Photoneo Add-In Manual

What is a Photoneo Add-In?

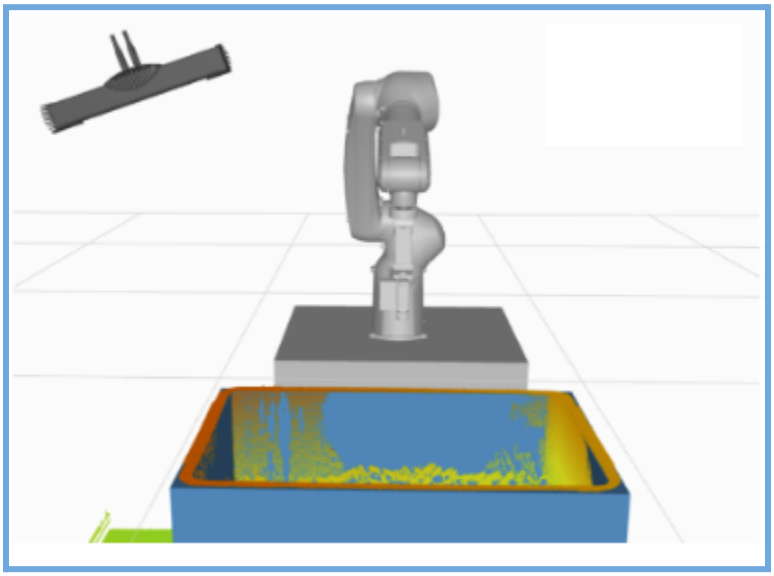
Photoneo is a leading provider of robotic vision and intelligence. Based on a patented 3D technology, Photoneo developed the world's highest-resolution and highest-accuracy 3D camera, thus unlocking the full potential of powerful, reliable, and fast machine learning and also reducing the training and deployment time.

The Photoneo add-in installs all necessary robotic modules and performs the required reconfiguration of the controller to save as much time as possible when starting a new project.

How you can install Photoneo devices

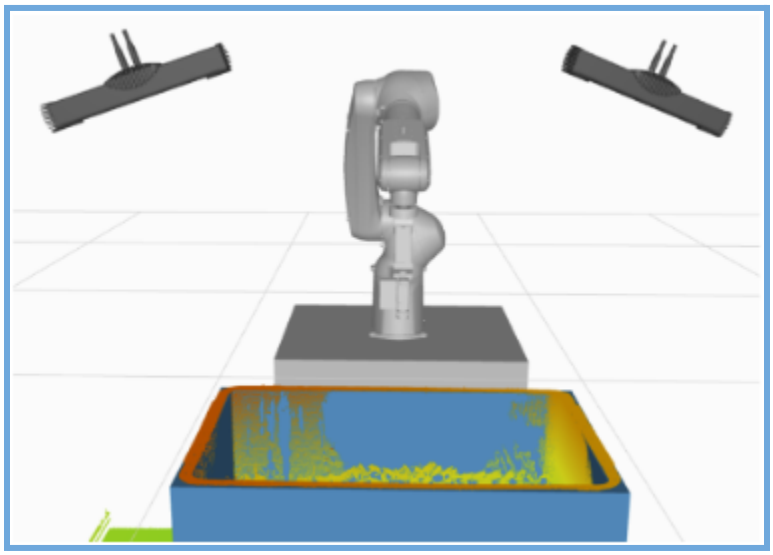
Extrinsic

The sensor is calibrated with the robot.



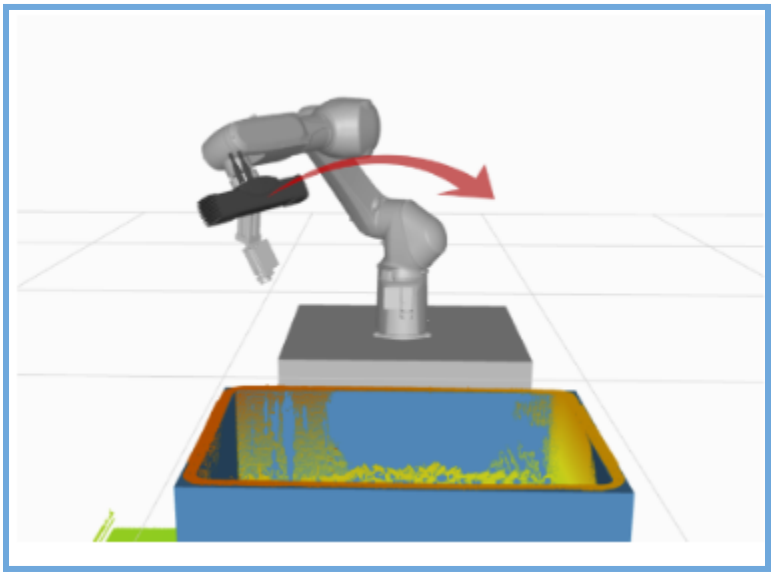
Multiple vision static

One primary sensor is calibrated with the robot, while secondary sensors are calibrated relative to the primary sensor. Scanning is triggered sequentially from each sensor.



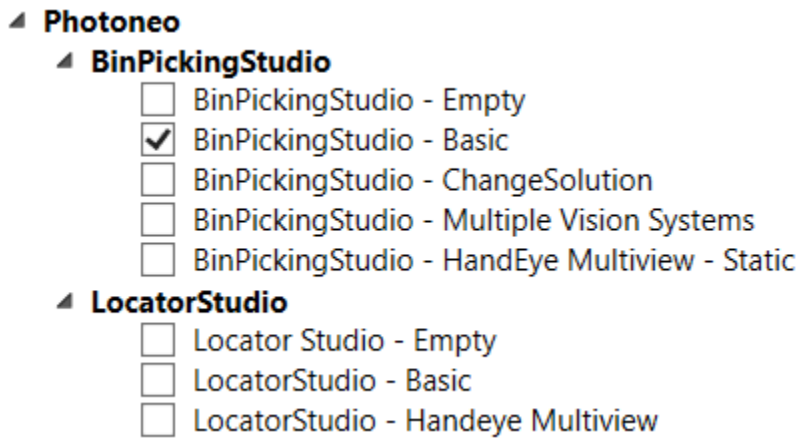
HandEye

This approach uses a Photoneo 3D sensor mounted on the robotic arm. This allows for virtually unlimited scans from various viewpoints instead of just a single scan.



Different modules according to the application

In the Photoneo Add-in, you'll find a variety of robotic module examples and a comprehensive library of requests for communication between the robot and the vision controller.



The modules ensuring communication between the robot and visual controller are:

- PhotoneoCommon.sys (Bin picking studio)
- CustomerDefinition.mod (Bin picking studio)
- PhotoneoLocator.sys (Locator studio)

The Empty module means, that only communication modules will be installed.

The Basic module means that the device is extrinsic. The sequence in loop is scan - localization - pick and place.

The Change Solution example includes the sequence for switching solutions in the visual controller.

The Multiple Vision System example demonstrates how to use more than one device.

The HandEye Multiview example can be used when the device is mounted on the effector.

What is installed automatically?

- Program and system modules according to the selected product (Bin picking studio / Locator studio)
- Sys.cfg file with automatic load modules or background task
- Copy examples to the Home folder of the robot controller according to the selected product
- Photoneo.coblox file for Wizard and modules with source code according to the product (only for Omnicore)

Product dependencies

- **Options Requirements:**
  The license for **IRC5** must include the **PC-Interface** option for the **Locator application**. For **Bin Picking** application, the license must include both **PC-Interface** and **Multitasking** options.
- The license for **Omnicore** must include the **Multitasking** options for the **Bin Picking** application.
  For the **Locator** application **does not necessary any special license**.
- RobotWare must be version 6.08.2003 or higher for IRC5 control.
- RobotWare must be version 7.14.3 or higher for Omnicore control.
- Installing the IRC5 version on Omnicore and vice versa is impossible.

