Zebra Aurora™ Vision Studio Support for Photoneo 3D Sensors using GenICam

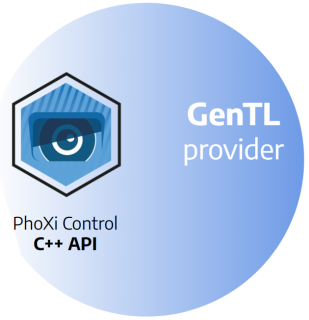
User guide on using the GenICam interface in Zebra Aurora™ Vision Studio (AVS)

What is GenICam?

The Generic Interface for Cameras standard is the base for plug & play handling of cameras and devices. It was developed by the European Machine Vision Association (EMVA) (GenICam – EMVA).

GenICam with Photoneo devices

GenICam support was introduced to Photoneo devices with PhoXi Control 1.8.2. The GenICam functionality is provided via the GenTL library that works as a wrapper around the PhoXi Control C++ API. PhoXi Control has to be running in order to use the GenICam interface.



What is Zebra Aurora™ Vision Studio?

It is machine vision software that is based on visual data flow programming and comes with a comprehensive set of image analysis tools. Typical applications include industrial quality inspection and robot guidance, using both ruler-based and deep neural networks-based algorithms.

GenICam support in Zebra Aurora™ Vision Studio

The requirements to run the example with GenICam

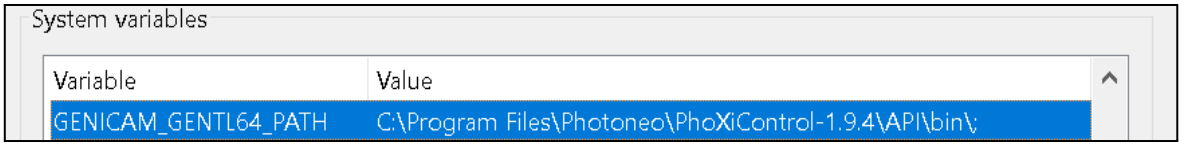
- Zebra Aurora™ Vision Studio 5.2.10.92454 or later
- PhoXi Control 1.9.4 or later

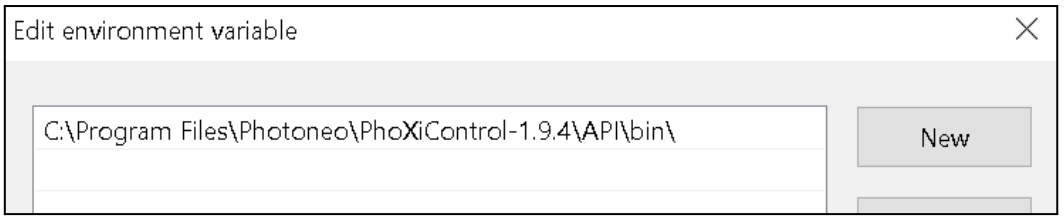
Note: *A lite version of Zebra Aurora™ Vision Studio is not sufficient to use the GenICam features*

Running the example

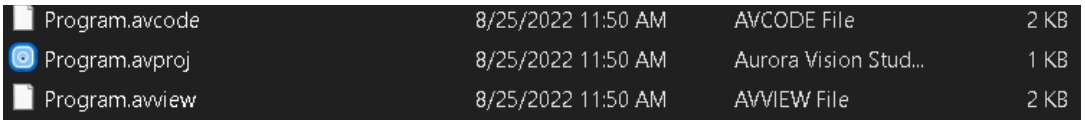
- Download the example project
- Install Zebra Aurora™ Vision Studio, then install and launch PhoXi Control

- In the *Environment Variables*, double-check if the *GENICAM_GENTL_PATH* variable is set correctly to the PhoXi Control *installation directory|API|bin* (default at C:\Program Files\Photoneo\PhoXiControl-1.X.X\API\bin)

