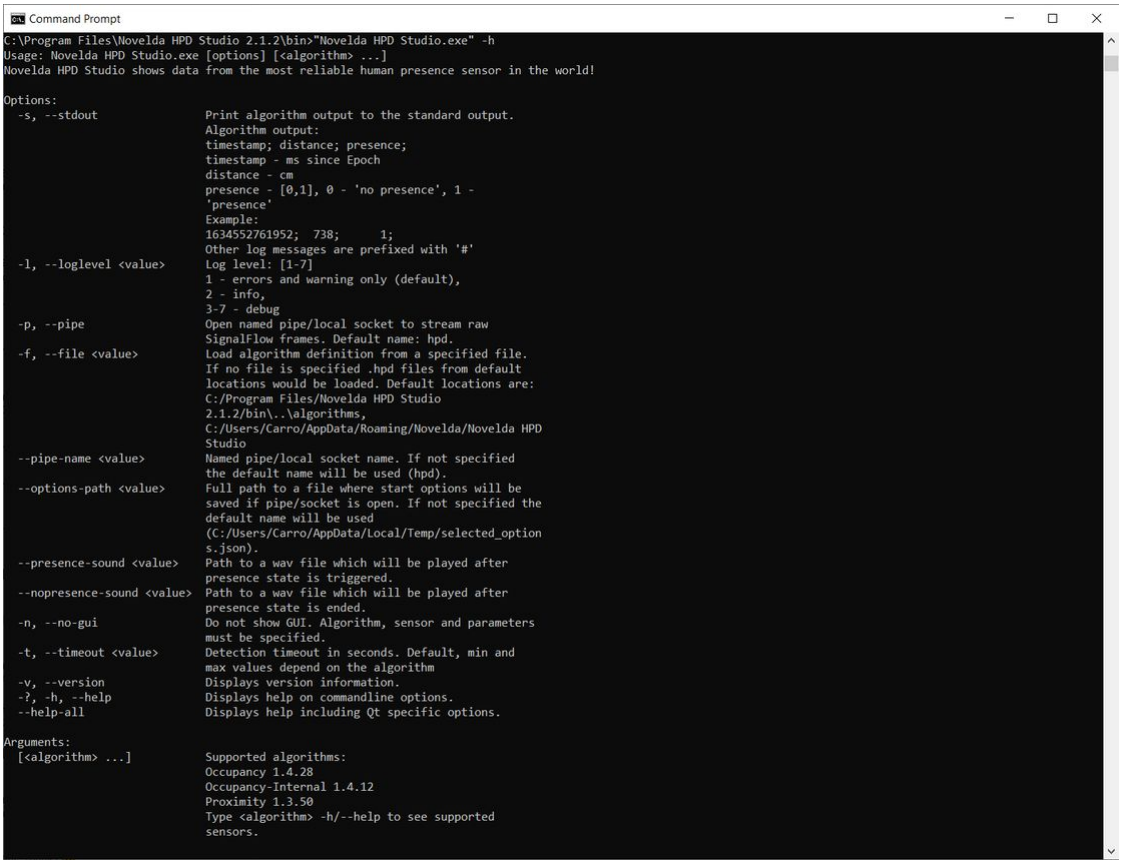


Figure 11 - Run With -H To Display All Options

To run with default settings, run the “Novelda HPD Studio.exe” followed by -s, -n, the wanted algorithm and the connected sensor, as shown in the next figure. The supported algorithms are listed at the bottom of the page when the help page is printed, as shown in the previous figure.