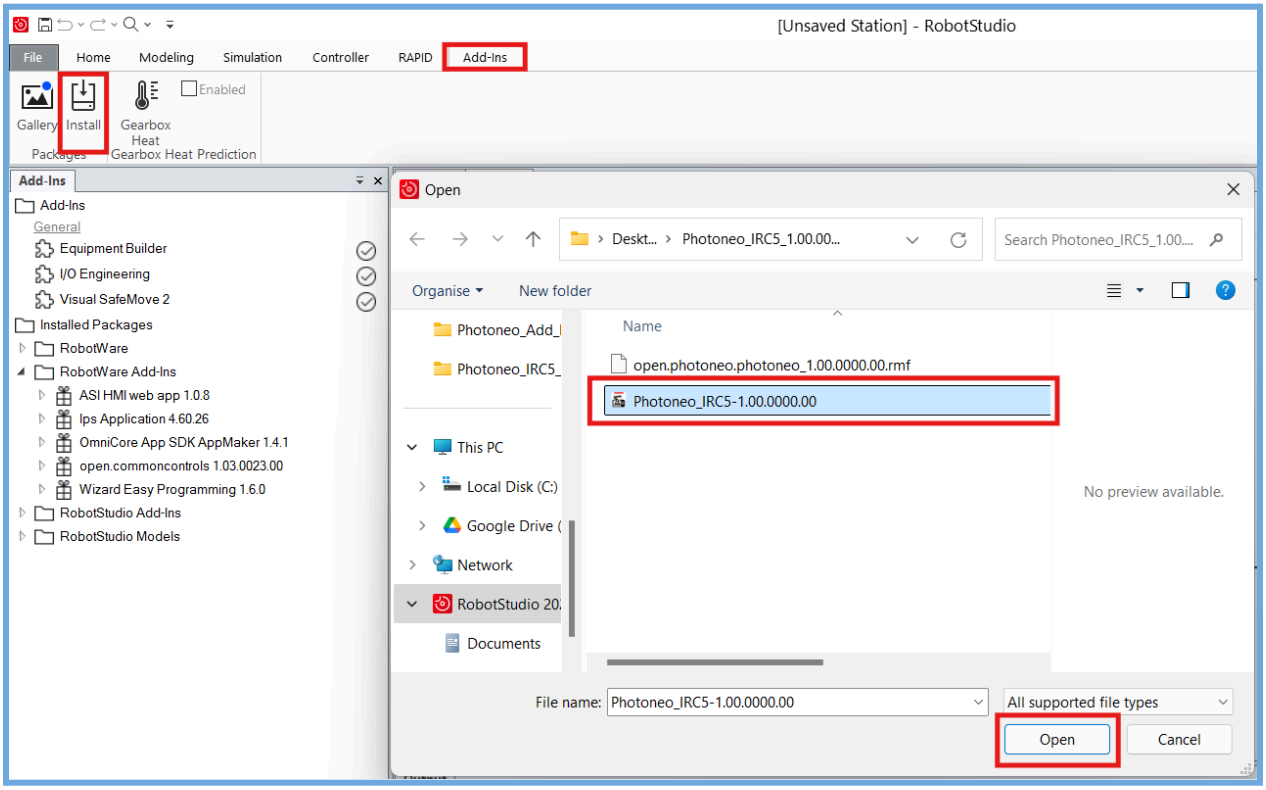
How to install Photoneo Add-In

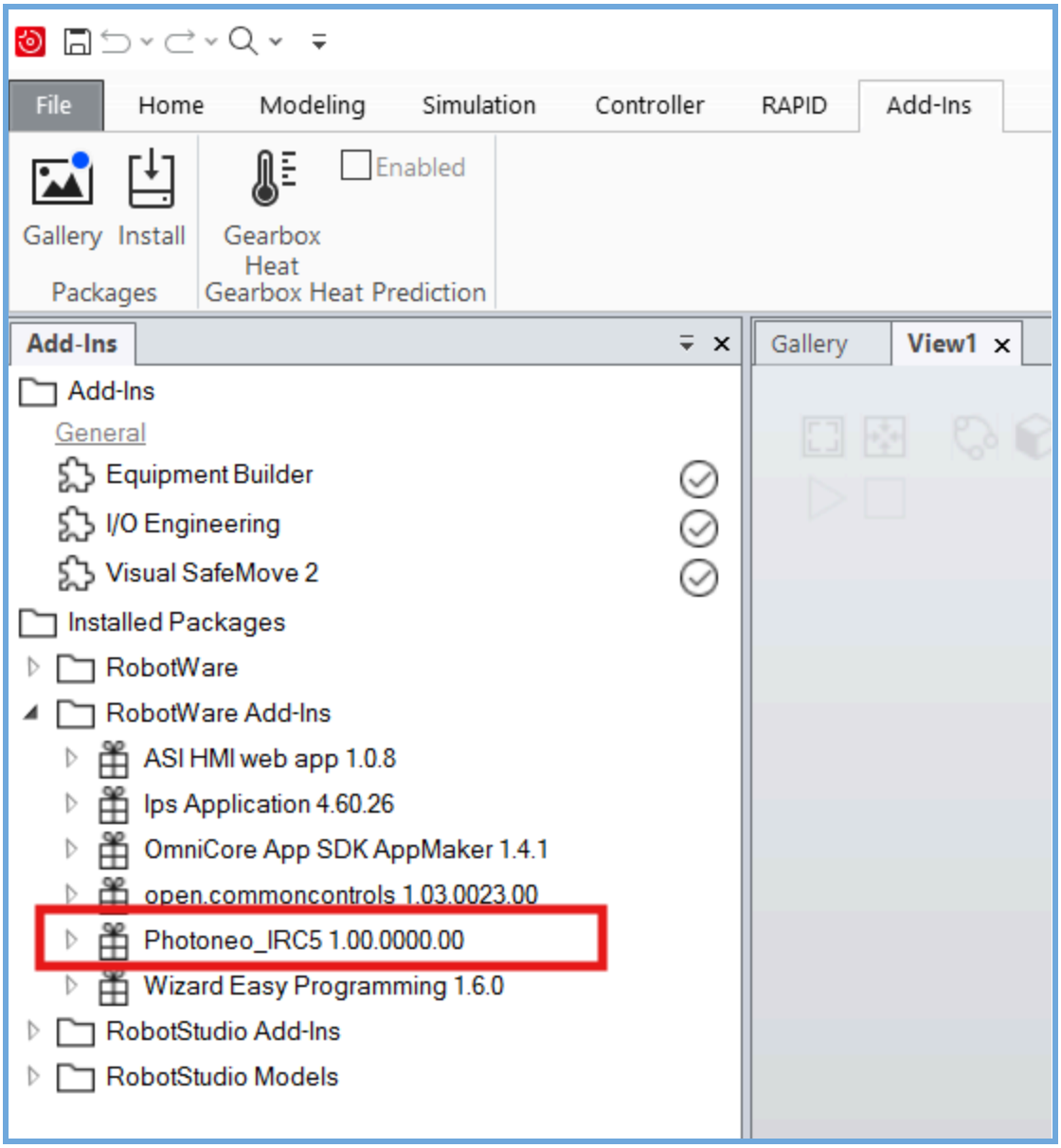
1. Download the Required Application

Download either **Photoneo_IRC5** or **Photoneo_Omnicore** from the website**.**

2. Install the Add-in in RobotStudio

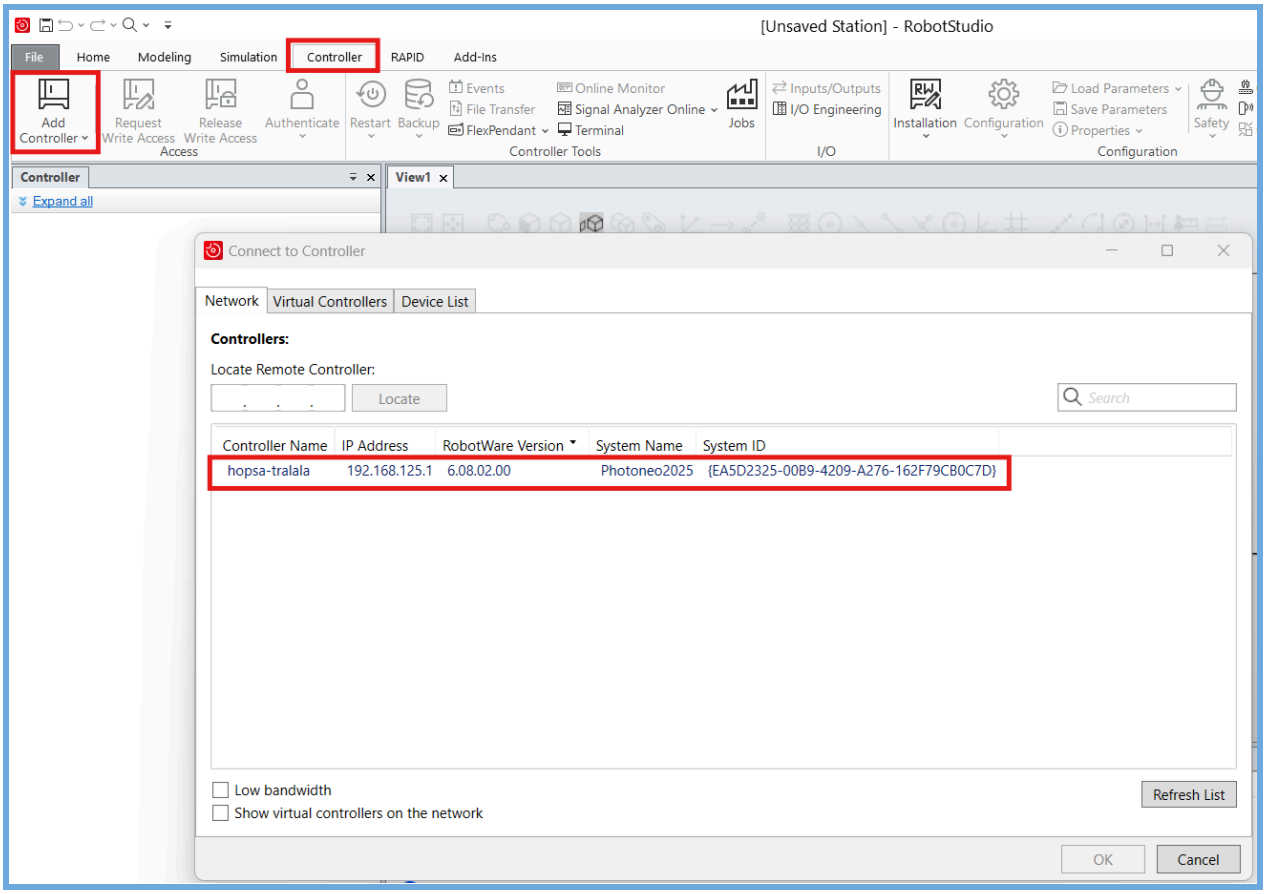
1. Open **RobotStudio**.
2. Navigate to the **Add-Ins** window.
3. Click the **Install** button.
4. Select **Photoneo_IRC5** or **Photoneo_Omnicore**.
5. Click **Open** to install the Add-in.
6. The Add-in will be automatically installed in RobotStudio.

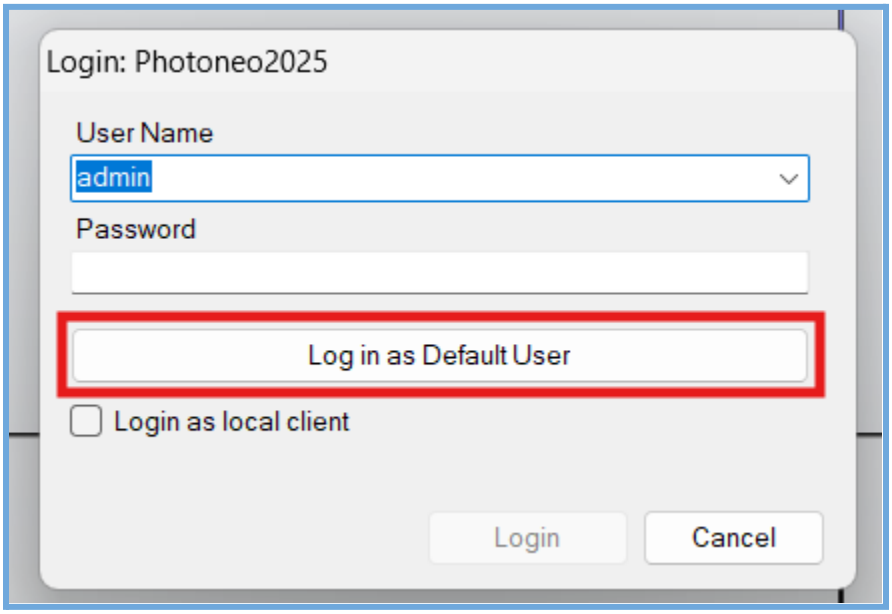




3. Create a New System or Add the Add-in to an Existing System

1. Open the **Controller** window.
2. Click **Add Controller**.
3. Double-click your controller in the list.
4. Log in as the **Default User** (or use your own account).

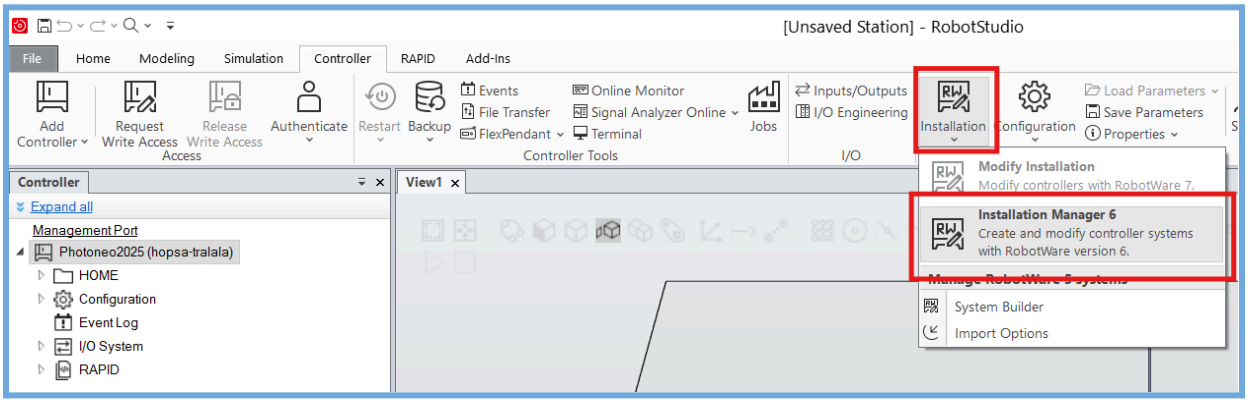

[Unsaved Station] - RobotStudio
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Add Controller
Request Write Access
Release Write Access
Authenticate
Restart
Backup
Events
File Transfer
FlexPendant
Online Monitor
Signal Analyzer Online
Terminal
Jobs
Inputs/Outputs
I/O Engineering
Installation
Configuration
Load Parameters
Save Parameters
Properties
Safety
Access
Controller Tools
I/O
Configuration
Controller
Expand all
View1
Connect to Controller
Network
Virtual Controllers
Device List
Controllers:
Locate Remote Controller:
Locate
Search
Controller Name
IP Address
RobotWare Version
System Name
System ID
hopsa-tralala
192.168.125.1
6.08.02.00
Photoneo2025
{EA5D2325-00B9-4209-A276-162F79CB0C7D}
Low bandwidth
Show virtual controllers on the network
Refresh List
OK
Cancel


Login: Photoneo2025
User Name
admin
Password
Log in as Default User
Login as local client
Login
Cancel

The next steps will be different for IRC5 and Omnicore.

