

Xavia Interfacing

Xavia offers two ways to interact with its data:

1. A web interface that includes **a live point cloud viewer**.
2. **An SDK to support integration** in applications and solutions.

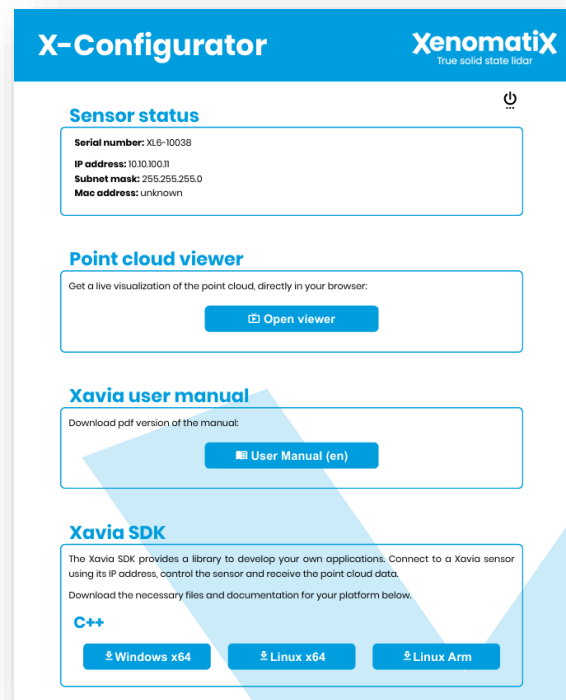
Together these two ways cover all use cases:

- The on-board point cloud viewer works anywhere, all the time, on all major browsers. No need for installation, no compatibility issues.
- Visualizing what the sensor sees helps with installation and setting up the LiDAR to your application needs.
- The real-time, optimized SDK allows you to feed the point cloud data into your applications for further processing. Use the data to monitor your area, activate your processes, count occurrences, or any other action that suits your application.
- Easy to use and filled with examples and boilerplate code, the SDK will reduce integration time of Xavia to only a few hours.

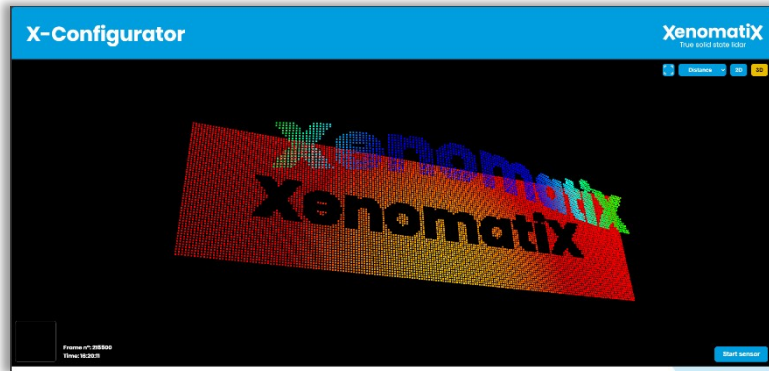
XenomatiX focusses on the sensor, so you can focus on the solution.

Web interface

The web interface is available at all times by browsing to Xavia's IP address. No separate installation is required. In the web interface, key information of the sensor is displayed on the front page. Download links for the user manual and SDK archives are provided. All downloads are from the Xavia itself and no internet connection is required to access these files.



The point cloud viewer in the web interface provides a 2D or 3D view of the point cloud. Color coding of the points can be selected to match the preferred view. The viewport can be manipulated to visualize the point cloud from different angles. Through this responsive viewer, you can easily install and setup the LiDAR according to the needs of your application.



SDK

Downloadable directly from Xavia, the SDK provides a C++ library for easy integration of the sensor into your solutions. This allows integrators to build turn-key solutions using Xavia or fusing Xavia with other sensors and devices.

The SDK is a modern C++17, object-oriented framework that simplifies the interaction with Xavia to 3 operations:

1. Configure how you will use the SDK
2. Connect to Xavia
3. Start streaming

This modern approach fully supports high quality code development using automated (unit) testing, modularity and SOLID principles.

The following code fragment demonstrates both the brevity and the self-explanatory, easy to read language the SDK provides. The result is that integration will not take weeks, but a matter of hours.

```

1  int main()
2  {
3      // CONFIGURATION
4      xavia::sdk::SensorFactory factory;
5      factory.SetPointCloudCallback(OnNewPointCloud);
6
7      // CONNECT TO SENSOR
8      auto sensor = factory.Build();
9
10     // RUN IT
11     sensor->Start();
12     std::this_thread::sleep_for(std::chrono::seconds(10));
13     sensor->Stop();
14
15     return EXIT_SUCCESS;
16 }
17
18 void OnNewPointCloud(std::unique_ptr<xavia::sdk::IPointCloud> pc)
19 {
20     std::string info = "New point cloud line received: "
21         + pc->GetMetaData().frameNumber
22         + ". First point: ("
23         + pc->GetXmm()[0] + ", " pc->GetYmm()[0] + ", " pc->GetZmm()[0] + ")";
24     std::cout << info << std::endl;
25 }

```

Through callbacks the point cloud information is continuously streamed to your application and the 3D reconstructed coordinates (X, Y, Z) of each measured point are available in real time. On top of this information, the SDK provides the radial distance to the sensor, the reflectivity (in %), a confidence level and the amount of ambient light for each point. Each data point is uniquely identifiable using timestamps, frame counters offering flexibility of use. At the same time the data is vectorized to provide the necessary optimization opportunities for real-time applications.

With support for Linux and Windows x64, the SDK can be used on any desktop or laptop. The support for Linux ARM64 provides the opportunity to use the SDK on edge computing devices like the Nvidia Jetson, NXP i.MX and TI, which are ideal for integrations with a small form factor.