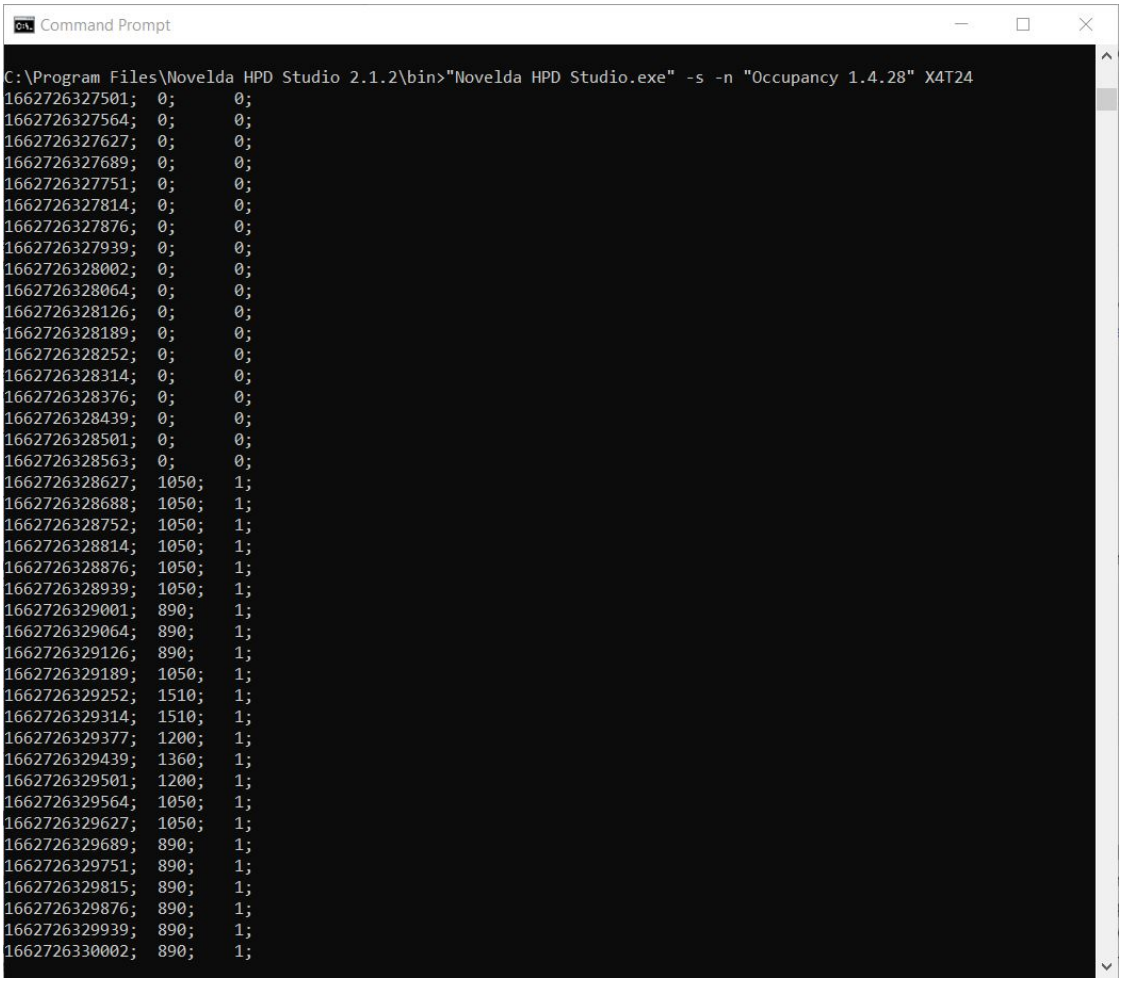


Figure 12 - Running Program With Default Settings.

The settings can be adjusted by adding the timeout (-t), range and sensitivity manually. To print the options for range and sensitivity, the following format is used: ".\Novelda HPD Studio.exe [algorithm] <module> [parameter] -h". (Example: .\Novelda HPD Studio.exe OccupancyMinimal X4T24 range -h)
An example on how to run the program with other settings are shown in the figure below.