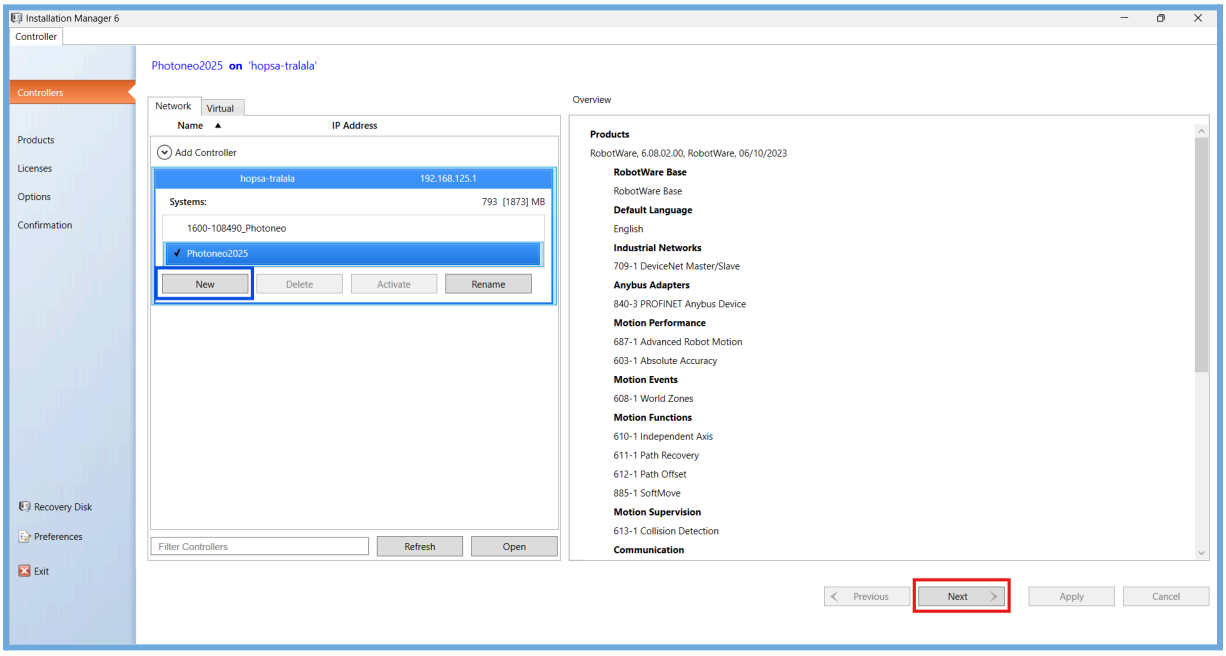
IRC5

5. In the **Controller** window, click **Installation**, then select **Installation Manager 6**.



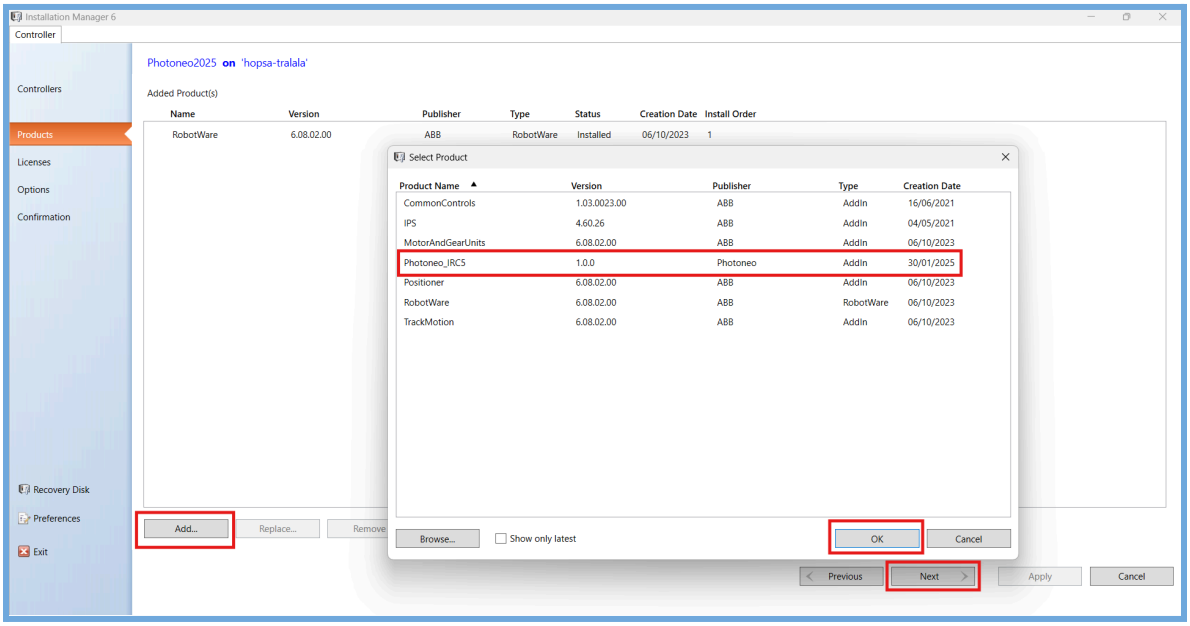
6. In **Installation Manager**, you can:

- Create a new system (blue option).
- Select an existing system, click **Next**, and modify it (red option).



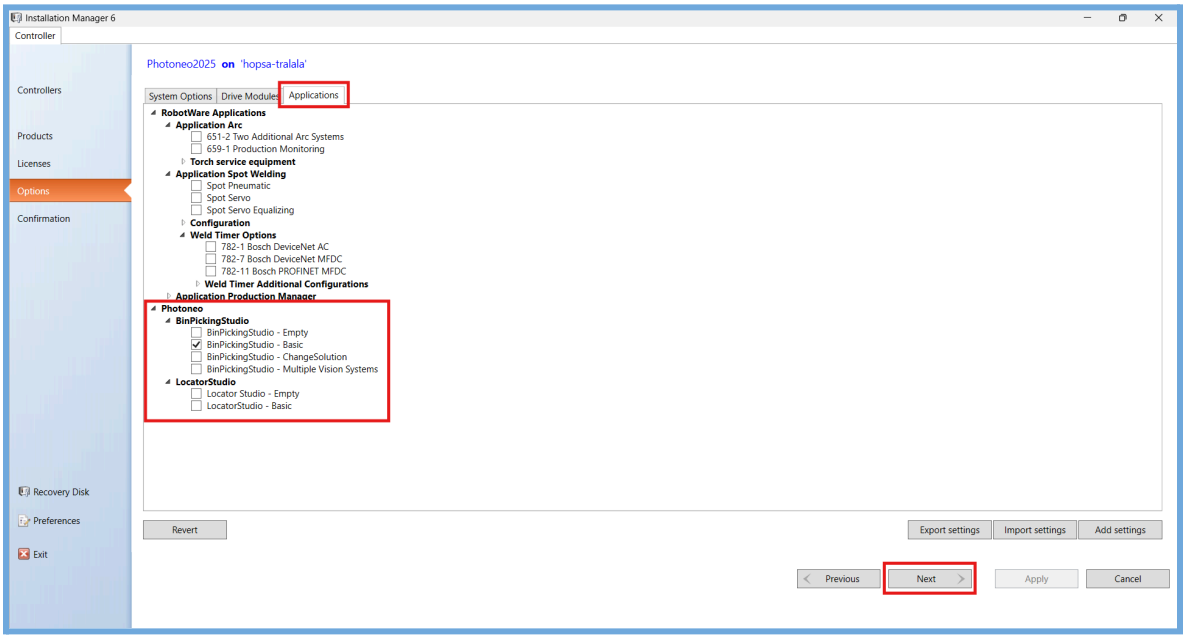
7. In the next step, add the **Photoneo_IRC5 Add-in**:

- Click **Add**.
- Select **Photoneo_IRC5** from the installed add-ins.
- Confirm with **OK**, then click **Next**.
- Skip the **Licenses** window by clicking **Next**.