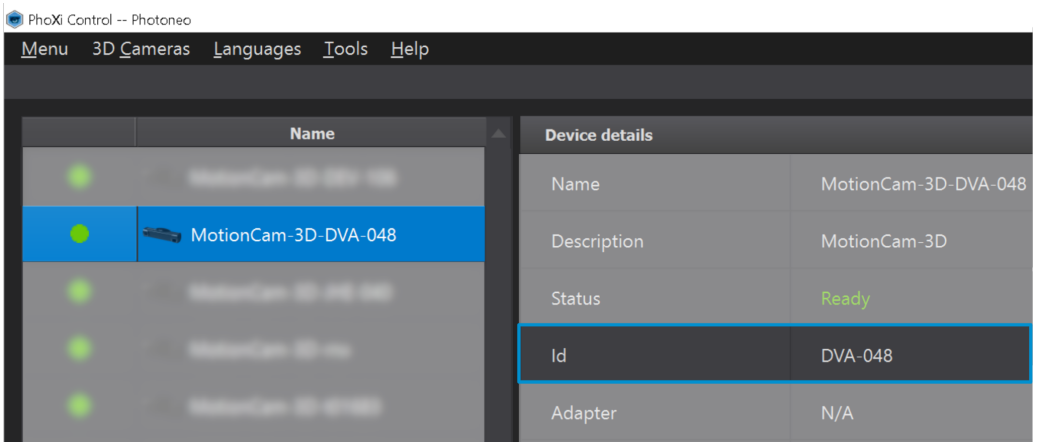




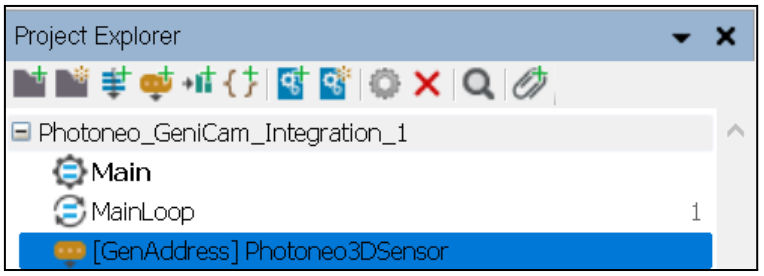
- Open the downloaded example and extract it in a folder with read/write permissions
- Run the *.avproj file



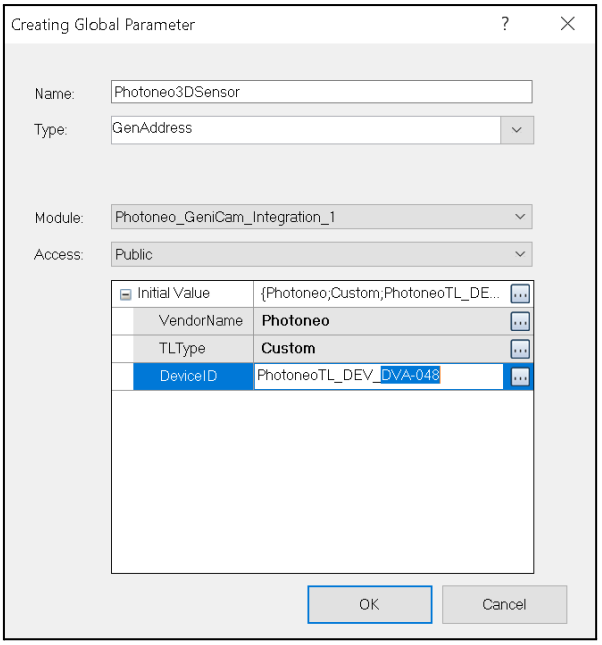
- In PhoXi Control, find and copy the ID of the device (using CTRL + C)