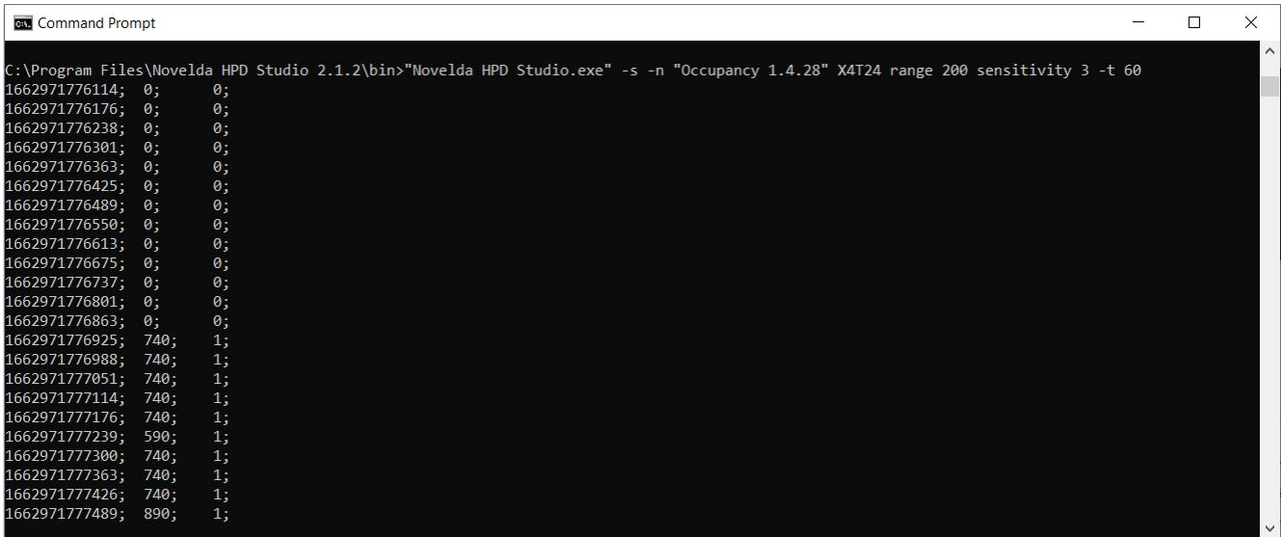


Figure 13 - Running Program With Adjusted Settings

4.5.1 MacOS Instructions

To run from CLI on macOS you have to navigate to the actual executable file inside the .app package. The command call to run HPD Studio is thus as shown below:

```
NoveldaHPDStudio_x64-osx % Novelda\ HPD\ Studio.app/Contents/MacOS/"Novelda HPD Studio"
```

4.6 Adding new .hpd files

If a new version of the algorithm is available, users can easily add this to their existing Novelda HPD Studio application. The algorithm will be provided as a .hpd file which should be added to the algorithms folder, as shown in the next image.

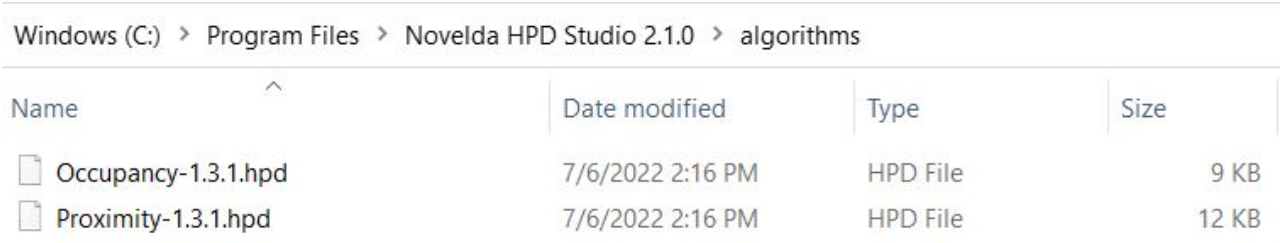


Figure 14 - Algorithms Folder

Novelda HPD Studio also supports drag-and-drop of .hpd files. This means that the user can drag .hpd-files directly from the download folder and into the application, see the next picture. The new algorithm should appear immediately in Novelda HPD Studio, and in the algorithms folder.

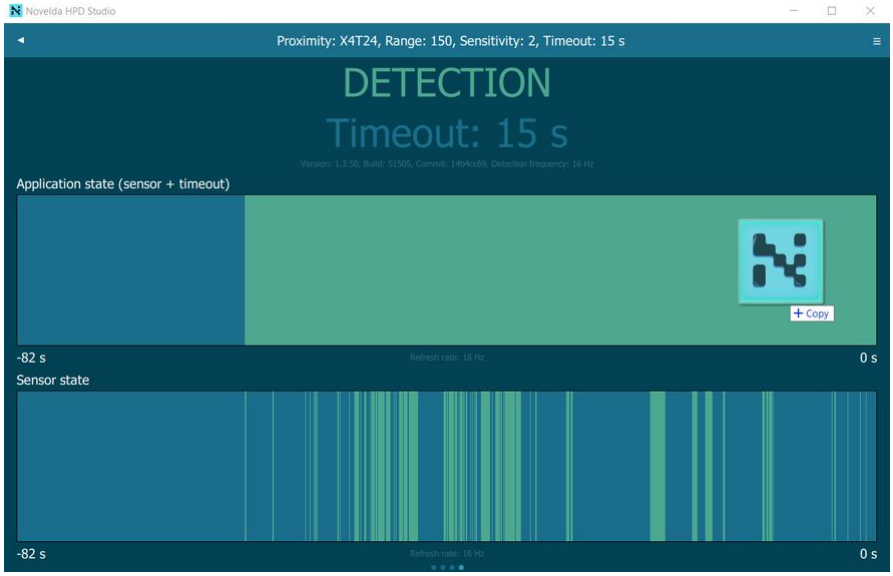


Figure 15 - Drag .Hpd-Files Into The Application

If double clicking a .hpd-file, Novelda HPD Studio will open with this specific algorithm. This will however not add the file to the correct folder, and the algorithm will not be there the next time the user open HPD Studio in the normal way.