8. The **Photoneo_IRC5 Add-in** will now appear in the **Application** section.

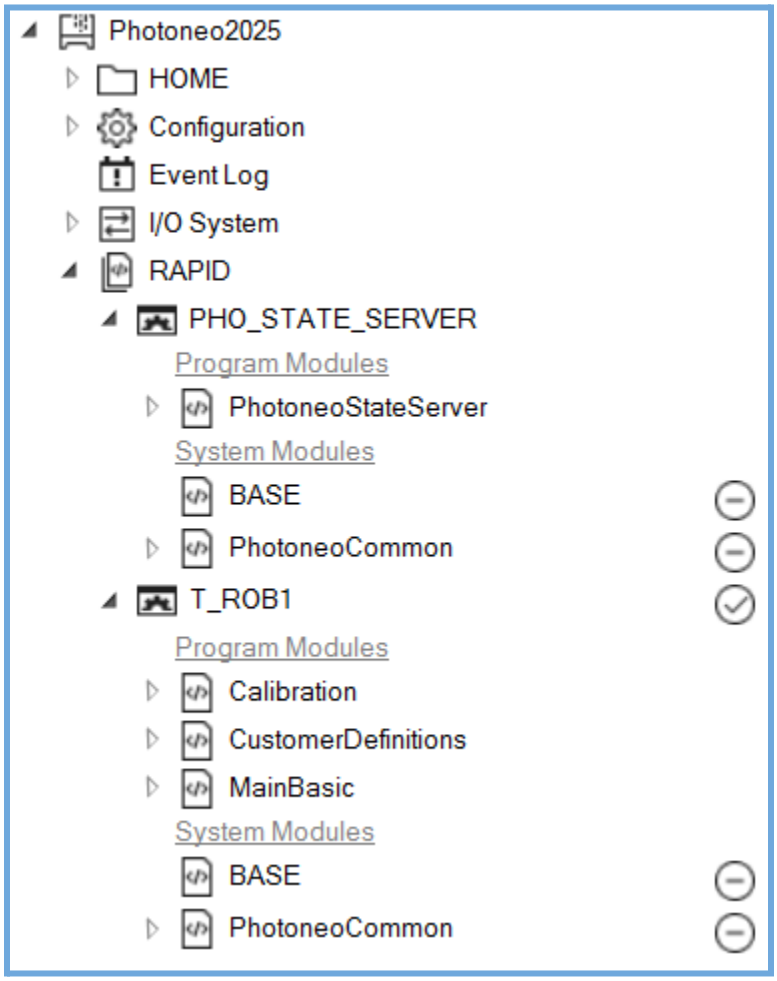
- Choose the example you want to install.



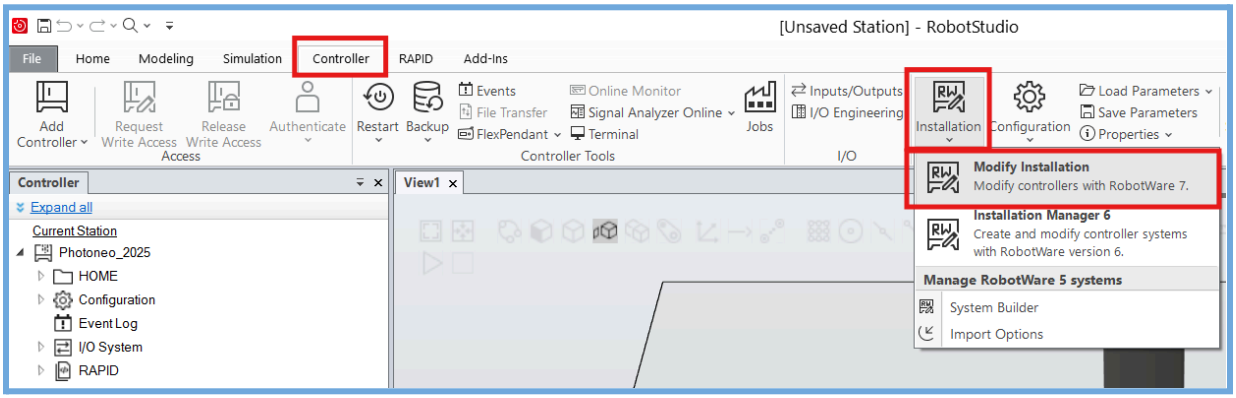
9. In the final step, review your selections and click **Apply**.

- Ensure you have **write access** on the FlexPendant.
- When applying a new configuration the controller RAPID program and configuration data will be deleted and reset to the factory default settings. Create a backup first if you want to save your RAPID program and data.
- The installation process may take a few minutes.

The installed system will depend on your selected product. The final setup might resemble the example shown below (**BinPickingStudio - Basic**).

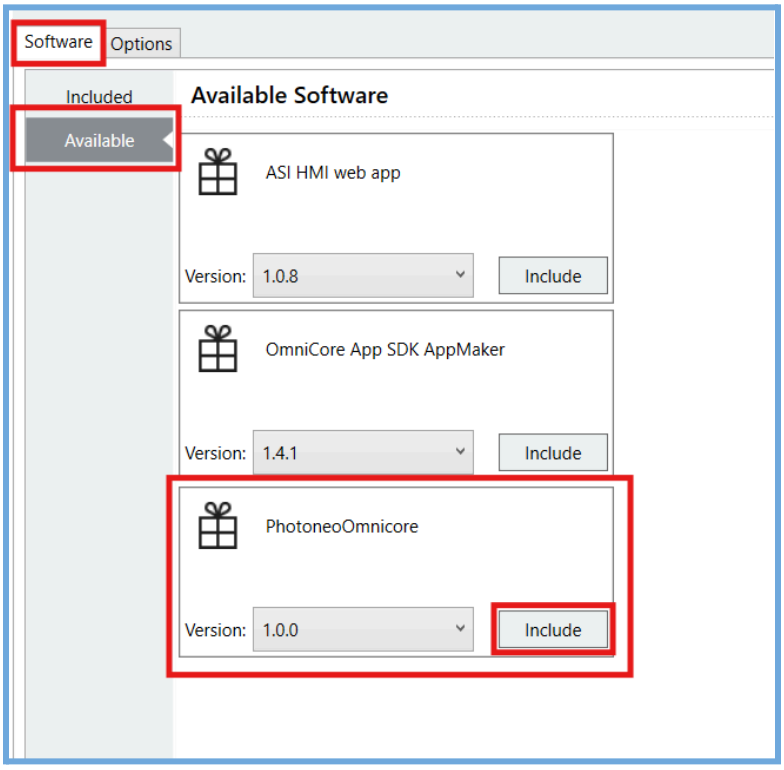


5. In the **Controller** window, click **Installation**, then select **Installation Manager 6**.

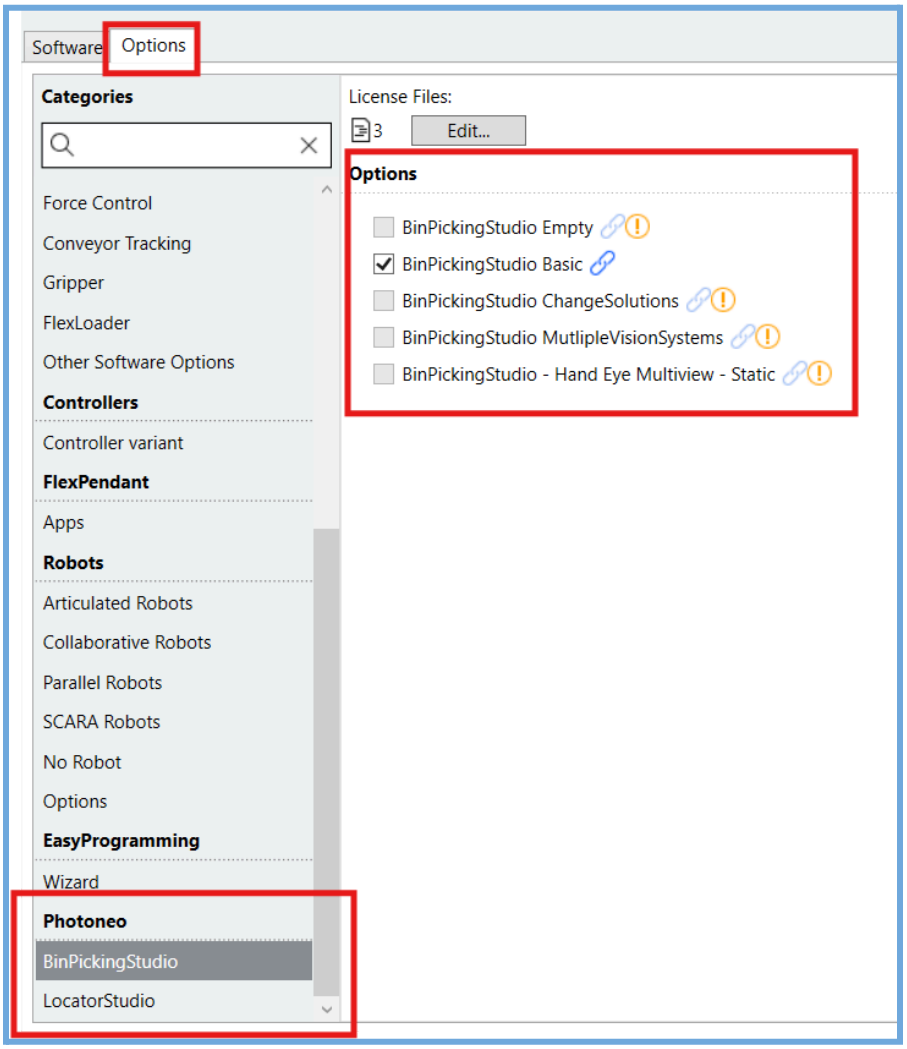


6. In **Modify Installation** you can adjust your Omnicore system

- In windows **Software** click **Available**.
- Select **PhotoneoOmnicore** from the installed add-ins.
- Confirm with **Include**.