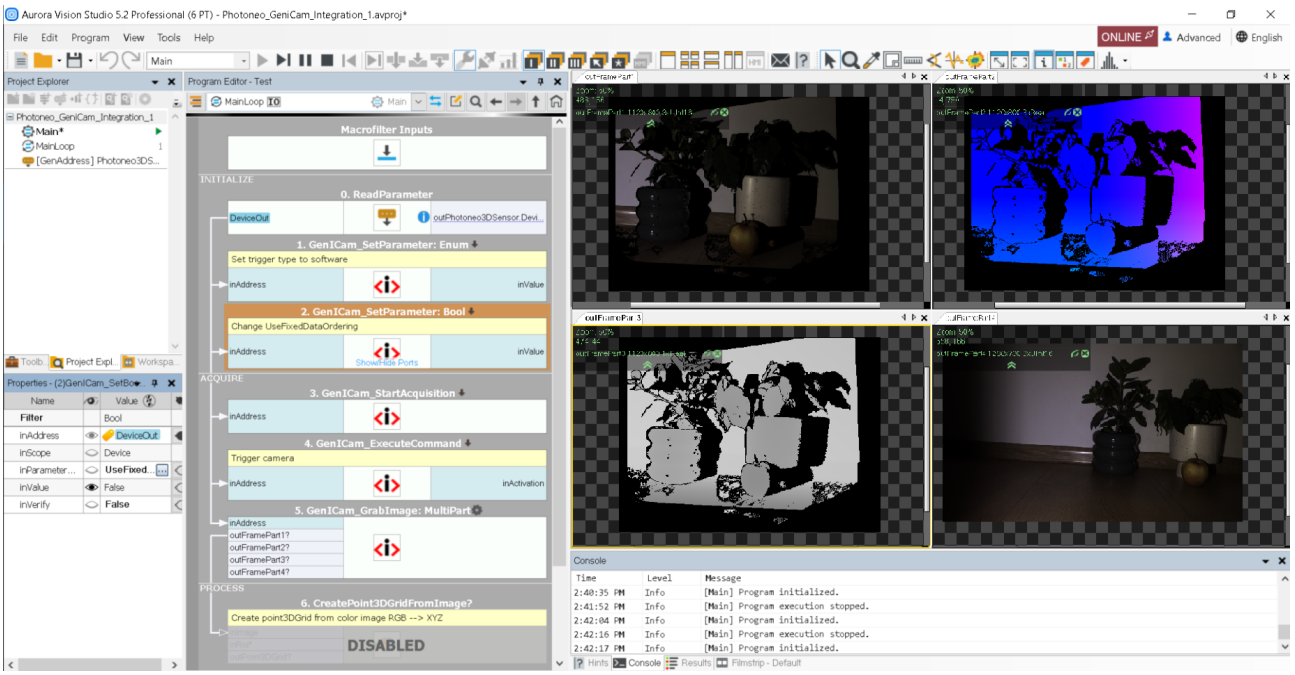
- In AVS, the ***Project Explorer*** tab contains a global parameter ***Photoneo3DSensor*** - that is the address of the camera to which the program connects. Change the **DEVICEID** to the device ID of the Photoneo 3D Sensor you operate.



Example: The device ID is *DVA-048*. The **DEVICEID** field in the AVS needs to contain the whole *PhotoneoTL_DEV_* prefix and the device ID - in this case, *PhotoneoTL_DEV_DVA-048*



- Run the project in Aurora Vision Studio, which automatically connects to the specified device and initiates a freerun acquisition on the device. Output can be observed on the right side of the UI.



- Aurora Vision Studio allows observing of up to 4 output maps at the same time. Therefore, the **UseFixedDataOrdering** parameter for the device needs to be set to **false** (done automatically in the example).

	Order	Output map
UsedFixedDataOrdering set to **true** keeps the default order of the output maps even if one or more are disabled. When it is set to **false**, disabling any of the output maps results in a shift of the subsequent output maps.	0	Texture or TextureRGB
	1	PointCloud [X,Y,Z,...]
	2	NormalMap [X,Y,Z,...]
	3	DepthMap
	4	ConfidenceMap
	5	EventMap
	6	ColorCameraImage

- Example: NormalMap, ConfidenceMap, and EventMap are disabled (blue represents the visible output maps in AVS).

UseFixedDataOrdering = True

Order	Output map
0	Texture or TextureRGB
1	PointCloud [X,Y,Z,...]
2	Empty
3	DepthMap

UseFixedDataOrdering = False

Order	Output map
0	Texture or TextureRGB
1	PointCloud [X,Y,Z,...]
2	DepthMap
3	ColorCameraImage

4	Empty
5	Empty
6	ColorCameraImage

4	Empty
5	Empty
6	Empty

Note: *EventMap a feature exclusive* to MotionCam-3D (Color) and *TextureRGB* and *ColorCameraImage* are exclusive to MotionCam-3D Color.