4.7 Customizable Python Demos

Novelda HPD Studio supports streaming sensor data to a temporary buffer. This functionality can be used to write custom application logic on top of, or in place of, the standard application logic in HPD studio (which is a simple timer). To use this functionality, the "Stream data over named pipe/FIFO" box can be checked under the settings menu in the HPD Studio application. It can also be activated via the Command Line Interface with the following argument `-p` (it is still possible to select and change options via the Graphical User Interface (GUI) if that is desired). In PowerShell, the full command looks like this, assuming you have navigated to the install/bin directory:

```
& '.\Novelda HPD Studio.exe' -p
```

If your application does not require visual output, you may want to specify the `-n/--no-gui` flag to start streaming data without having to select options via the GUI. When using the `-n` flag, it is necessary to explicitly specify which algorithm you want to run, and which sensor you are using. All other options will use the default value unless explicitly specified. Minimalistic example in PowerShell:

```
& '.\Novelda HPD Studio.exe' "Occupancy 1.1.13" X4T24 -n -p
```

For more details, please refer to the section titled "Command Line Interface (CLI)".

4.7.1 Picking up the streamed data in Python.

Once Novelda HPD Studio is running, and streaming data to a buffer, the data can be picked up by another program. Novelda provides a collection of python scripts for quickly getting started with reading this data.

These scripts can be found in the zipped folder “NoveldaPythonDemo.zip” that is located on: https://novelda.com/x4-demo-kit/. Unzip this folder in any location locally on your computer.

First-time use

Before first-time use of the Python scripts, it’s important to install all dependencies. First and foremost, you will need to install Python 3 and add the location of this installation to your PATH environment variable. Open a terminal window, like PowerShell, and navigate to the “NoveldaPythonDemo” folder. Execute the following commands to install the required python packages:

```
python -m pip install novelda_serialization-0.1.1-py3-none-any.whl
python -m pip install -r requirements.txt
```

Running examples

There are two examples for getting started with reading streamed data in Python. Run the command `python NAME_OF_EXAMPLE` to run the wanted example from this folder.

The script called `ex1_basic_app_logic.py` is the most minimalistic example. It shows how to read which settings Novelda HPD Studio is currently using, including where to find the data buffer, and how to attach a callback function to this data stream so that you can write your own application logic. The callback function is called each time the result of a new radar frame is added to the buffer.

The script called `ex2_basic_app_logic_with_holdtime.py` is slightly more advanced, and expands on example 1 by adding a state machine to keep track of a variable timer. This timer is used to add a hold time to the sensor data, which is useful in applications where the sensor is waking up another system. For Novelda’s OccupancyMinimal and ProximityMinimal software-products, it is recommended to always use a hold time for best performance (minimum recommended hold-time differs between products). The hold time can be specified when starting Novelda HPD Studio - either in the GUI or by using the `-t` option in the CLI.

4.8 Troubleshooting

When starting up the Novelda HPD Studio application, an error message like the one below can be displayed if there are something that doesn’t work as expected.

Figure 16 - Novelda HPD Studio Error Message

To troubleshoot this, please start the application from command line using the “-l7” parameter:
`.\Novelda HPD Studio.exe "Algorithm" -l7`
The printed output will show a more detailed error message, that can be debugged in collaboration with Novelda.

5 References

[1] Novelda X4 Demo Kit web page: https://novelda.com/x4-demo-kit/

[2] NEMA standard WD 7-2011 (R2016): https://www.nema.org/standards/view/occupancy-motion-sensors-standard

6 Document History

Revision	Release Date	Changes
G	12. sep. 2022	• Updated for new .hpd features • Updated for changes in the GUI.
F	21. apr. 2022	• Changed title • Removed all algorithms/products descriptions • Removed info about the kit. • Removed information on how to remove plastic cover
E	8. apr. 2022	• Updated for Occupancy sensing. • Updated for Proximity sensing. • Removed support for X4C007. • Added information about python demos
D	30. nov. 2021	• Added sensitivity settings. • Added detection limits for ceiling mounted X4T24.
C	29. okt. 2021	• Fixed minor typos. • Added information about removing plastic housing. • Added information about Windows CLI.
B	20. sep. 2021	• Changed title to reflect new algorithms. • Changes throughout the document to reflect new HW, new algorithms and GUI updates.
A	21. mai 2021	• First release

7 Disclaimer

The information in this document is provided in connection with Novelda products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Novelda products. EXCEPT AS SET FORTH IN THE NOVELDA TERMS AND CONDITIONS OF SALES LOCATED ON THE NOVELDA WEBSITE, NOVELDA ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL NOVELDA BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF NOVELDA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Novelda makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Novelda does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Novelda products are not suitable for, and shall not be used in, automotive applications. Novelda products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.