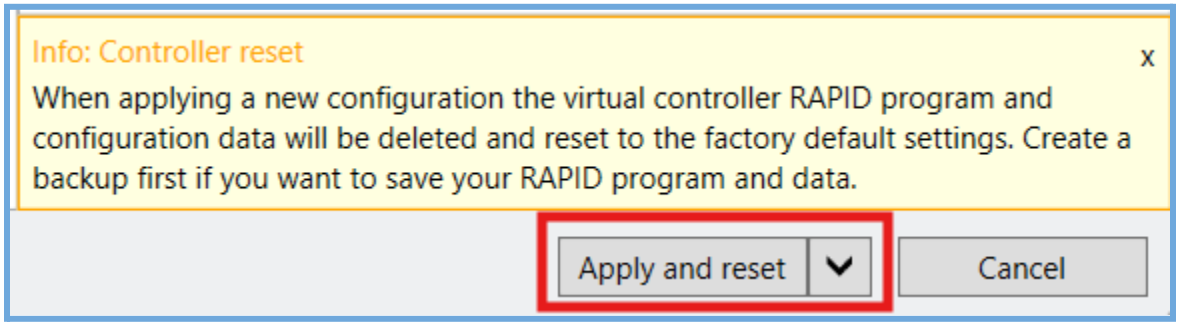


7. In the **Options** window, scroll to the bottom and select a product - **Bin Picking Studio** or **Locator Studio**.
    - In the middle section of the window, choose a module example that best fits your application.

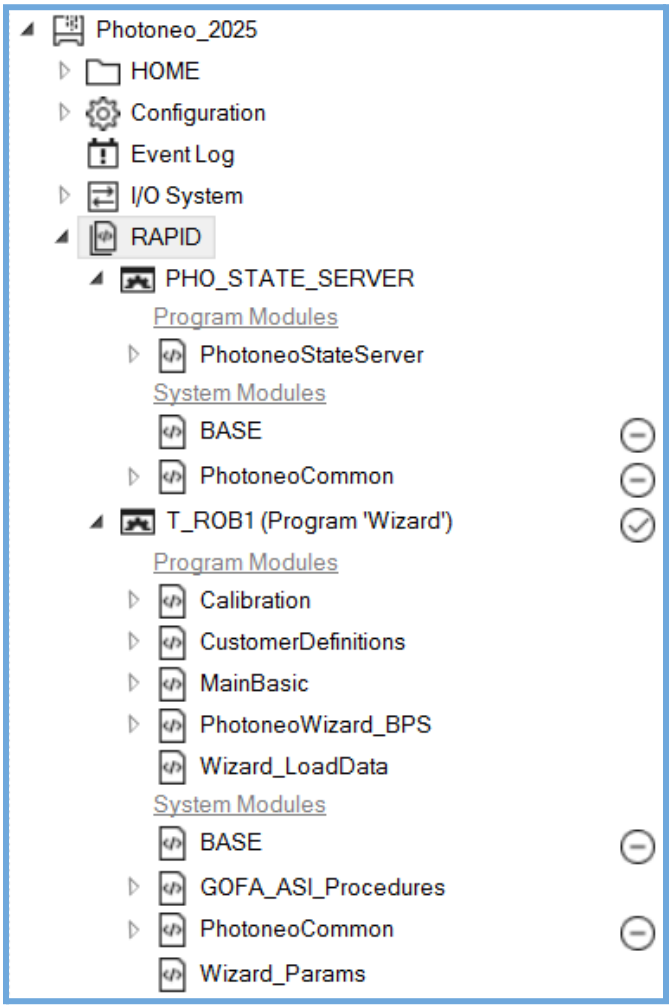


8. In the final step, review your selections and click **Apply and reset**.

- Ensure you have **write access** on the FlexPendant.
- When applying a new configuration the controller RAPID program and configuration data will be deleted and reset to the factory default settings. Create a backup first if you want to save your RAPID program and data.
- The installation process may take a few minutes.

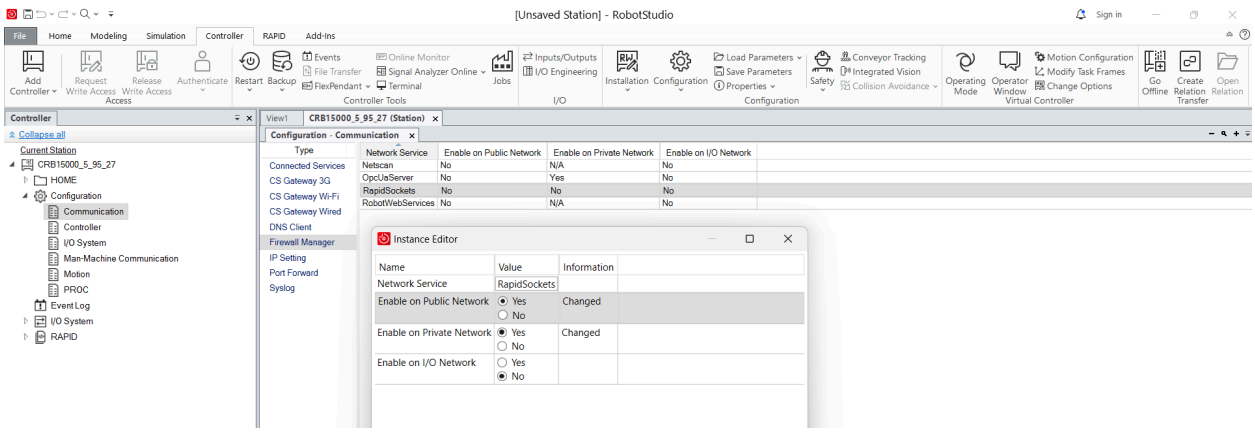


The installed system will depend on your selected product. The final setup might resemble the example shown below (**BinPickingStudio - Basic**).



9. THE FIREWALL

The firewall for **RapidSockets** needs to be enabled. In your **RobotStudio** select **Configuration -> Communication -> Firewall Manager -> RapidSockets**. Enable the Firewall for RapidSockets on both the **Public** and **Private** Network:



Initial setup before the first test

Before running your first **Bin Picking** application test, you need to configure the following:

- **Tool Data (Photool):**
  - Adapt **the mass** and **center of gravity** of the tool based on your application.
- **IP Addresses:**
  - Adjust the **robot controller's IP address**.
  - Update the **VC IP address** in the robot module to match the settings in the vision controller.
- **Robot Positions:**
  - Modify the ***Home Position, Start Bin Picking Position,*** and **End Bin Picking Position** as needed.
  - Adjust the **placing positions** to fit your application.
- **Gripper Control:**
  - Add gripper control commands to the routines **Gripper_attach** and **Gripper_detach** in **CustomerDefinition.mod**.
- **Firewall Manager (Omnicore):**
  - Check → Configuration → Communication → Firewall Manager → RapidSockets → for Public Network and Private Network → **YES**
- **Calibration:**
  - Adjust the **9 calibration positions** and perform the calibration process.

For more details, refer to the BPS documentation.

Before running your first **Locator** application test, you need to configure the following:

- **Tool Data (Photool):**
  - Adapt **mass** and **center of gravity** of the tool based on your application.
- **IP Addresses:**
  - Adjust the **robot controller's IP address**.
  - Update the **VC IP address** in the robot module to match the settings in the vision controller.
- **Robot Positions:**
  - Adjust the **placing positions** to fit your application.
- **Firewall Manager (Omnicore):**
  - Check → Configuration → Communication → Firewall Manager → RapidSockets → for Public Network and Private Network → **YES**
- **Calibration:**
  - Adjust the **9 calibration positions** and perform the calibration process.

For more details, refer to the LS documentation.

Wizard (Omnicore)

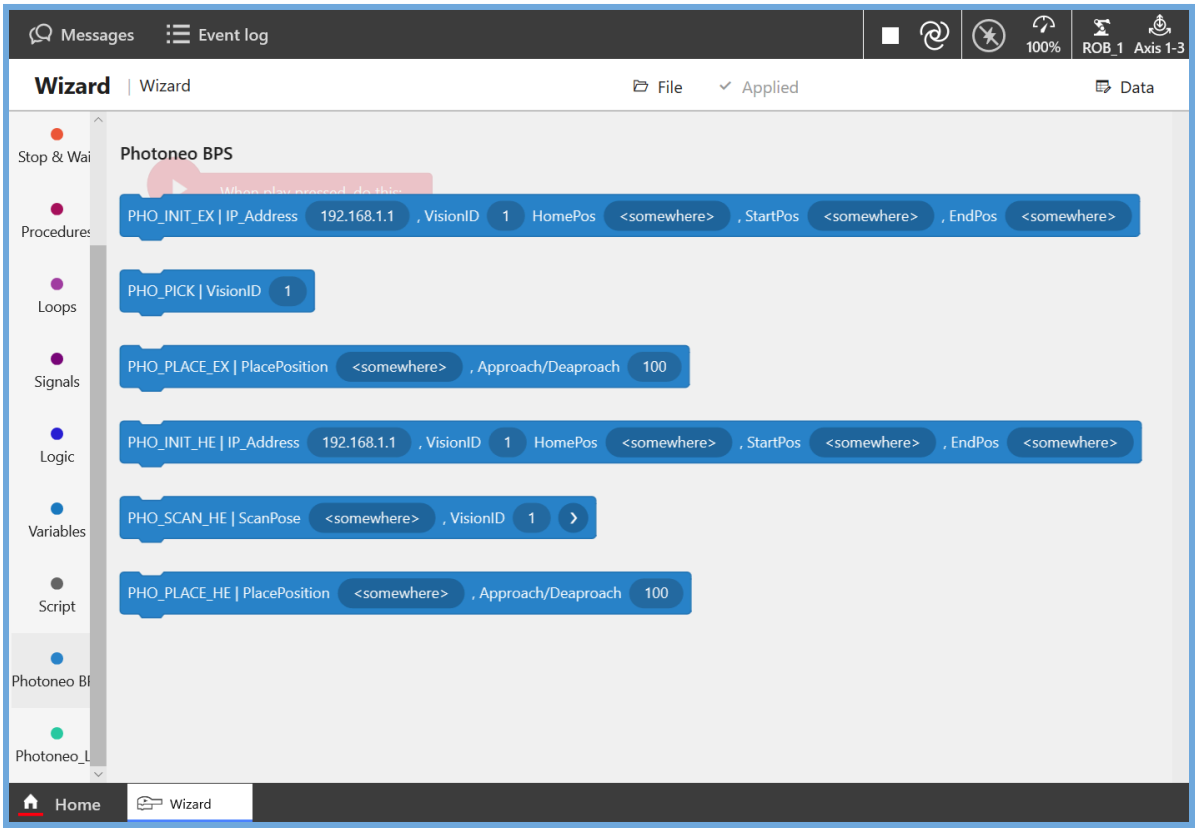
Wizard Easy Programming is an intuitive graphical tool designed to simplify robot programming for both beginners and experienced users. It enables effortless, fast, and efficient programming of collaborative and industrial robots across various applications. Users can easily create programs by dragging and dropping predefined blocks or selecting them with a button press. Once the setup is complete, simply press play to execute the program—making automation more accessible than ever.

Now you can quickly create applications for bin picking or part localization with ease.For both products, we have pre-designed blocks that simply need to be connected in the correct sequence, parameters filled in, and the application is ready to run.

The source code is located in the **PhotoneoWizard_BPS.mod** or **PhotoneoWizard_LS.mod** module. It is recommended to review the code and modify it to suit your needs.

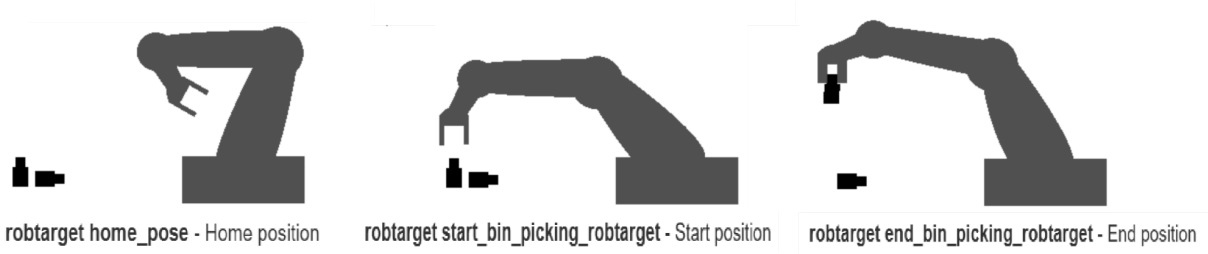
Bin Picking blocs

EX means that these blocs are pre-designed for Extrinsic applications.
HE means that these blocs are pre-designed for HandEye application.
The block without a suffix is universal and can be used in both cases.



PHO_INIT_EX

- **IP_Address** → IP_Address of the Visual controller, same as the settings in the vision controller
- **VisionID** → The ID of the vision system in the currently deployed solution in Studio.
- **HomePos** → The robot's home position, located away from the scanning area. The robot moves to this position during initialization, if the system does not detect any objects, or in case of an error.
- **StartPos** → The starting position from which the system begins executing the planned path.
- **EndPos** → The final position where the robot completes path planning after picking up the part.



robtarget home_pose - Home position **robtarget start_bin_picking_robtarget** - Start position **robtarget end_bin_picking_robtarget** - End position

PHO_PICK

-
- **VisionID** → The ID of the vision system in the currently deployed solution in Studio. This block performs scanning and localization based on the VisionID in the active solution within Studio.

PHO_PLACE_EX

- **PlacePosition** → The position where the robot places the picked-up part.
- **Approach/Deapproach** → The distance along the Z-axis, in millimeters, within the robot's base coordinate system that the robot reaches before or after the **Place Position**.

PHO_INIT_HE

- **IP_Address** → IP_Address of the Visual controller, same as the settings in the vision controller
- **VisionID** → The ID of the vision system in the currently deployed solution in Studio.

- **HomePos** → The robot's home position, located away from the scanning area. The robot moves to this position during initialization, if the system does not detect any objects, or in case of an error.
- **StartPos** → The starting position from which the system begins executing the planned path.
- **EndPos** → The final position where the robot completes path planning after picking up the part.

PHO_SCAN_HE

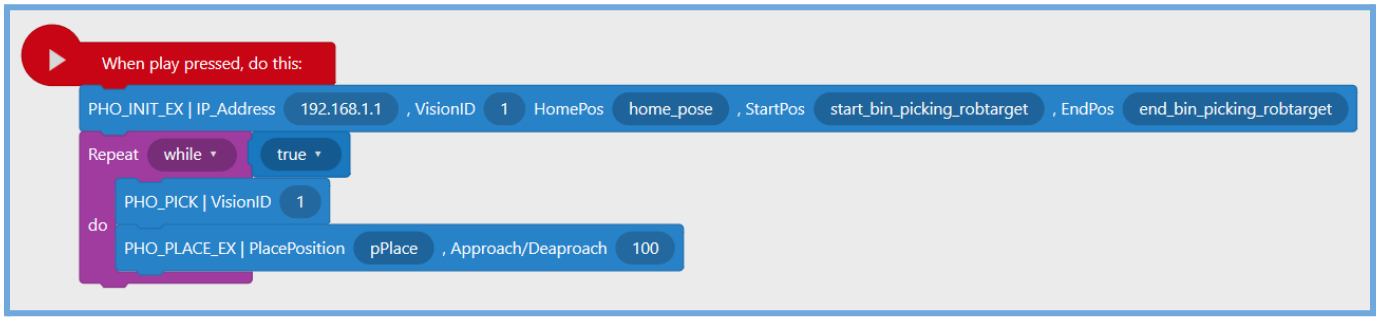
- **ScanPose** → The robot position where the user wants to perform scanning.
- **VisionID** → The ID of the vision system in the currently deployed solution in Studio.
- **WaitTime** → An optional argument that introduces a delay between the robot stopping at the target and scanning. This helps filter out shaking that may occur if the robot's end effector is not rigid enough.

PHO_PLACE_HE

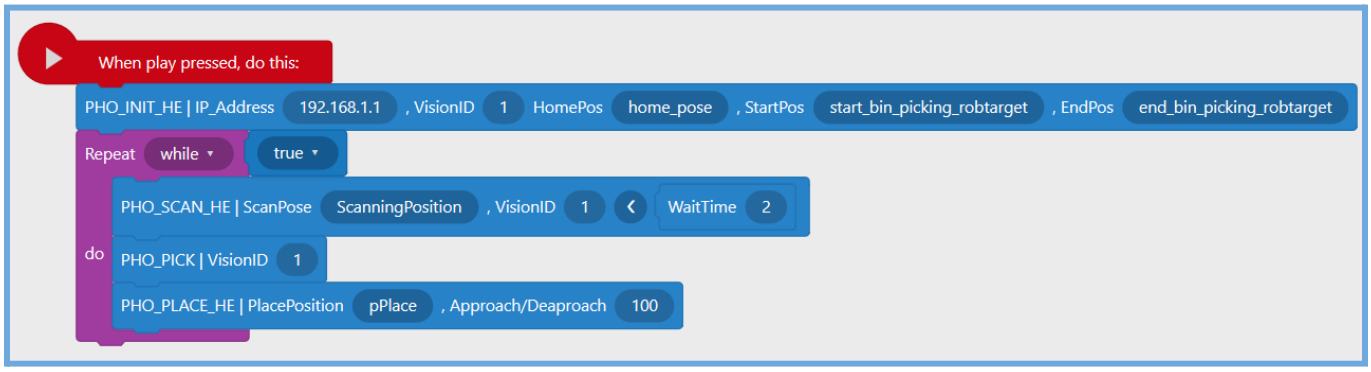
- **PlacePosition** → The position where the robot places the picked-up part.
- **Approach/Deapproach** → The distance along the Z-axis, in millimeters, within the robot's base coordinate system that the robot reaches before or after the **Place Position**.

The difference between **PHO_PLACE_HE** / **PHO_PLACE_EX** and **PHO_INIT_EX** / **PHO_INIT_HE** is that in an **Extrinsic** application, scanning can be performed because the robot is likely outside the scanning volume, reducing cycle time. In a **Hand-Eye** application, however, the robot must first reach the scanning position before scanning.)

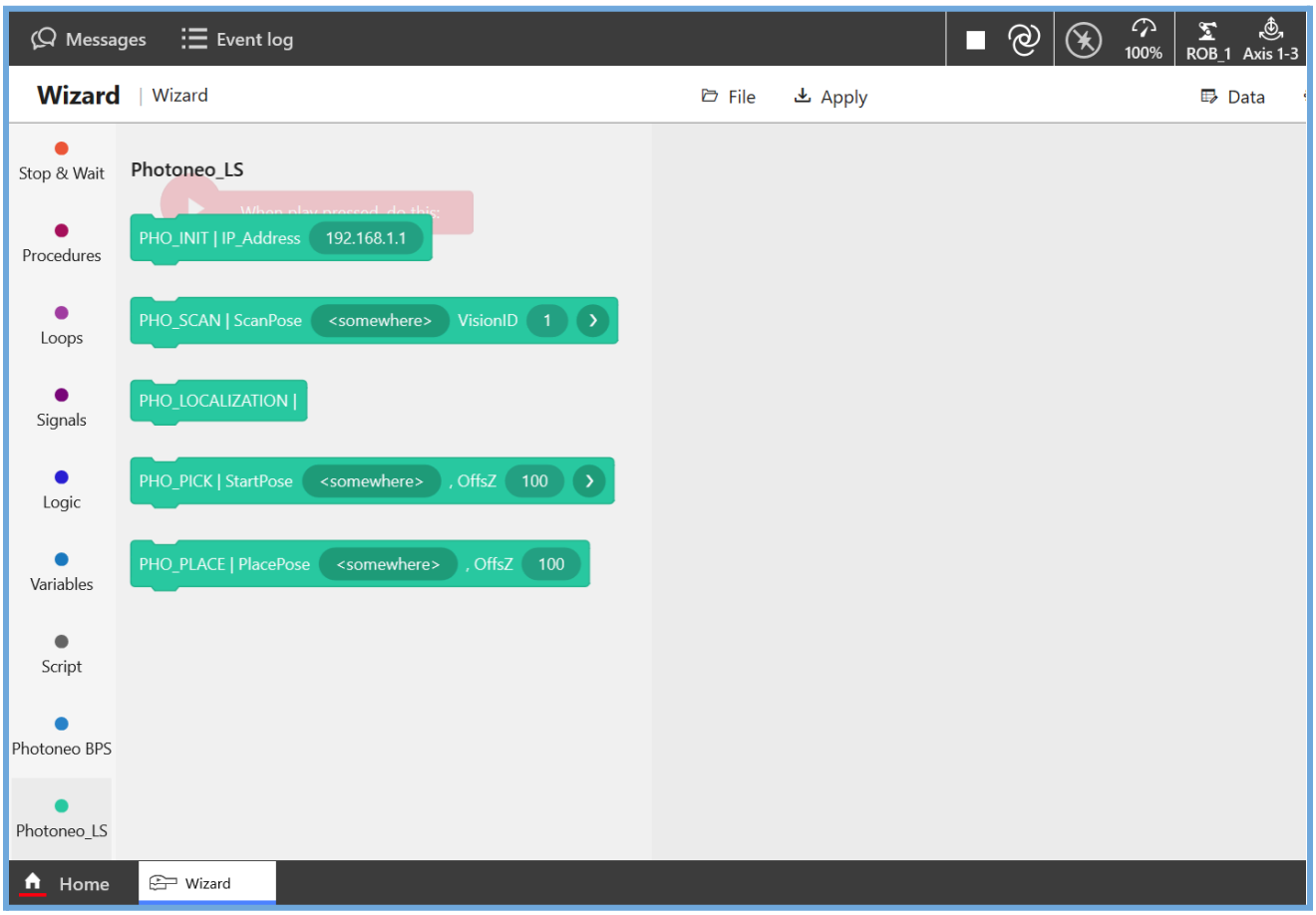
Wizard Bin picking Extrinsic example:



Wizard Bin picking Hand Eye example:



Locator blocs



PHO_INIT

- **IP_Address** → IP_Address of the Visual controller, same as the settings in the vision controller

PHO_SCAN

- **ScanPose** → The robot position where the user wants to perform scanning.
- **VisionID** → The ID of the vision system in the currently deployed solution in Studio.
- **WaitTime** → An optional argument that introduces a delay between the robot stopping at the target and scanning. This helps filter out shaking that may occur if the robot's end effector is not rigid enough.

PHO_LOCALIZATION

- It contains a sequence of requests for starting localization in **Locator Studio**.

PHO_PICK

- **StartPose** → The position from which the robot moves to the position reported by Locator Studio**.** It is typically located above the area where parts are being localized.
- **OffsZ** → The distance along the Z-axis, in millimeters, that the robot reaches before and after the Pick Position (Approach/Deapproach)
- **Wobj | Tool** → An optional argument that defines whether the offset in the Z-axis will be applied in the robot's base coordinate system or the tool coordinate system. For the tool, the value may be positive or negative depending on the orientation of the TCP. If the user does not enable either option, the robot will perform the movement in the base coordinate system by default.

PHO_PLACE

- **PlacePosition** → The position where the robot places the picked-up part.
- **OffsZ** → The distance along the Z-axis, in millimeters, within the robot's base coordinate system that the robot reaches before or after the **Place Position**.

The **PhotoneoWizard_LS** module contains the routines **GRIPPER_ATTACH** and **GRIPPER_DETACH**, where commands for controlling the end effector need to be added.

Wizard Locator example: