Kawasaki Robotics

astorinoIDE

User manual 2024

Introduction

This manual describes the operation of astorinoIDE software used for programming and operating astorino robots. It does not describe the operation of individual functions and the behavior of the robot. These items are described in the astorino user manual.

This manual is valid from firmware version 3.8.1 and astorinoIDE version 0.95

ASTORINO is an educational robot that has been developed specifically for training facilities and institutions. Students can use ASTORINO to learn the automation and robotization of industrial processes in practice.

1. The "astorinoIDE" software included with Astorino is licensed solely for use with this robot and may not be used, copied or distributed in any other environment.
2. ASTOR and Kawasaki Robotics are not liable for accidents, damages and/or problems caused by improper use of the Astorino robot.
3. ASTOR and Kawasaki Robotics reserve the right to change, amend or update this manual without prior notice.
4. This manual may not be reprinted or copied in whole or in part without the prior written permission of ASTOR and Kawasaki Robotics.
5. Keep this manual safe and within easy reach so you can use it at any time. If the manual is lost or severely damaged, contact ASTOR.

Symbols

Elements that require special attention in this manual are indicated by the following symbols.

Ensure proper operation of the robot and prevent injury or damage to property by following the safety instructions in the boxes with these symbols.

⚠ WARNING

Failure to follow the instructions below may result in injury.

[**NOTE**]

Specifies precautions for robot specifications, operation, teaching, and maintenance.

⚠ WARNING

1. **The accuracy and effectiveness of the graphs, procedures and explanations contained in this manual cannot be confirmed with absolute certainty. If you experience any problems, please contact Kawasaki Robotics GmbH or Astor at the above address.**

2. **To make sure that all work is done safely, read the instructions with understanding. In addition, you should review all applicable laws, regulations and related materials, as well as the safety statements described in each section. Have proper safety measures and procedures in place for actual work.**

Paraphrases

This guide uses the following spelling rules:

- For a specific press, the corresponding button is enclosed in angle brackets, such as <F1> or <Enter>.
- For a dialog box or toolbar button, the button name is enclosed in square brackets, such as [OK] or [Reset].
- The pick-up fields are marked with a square field. If they are activated,☐ there is also a small check mark ☑ inside the symbol .

ASTORINO User Manual

Table of contents

1 Naming in this manual

In this section you will find definitions of terms used in this manual.

The author of the textbook tries to use generally applicable terminology, maintaining the greatest possible logic. Unfortunately, it should be noted that the perception of the terminology used can vary depending on the point of view, even when considering the same topic. It should also be noted that over the course of the history of the development of robots, computers and software, terminology has developed in different ways. In the modern manual we will not find terminology that will always be 100% consistent with the opinions of all users and experts.

2 Description of the ASTORINO robot

The ASTORINO is a 6-axis learning robot developed specifically for educational institutions such as schools and universities. The robot design is based to be 3D printed with PET-G filament. Damaged parts can be reproduced by the user using a compatible 3D printer.

Programming and control of the robot is done by the "astorino" software.

The latest software version and 3D files can be downloaded from the KAWASAKI ROBOTICS FTP server:

https://ftp.kawasakirobot.de/Software/Astorino/

Just like Kawasaki’s industrial Robots the ASTORINO is programmed using AS language. Providing transferable programing skills from the classroom to real industrial applications

3 Safety notes

[**NOTE**]

Always take care of the personal safety of users and others when operating a robot arm or starting a robot cell!

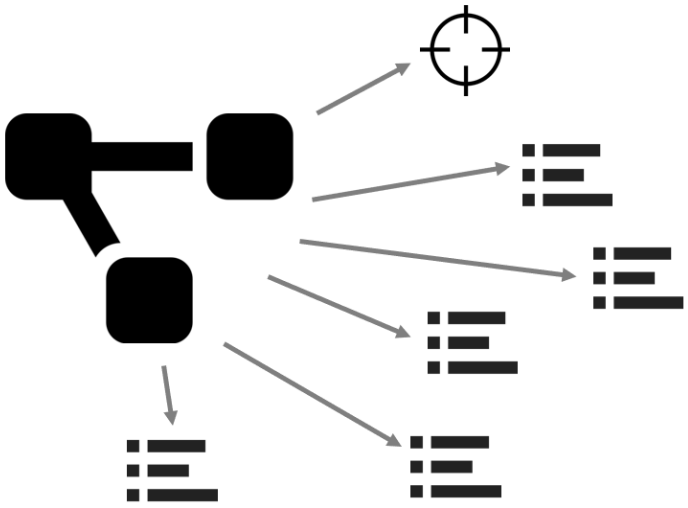
- In the basic version, the robot does not have elements related to the safety of the robotic station. Depending on the app, you may need to add them. The basic version of the robot is equipped with an emergency button.
- CE marking: The robot arm must be subject to a risk assessment when working in production applications and must comply with applicable safety regulations to ensure personal safety. Depending on the outcome of the assessment, further safety components should be integrated. These are usually safety relays and door switches. The system boot engineer is responsible. Education apps don't require additional security features.
- The robot controller includes a 24 V power supply, which itself requires mains voltage (100/240 V). Check the label on the power supply. Only qualified personnel can connect the power supply to the network and start it.
- Work on the robot's electronics should be carried out only by qualified personnel. Check the current electrostatic discharge (ESD) guidelines.
- Always disconnect the robot from the power supply (100/240 V) while working in the robot base (controller) or any electronics connected to the robot controller.
- DO NOT connect hot! This may cause permanent damage to the engine modules. Do not install or remove any plug/disconnect modules or connectors (e.g. emergency stop button, DIO modules, engine connectors) while the power is on.
- The robot arm must be positioned on a stable surface and screwed or secured in some other way.
- Use and store the robot only in a dry and clean environment.
- Use the system only at room temperature (15° to 32°C) - recommended.

4 AstorinoIDE Software

4.1 Basic information

The astorinoIDE software is an astorino robot programming environment designed for advanced robot users. astorinoIDE, unlike the classic astorino environment, is based on projects that are also saved on the user's computer in the Documents folder.

A project is a collection of programs and saved points for a given application.



This approach allows you to create many different applications on the robot without having to delete or overwrite already written programs or points.

There can always be only one project in the robot's memory, while many projects can be saved on the computer.

5 System Requirements

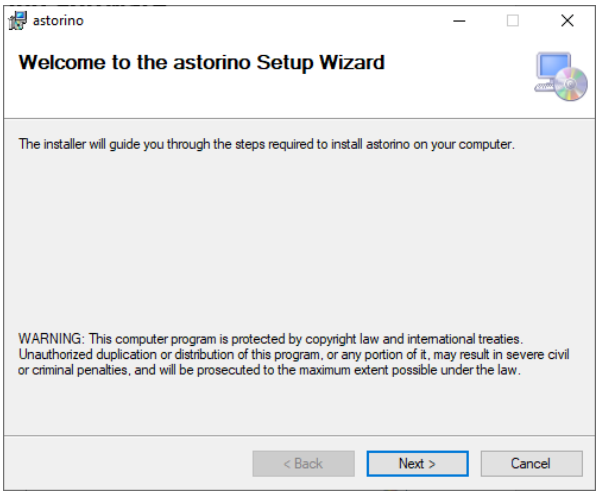
Before installing astorino software, ensure that the computer meets the following hardware and software requirements.

Part	Requirements
CPU	2.0 Ghz or faster processor
Memory	4 GB minimum
Disk	100 MB free space
Graphics card	Any
Display settings	1280 x 720 pixels minimum resolution, 100 % display scaling recommended
Mouse	Three-button mouse

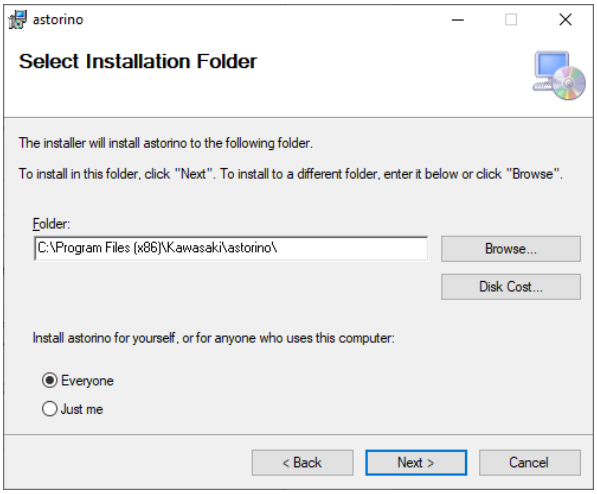
System	Version
Windows	7, 8, 8.1, 10, 11

6 Installing astorinoIDE software

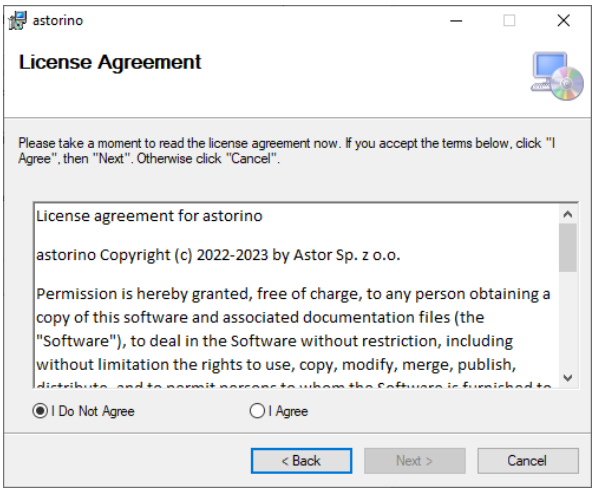
Run astorinoIDE_x.x.x.exe



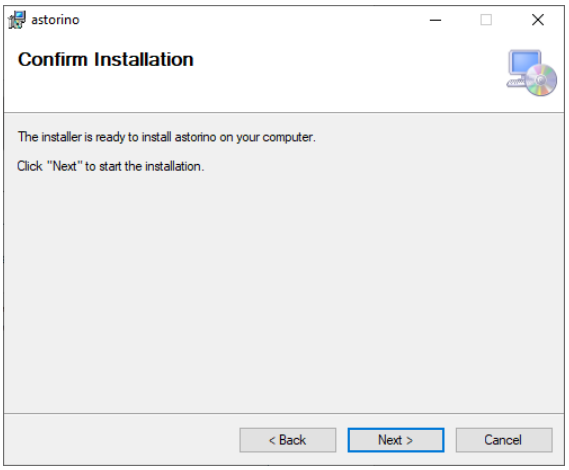
Confirm or customize the installation directory



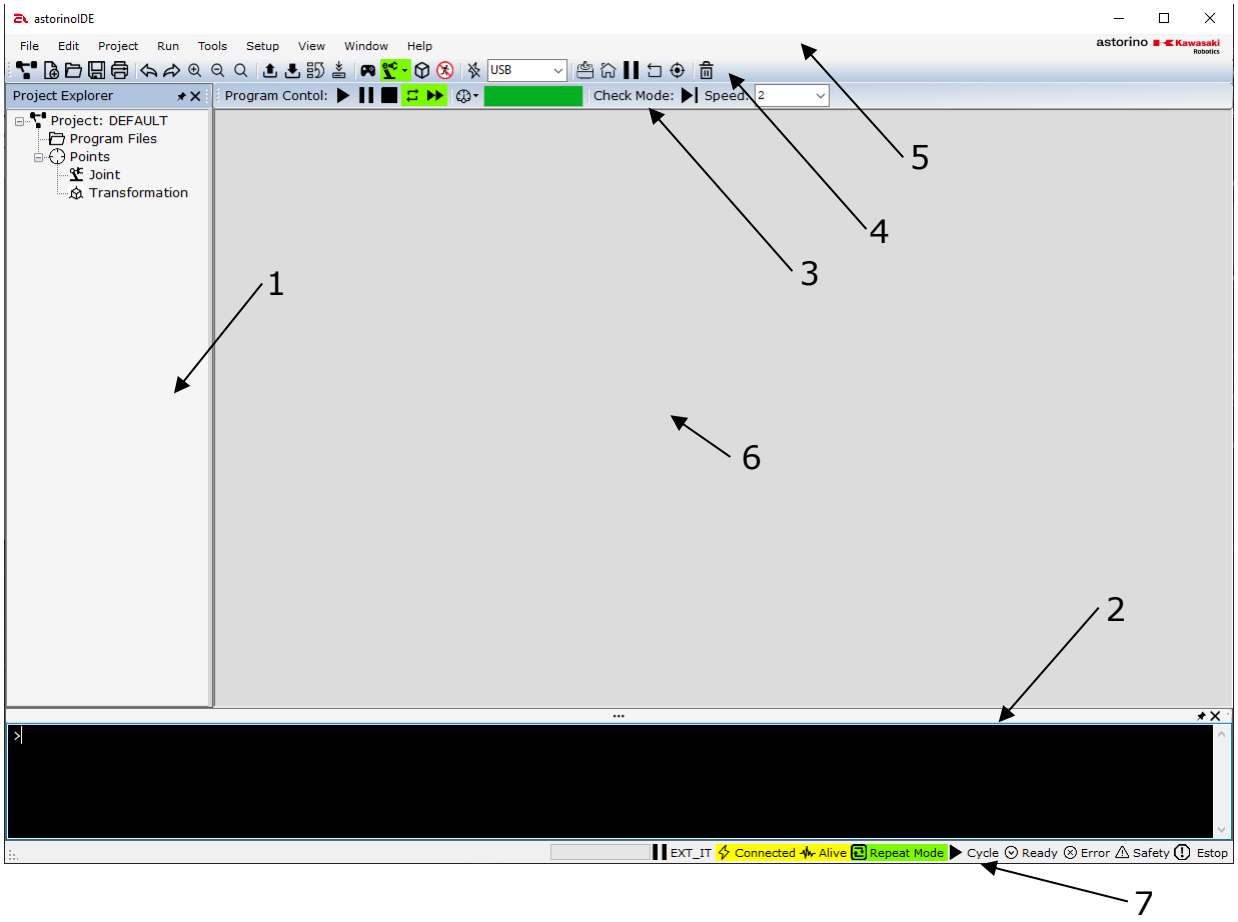
Accept license



Start installation

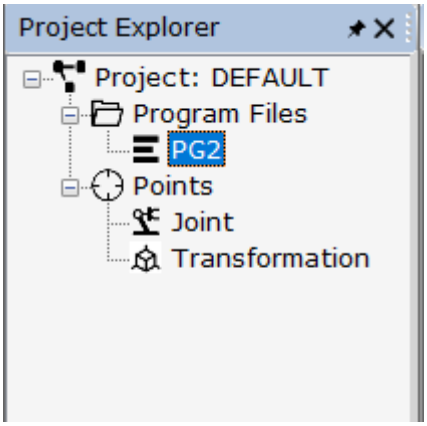


7 Main window

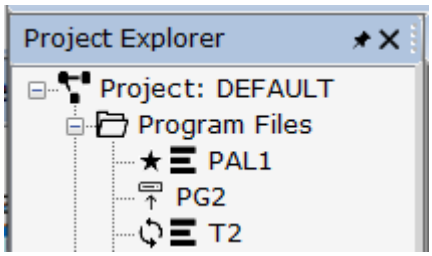


1. Project Explorer	Tree of the currently open project
2. Terminal window	Terminal for receiving and sending commands
3. Program control bar	Cycle on/off, speed change
4. Control bar	Project and robot management
5. Menu bar	Software Management
6. Main area	Robot control and program editing
7. Status bar	Current status of robot a and connections

7.1 Project Explorer



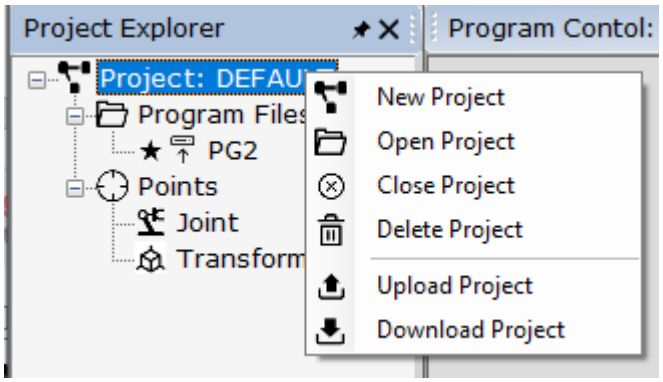
	Project name
	Folder of all created programs
	Lock/unlock project explorer width change
	Points catalogue
	Close the project explorer window
	Saved joints angles [JT1... 7]
	Saved Cartesian points [XYZ OAT JT7]
	Name of the program



	Program currently loaded into RAM and ready to run
	Bootloader, loaded into RAM when power on
	Program modified but not uploaded to robot memory

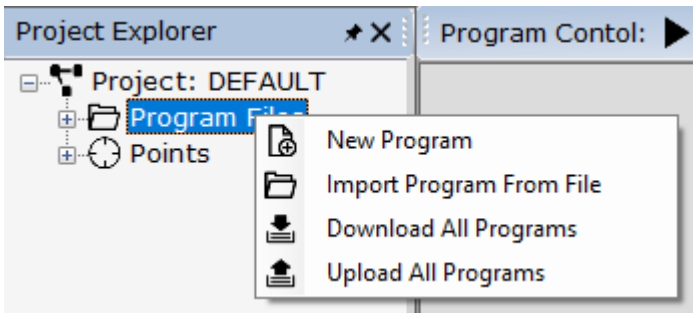
7.1.1 Context menu

- Right-clicking on the [Project:] field opens the context menu for managing the project.



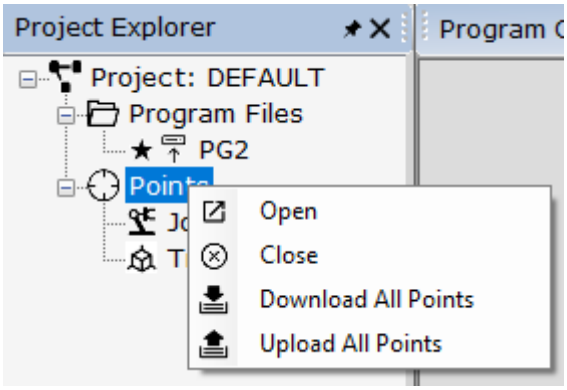
	Opens the window for creating a new project
	Opens the project opening window
	Closes and disables the current project
	Opens the delete projects window
	Uploads the entire project to the robot's memory
	Downloads the entire project from the robot's memory

- Right-clicking on the [Program Files] field opens the context menu for managing programs



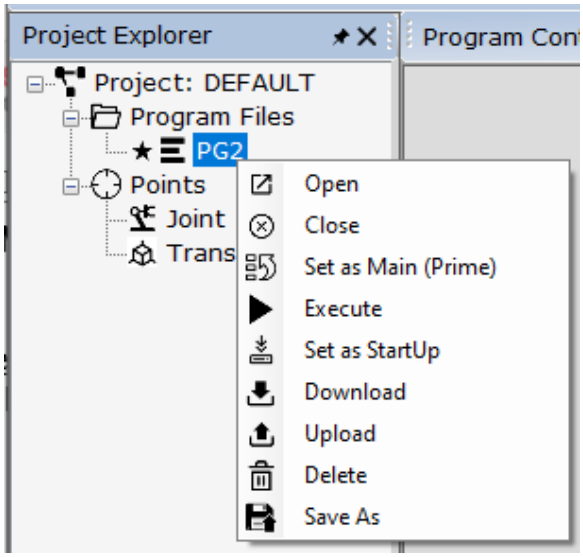
	Opens the window for creating a new program
	Imports the program from a * file. pg
	Rips all programs from the robot's memory
	Uploads all programs to the robot's memory

- Right-clicking on the [Points] field opens the context menu for managing points



	Opens the window for editing and viewing points
	Closes the edit and preview points window
	Downloads all points from the robot's memory
	Uploads all points to the robot's memory

- Right-clicking on the program name field opens a context menu that allows you to manage the program



	Opens the editing program window
	Closes the program window
	Loads the program into the robot's RAM and prepares it to run
	Loads the program into the robot's RAM and runs the program
	Sets the program as bootable, the boot program is loaded into the robot's RAM when the power is turned on
	Downloads the program from the robot's memory
	Uploads the program to the robot's memory
	Removes the program from the robot's memory and the project
	Saves the program to a separate file

7.2 Terminal window

The terminal is used to display information from the robot, but also to issue commands to the robot.

	Lock/Unlock Terminal Window Width Change
	Close a terminal window
	Change the height of a terminal window

All motion commands like LMOVE, HOME, etc. They must be preceded by the word "TO" and the robot must be READY and in REPEAT mode. For example, "DO LMOVE P1"

You can also use the terminal to read the values of variables (for example, "PRINT x"), learn points (for example, HERE P1), set variables (for example, x = 10), and so on.

Here is a list of Terminal commands:

CPUTEMP	Shows CPU temperature
FREE	Shows available RAM in %
ERESET	Resets the error
ZPOWER ON	Turns on ENGINES
ZPOWER OFF	Turns off ENGINES
HOLD	Pauses the currently running program
CONTINUE	Continues the paused program
ZZERO x	Starts resetting the specified axis - x

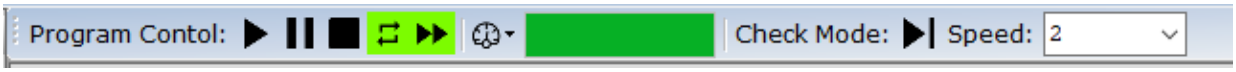
7.3 Menu bar

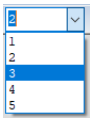
	Opens the window for creating a new project
	Opens the window for creating a new program
	Imports the program from a * file. pg
	Saves the active program to a file on your computer
	Prints the currently active program
	Reverts the last operation
	Renews the last operation
	Increases the size of text in the active program window
	Reduces the size of text in the active program window
	Resets the zoom in the active program window
	Uploads the program/project to the robot's memory
	Downloads the program/project from the robot's memory
	Loads the program into the robot's RAM and prepares it to run
	Sets the program as bootable, the boot program is loaded into the robot's RAM when the power is turned on
	Opens manual control window - RobotManager
	It allows you to change the robot operating mode. changes to Repeat mode changes to Teach mode
	Opens the visualization window
	Enables or disables DryRun mode
/	Opens or closes the connection to the robot
USB Ethernet	Connection method selection window
	Turns drives on or off
	Activates the movement of the robot to the home position
	Resets the error
	Pauses or resumes the robot
	Enables the procedure for zeroing the axes
	Deletes the selected program

If the background color of the following buttons is yellow, it means that:

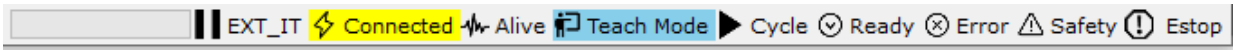
1.		DryRun mode included
2.		Drives are turned on
3.		The robot is in the home position
4.		The robot's work is suspended
5.		Axes zeroing has been done

7.4 Program control bar



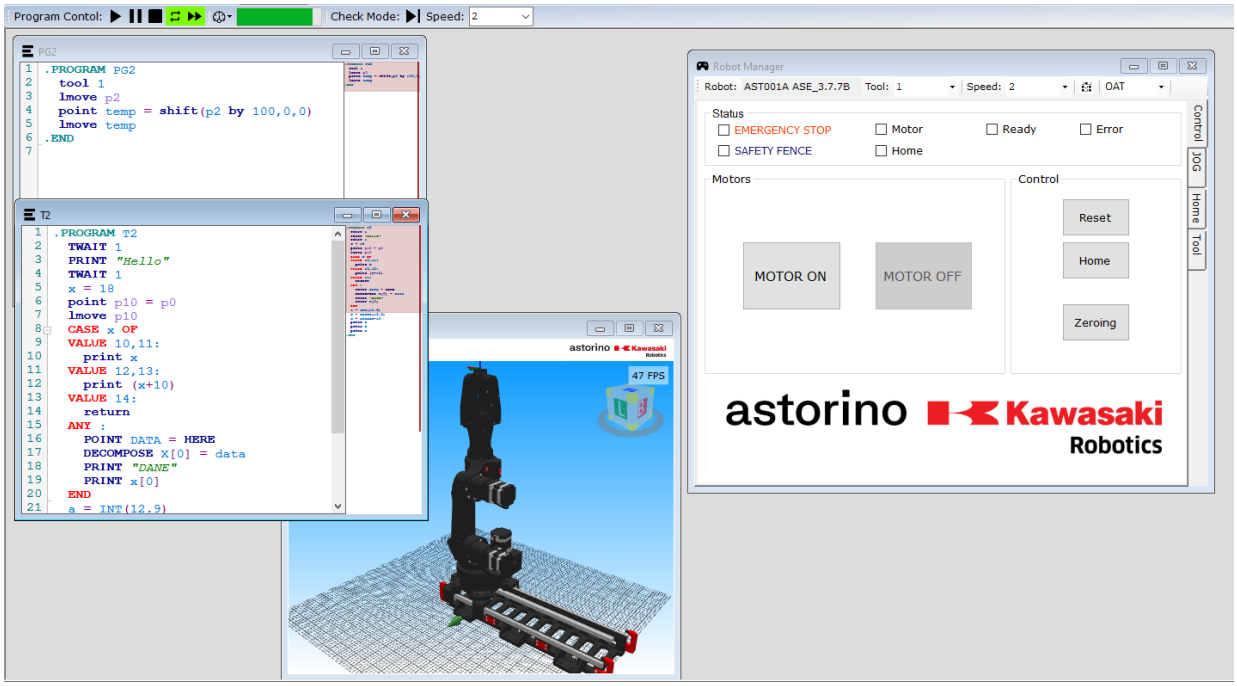
	Enables the execution of the active program
	Pauses or resumes the robot
	Stops the currently executed program
	Enables or disables the looping of the program
	Switches between continuous or stepwise program execution modes
	Changes the monitoring speed
	Current monitoring speed value [0-100%]
	Activates the transition to the next step of the program
	Sets the motion speed in teach-in mode (Teach)

7.5 Status bar



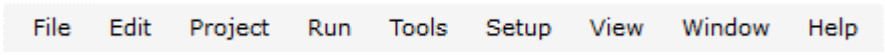
	Status of receiving or sending data from/to the robot
	Robot standby status
EXT_IT	Robot external standstill status
Connected	Status of the connection
Alive	Lifebit of the connection
Mode	The current mode of the robot. Teach or Repeat mode
Cycle	Program playback status
Ready	Robot status
Error	Error Information
Safety	Status of Safety Fence circuit
Estop	Status of Emergency Stop circuit

7.6 Main area



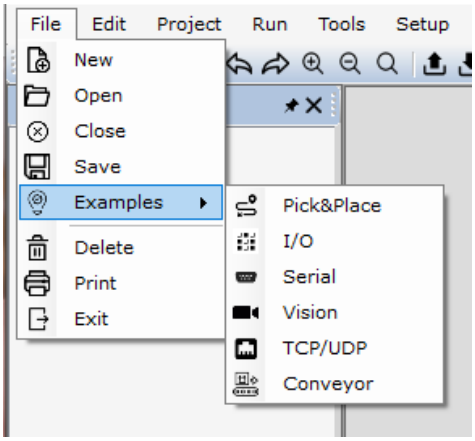
The main area of the astorinoIDE software is mainly used for writing and editing programs, as well as, depending on the settings, for viewing visualizations, controlling the robot in Teach mode, viewing robot inputs/outputs, as well as viewing and editing points.

7.7 Menu bar



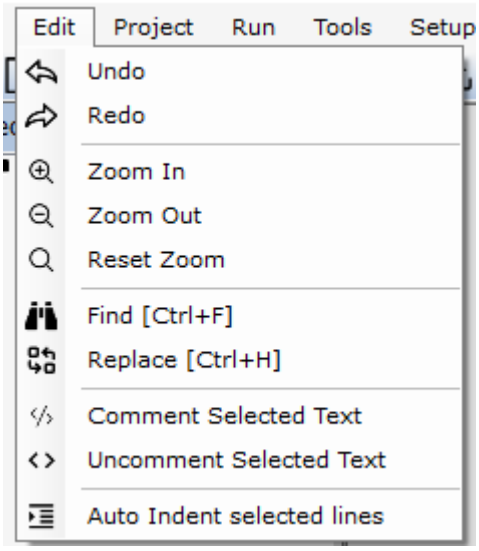
The Menu bar is used to control the program, as well as to enable software and robot configuration windows

7.7.1 File



	Opens the window for creating a new program
	Imports the program from a *.pg file
	Saves the active program to a file on your computer
	Prints the currently active program
	Closes the active program window
	Examples of programs
	Deletes the selected program
	Closes the application
	Sample project with Pick & Place program
	Example project with a program using Robot I/O
	Example project with serial communication program
	Example project using a vision system
	Sample project with programs implementing communication over TCP/IP and UDP
	Sample project with a conveyor tracking program

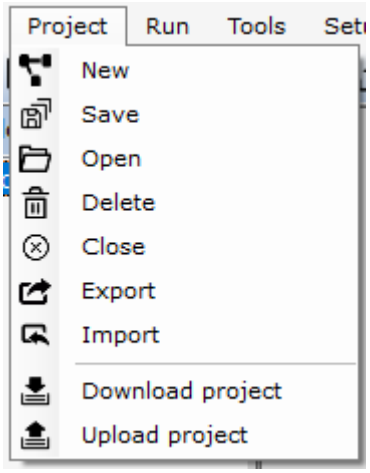
7.7.2 Edit



	Undoes the last edit operation of the program code
	Retries the last edit operation of the program code
	Enlarges text in the program window
	Reduces text in the program window
	Resets the zoom to its default state
	Search for specific text in the program window

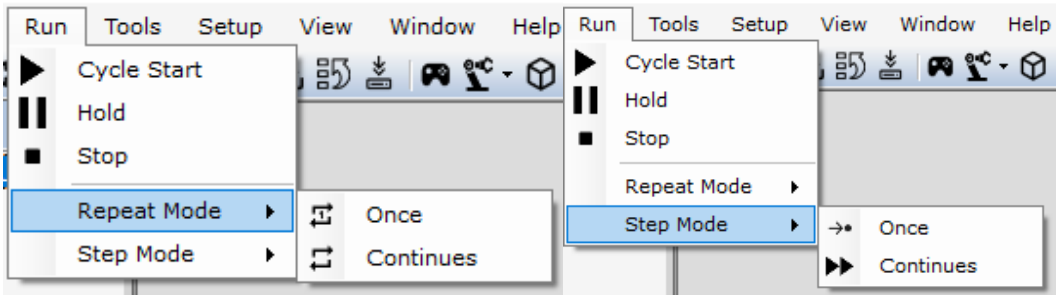
	Search for and change specific text in the program window
	Comment out the selected code fragment
	Uncomment the selected code snippet
	Automatically tab the selected code fragment

7.7.3 Project



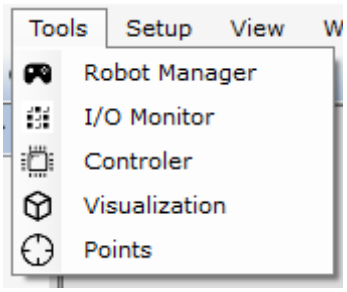
	Opens the window for creating a new project
	Saves the entire project to disk on your computer
	Opens a project selection window to open
	Opens the project selection window to delete
	Closes the currently open project
	Exports the entire project to a separate compressed file in * format. ASzip
	Imports the project from the format *. ASzip
	Downloads the entire project from the robot's memory
	Sends the entire project to the robot's memory

7.7.4 Run



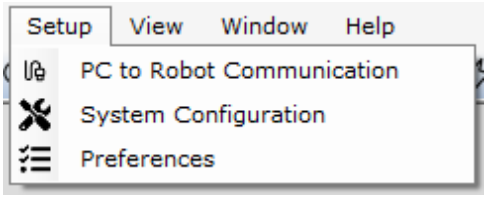
▶	Enables the execution of the active program
⏸	Pauses or resumes the robot
■	Disables the currently executed program
	Enables a program loop
	Disables program looping
→•	Switches to stepwise program execution mode
▶▶	Switches to continuous execution of a program

7.7.5 Tools



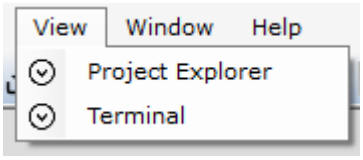
	Opens Robot Manager – allows movement in manual mode (Teach)
	Opens a window for viewing the robot's inputs and outputs
	Opens the backup management window
	Opens the visualization window
	Opens the window for viewing and editing saved points

7.7.6 Setup



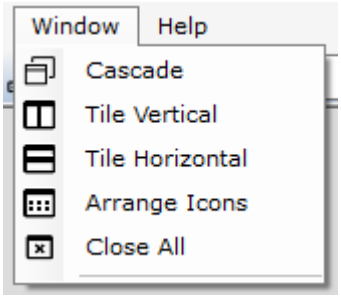
	Opens a window to select how USB or Ethernet communicates
	Opens the robot system settings window
	Opens the astorinoIDE settings window

7.7.7 View



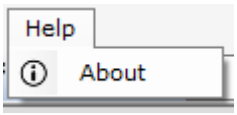
	Hides or shows the Terminal or Project Manager view

7.7.8 Window



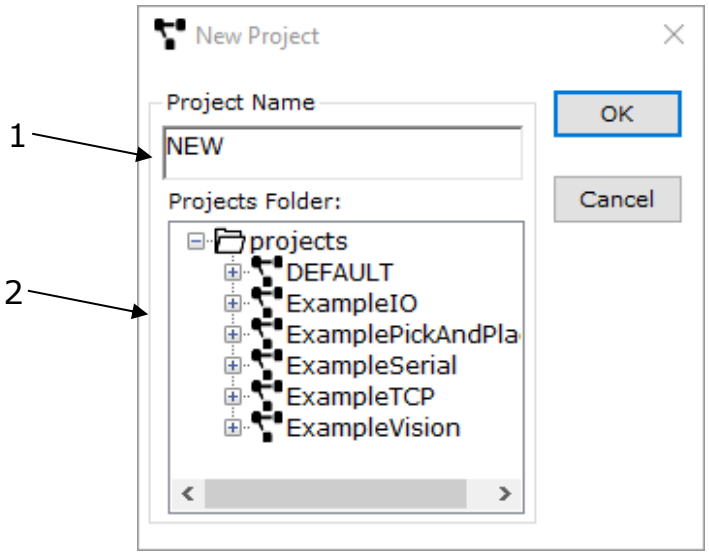
	Switches the main area window to cascade mode (panes)
	Sets windows to column mode
	Sets windows to line mode
	If the window is minimized, it is arranged in the lower left corner of the main area
	Closes all open windows

7.7.9 Help



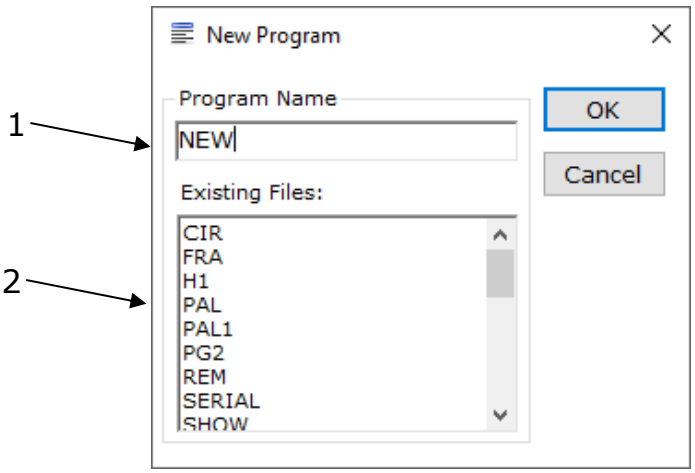
ⓘ	Opens the software version information window

8 New Project Window



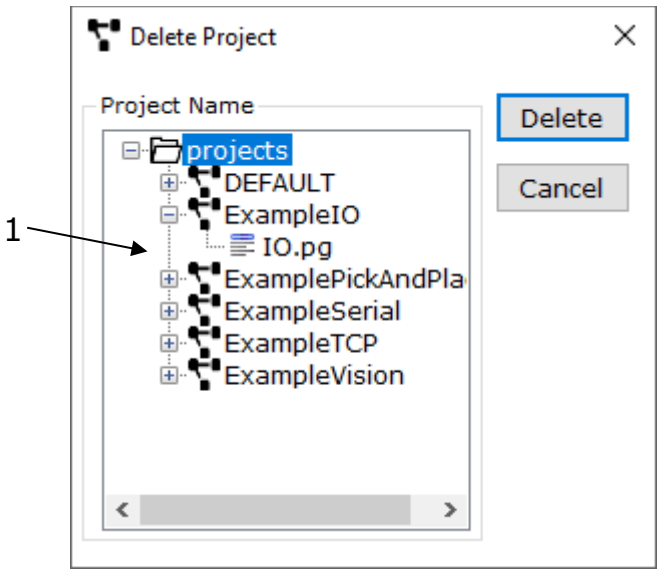
1	Name of the newly created project
2	Names of projects currently on disk and programs in them

9 New program window



1	Name of the newly created program
2	Names of programs currently in the project

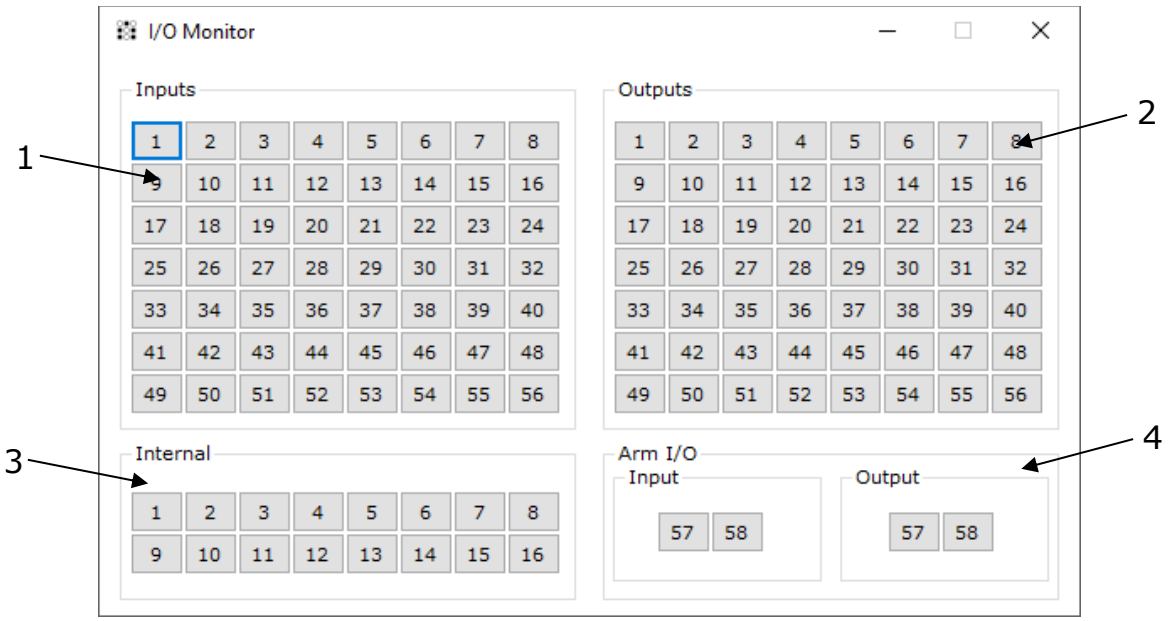
10 Delete Projects window



1	Names of projects currently on disk and programs in them

The [DELETE] button deletes the currently selected project.

11 IO Monitor



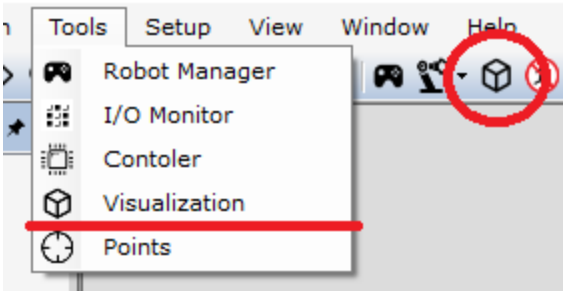
1	Physical inputs and inputs of the MODBUS TCP protocol
2	Physical and MODBUS TCP protocol outputs
3	Internal signals
4	Inputs and outputs located on the robot arm (version B of the astorino robot)

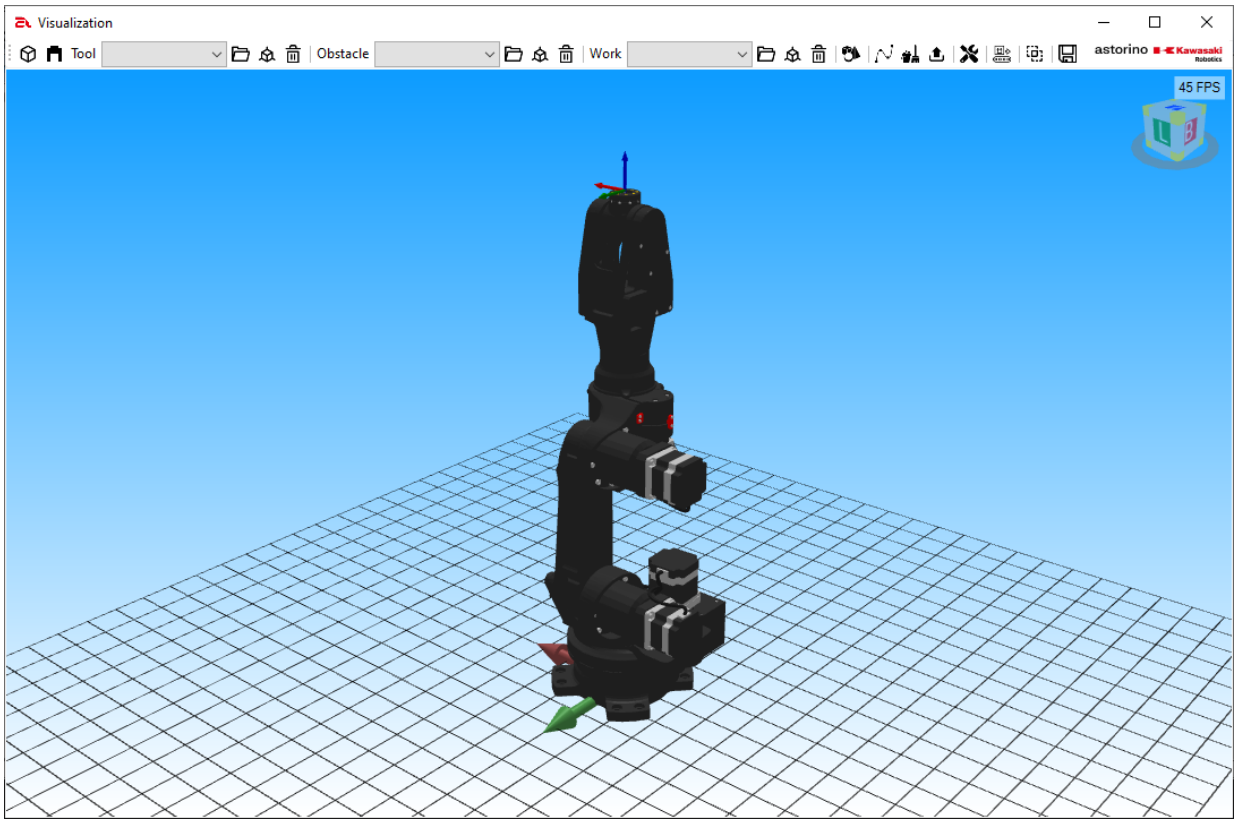
The high state of the signal is indicated by the yellow button lights up.

Outputs and internal signals can be controlled by clicking the mouse on the appropriate signal number.

12 Visualization window

To open the visualization window and see the operation of the Astorino robot in real time, click one of these two buttons





12.1 Visualization window - operations

The visualization window allows you to add 3D objects to the scene with the robot. The program supports stl files and allows you to add basic three-dimensional shapes. You can add each feature as one of three object types:

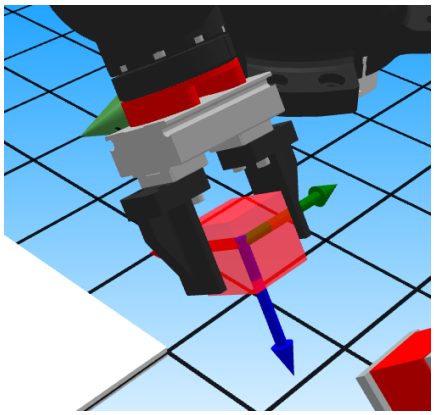
1. Obstacle – objects of this type are static objects of the scene
2. Work – objects of this type can be moved by a robot
3. Tool – objects of this type always move according to the wrist of robots.

The visualization window menu consists of the following elements:

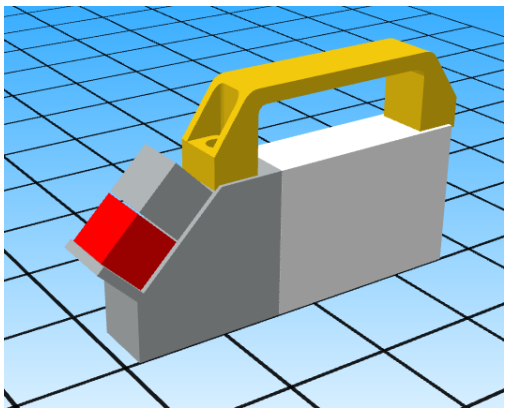
	Enables or disables the Working Space view
	Enables or disables the 3D model of a standard gripper
	Lists Tool, Obstacle, or Work objects
	Opens the .stl file and loads it as one of the Tool, Obstacle, or work class objects
	Enables the object modification menu, allows you to change the position of the object or change its color
	Deletes the currently selected object in the drop-down list
	Enables the menu of the 3D Simple Shapes Generator
	Enables the generation of robot trajectory visualizations
	Disables and clears the visualization of the robot's trajectory
	Saves robot trajectory visualization points to .traj files
	Enables the visualization window settings menu
	Enables the virtual conveyor settings menu
	Enables the object collision detection settings menu
	Saves the visualization to a .xml file in the project folder

12.2 Object types

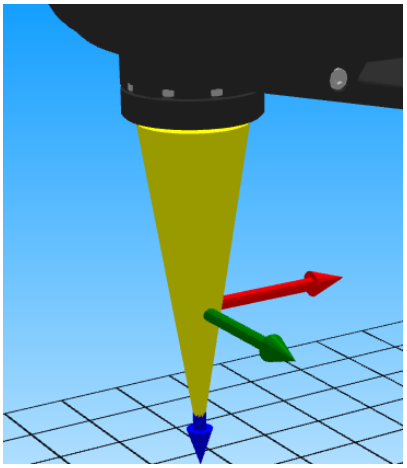
Work – Work class objects can be moved by a robot. For an object to be captured, the TCP point must be inside the work object and the control signal must be in a high state.



Obstacle – Obstacle objects are static visualization elements. They allow you to build a visualization scene, are a visual aspect and potential obstacles.



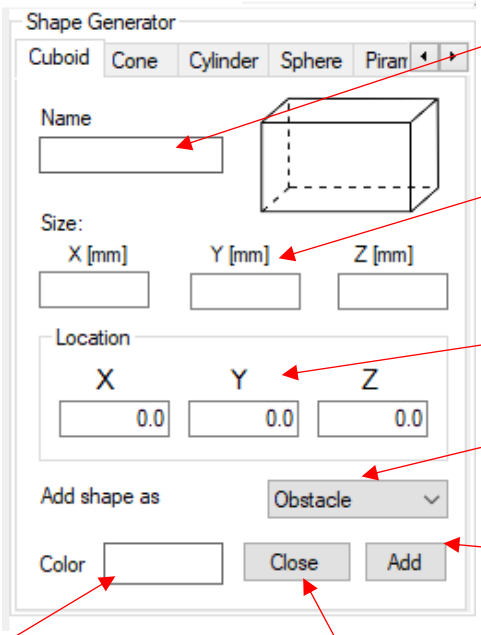
Tool – Tool class objects are objects that are permanently attached to a wrist of robots. Thanks to these objects, you can create your own tools, which are mounted on the robot flange.



12.3 Simple Shape Generator

The generator of simple three-dimensional shapes allows you to generate the following elements:

- Cube,
- Cuboid,
- Cone,
- Cylinder,
- Sphere,
- Pyramid,
- Pipe,



Object name, when not entered – assigns another free name to the object automatically.

The size of the object, depending on the figure, should be given from 1 to 3 parameters.

The position under which the figure is to be created.

Selecting the class of the object to be created: Tool, Obstacle, Work.

Add an object to a visualization

Choosing an object color

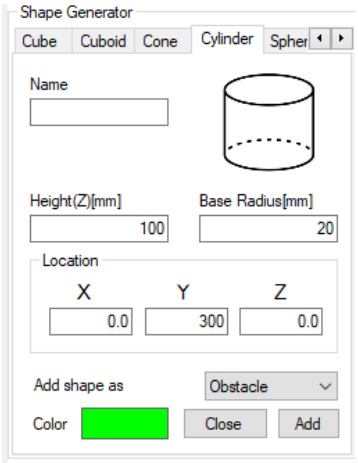
Close the generator menu

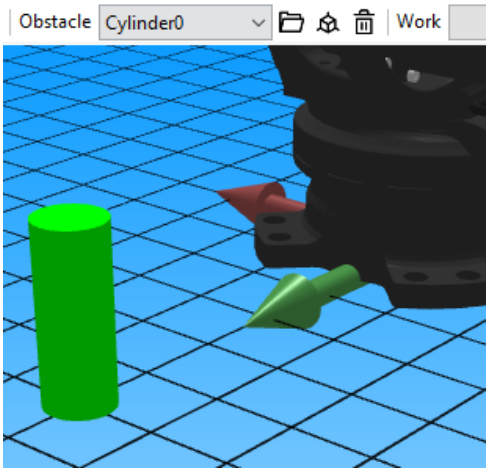
Example

Add an obstacle cylinder with the following parameters to the visualization:

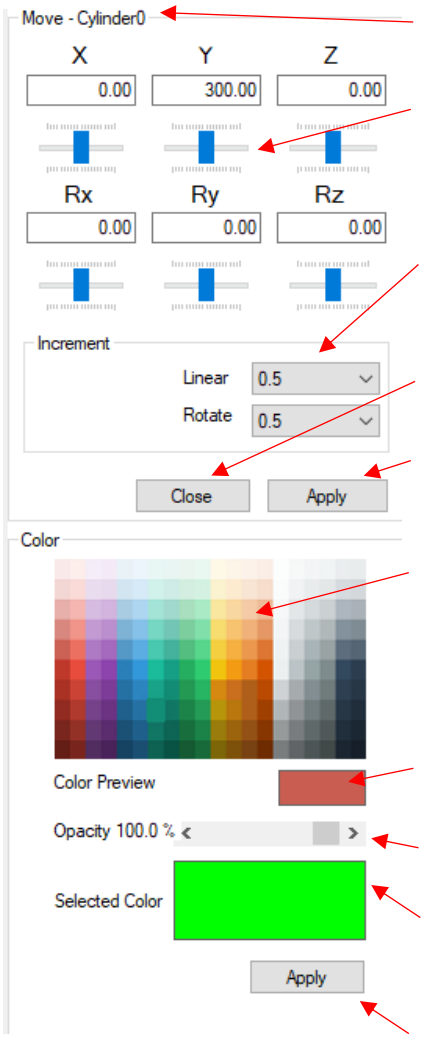
- 100 mm high,
- Base radius 20mm,
- Green color
- Starting position (0.300.0 [x,y,z])
- Any name

To add such an object, enter the following data in the generator menu and confirm with the [ADD] button. The object is added.





12.4 Objects modifications menu



Name of the currently modified model.

Sliders and text boxes to change the position of an object on a visualization.

Setting the resolution of the above sliders

Close the modification window.

Approval of object position changes.

The quick color selection area, clicking the mouse button on a specific color will allow you to select it as the color of the object being modified

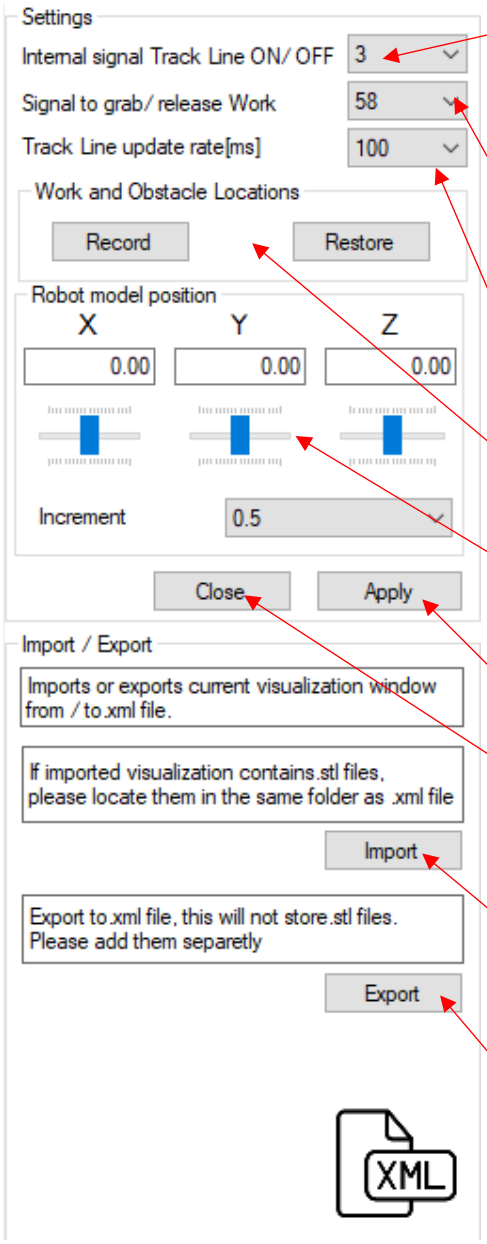
Preview window selected colors.

Set an object opacity.

Currently selected color, clicking on this area will open the advanced color selection window

Approval of amendments.

12.5 Visualization Settings menu



Internal signal (from the pool 2001-2016) that enables or disables the generation of trajectory points on the visualization.

Output signal controlling the capture of Work class objects in visualization by the robot.

Time every another trajectory visualization point is created.

Save and restore the position of elements on the visualization.

Changing the position of the robot on the visualization.

Apply settings.

Close the settings menu.

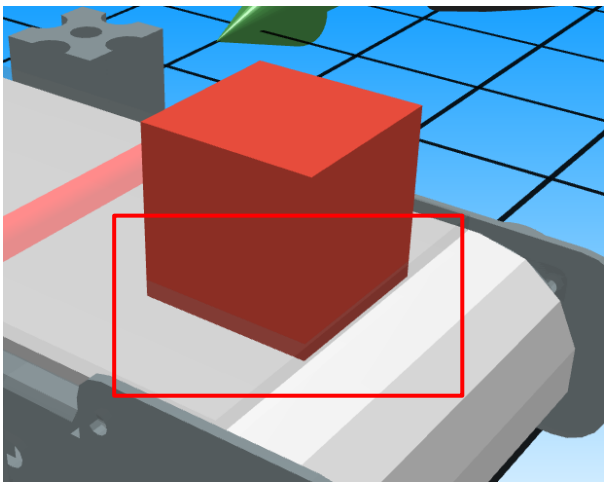
Import visualizations from an .xml file.

Export visualizations to xml files.
The export does not save the opened stl files. Only their names. Copy the files separately.

12.6 Virtual conveyor

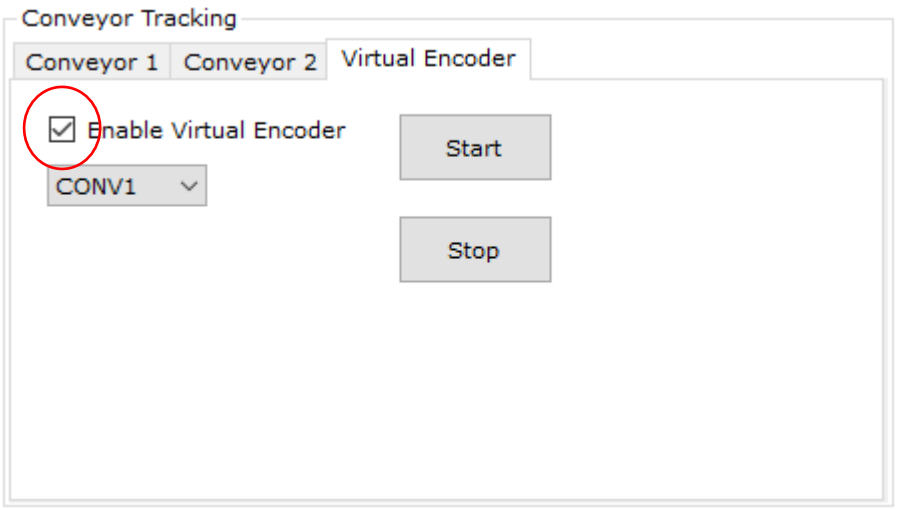
The virtual conveyor belt allows you to simulate applications that use conveyor tracking. Only Work objects can be moved in a visualization by this object.

In order for an object to be captured by a conveyor belt, it must penetrate the shape of the created cuboid representing the virtual conveyor belt.

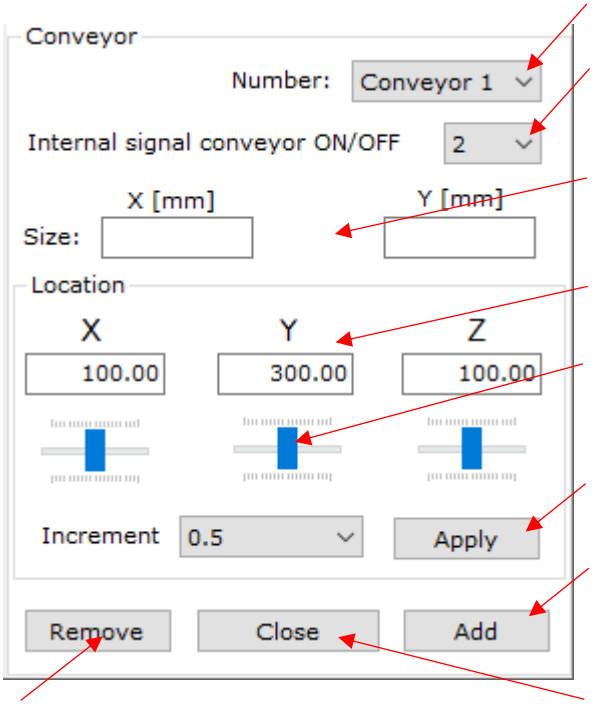


The conveyor belt in the visualization can read displacement data from a physical external encoder or from a virtual one.

To use the virtual encoder, this option must be enabled:



WARNING!
Virtual and physical encoder is based on the correct settings of the resolution and direction of the conveyor.



Conveyor number - 1 or 2

Internal signal used to turn the virtual conveyor belt on or off

Conveyor belt size in mm

Conveyor position

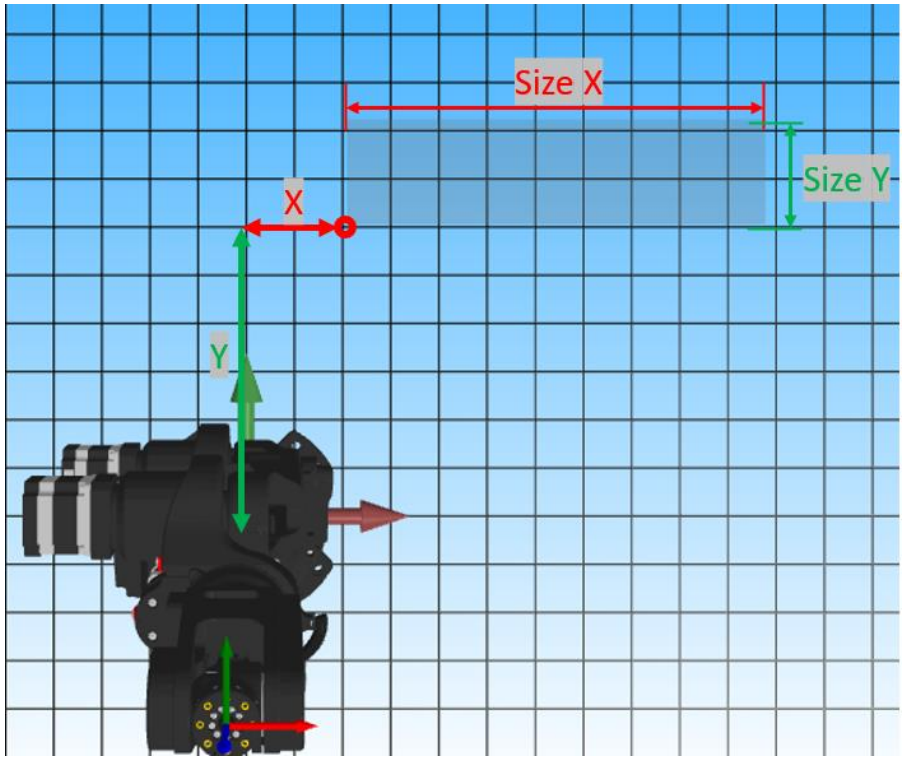
Change the position of the selected conveyor belt

Position confirmation

Add conveyor

Close the conveyor configuration window

Delete the selected conveyor belt



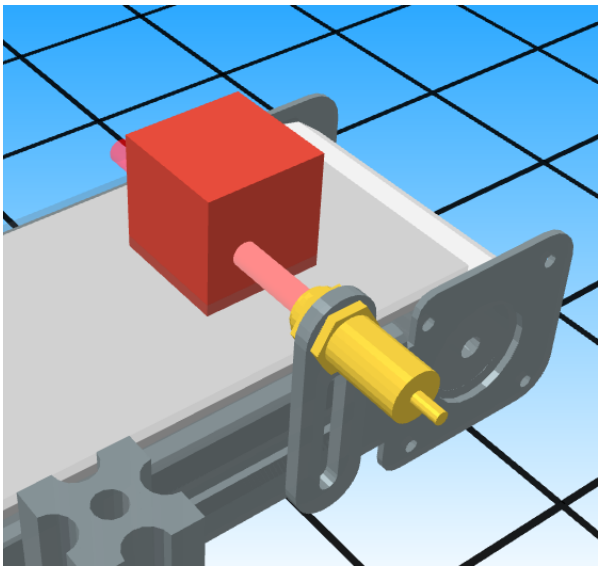
Basic dimensions of a virtual conveyor belt

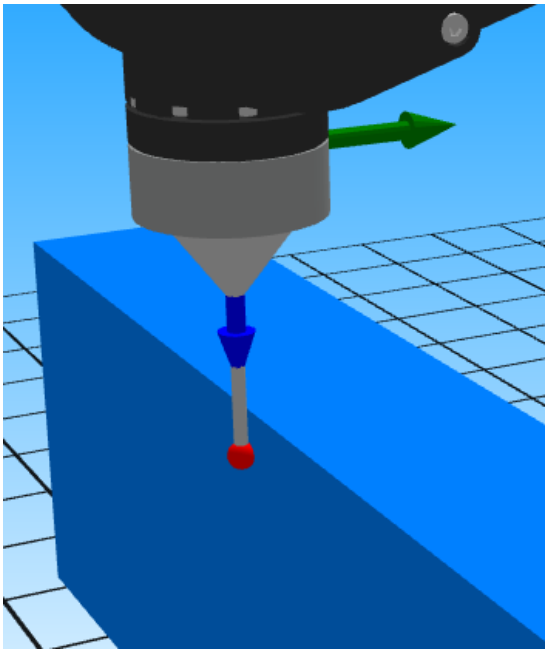
12.7 Collision detection of virtual objects

Collision detection of virtual scene elements can be set between objects of type:

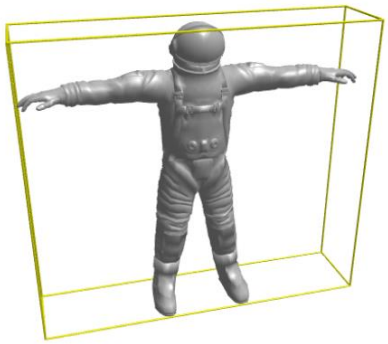
- Obstacle – Work
- Obstacle – Tool

The application is, for example, object detection on a virtual conveyor belt, or a virtual touch probe.





Collision detection detects the interpenetration of virtual object boundaries. The boundary is the perpendicularities inside which the object can be placed.



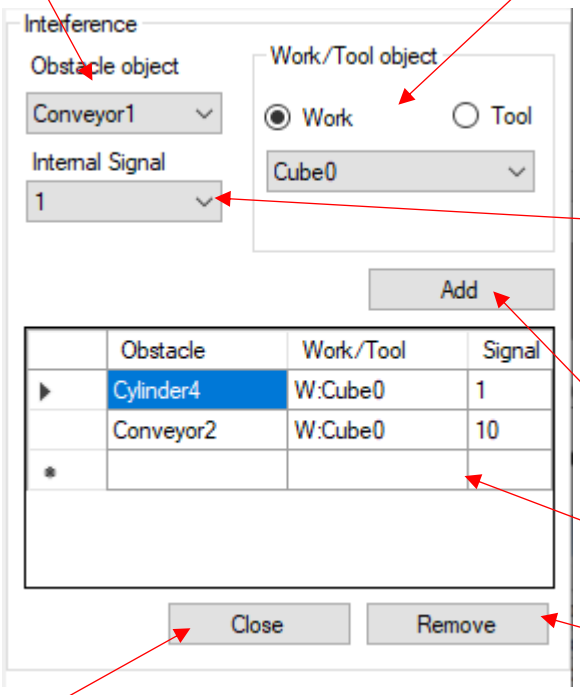
This procedure significantly reduces computational complexity, but does not ensure 100% correct detection. Keep this in mind when creating an app.

WARNING!

As in the above example with an astronaut, we can imagine a situation where another object can move under the arm of the model. Collision detection will detect penetration, but from the perspective of a human (user) such a collision did not occur.

Selecting an Obstacle object

Selecting a Work or Tool object



The number of the internal signal that will be activated when a collision between two objects is detected

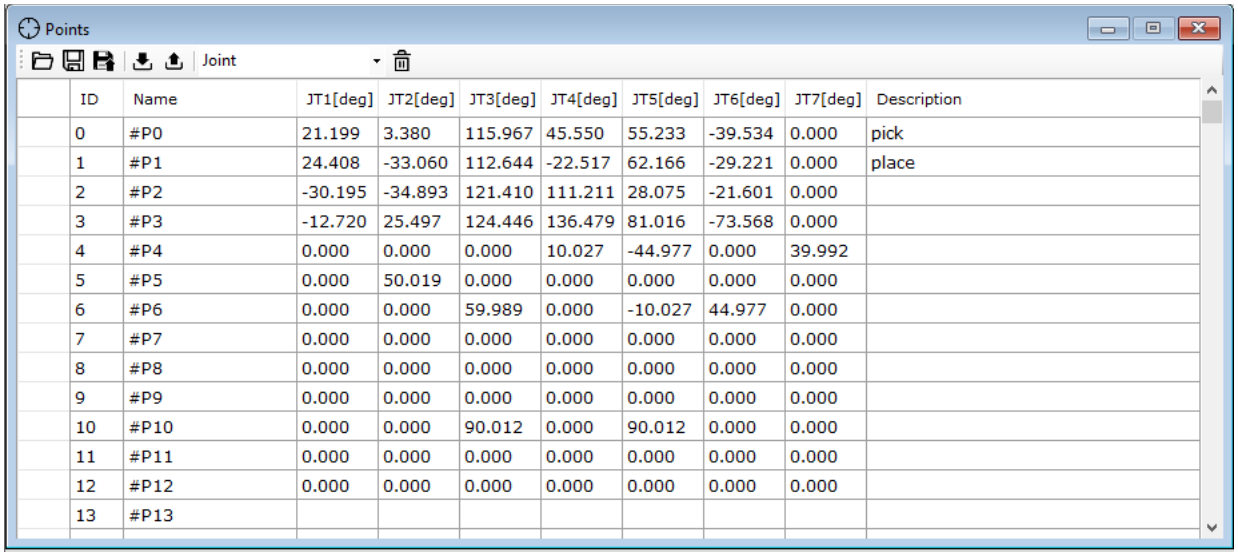
Adding a collision detection task between selected objects

Table with all set detections

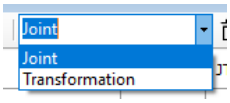
Removal of the selected pair of objects for which collision detection is attached

Closing the collision detection configuration window

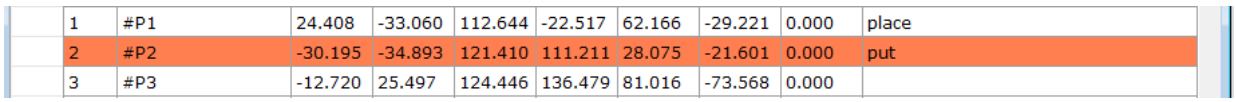
13 Points window



ID	Name	JT1[deg]	JT2[deg]	JT3[deg]	JT4[deg]	JT5[deg]	JT6[deg]	JT7[deg]	Description
0	#P0	21.199	3.380	115.967	45.550	55.233	-39.534	0.000	pick
1	#P1	24.408	-33.060	112.644	-22.517	62.166	-29.221	0.000	place
2	#P2	-30.195	-34.893	121.410	111.211	28.075	-21.601	0.000	
3	#P3	-12.720	25.497	124.446	136.479	81.016	-73.568	0.000	
4	#P4	0.000	0.000	0.000	10.027	-44.977	0.000	39.992	
5	#P5	0.000	50.019	0.000	0.000	0.000	0.000	0.000	
6	#P6	0.000	0.000	59.989	0.000	-10.027	44.977	0.000	
7	#P7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
8	#P8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
9	#P9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
10	#P10	0.000	0.000	90.012	0.000	90.012	0.000	0.000	
11	#P11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
12	#P12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
13	#P13								

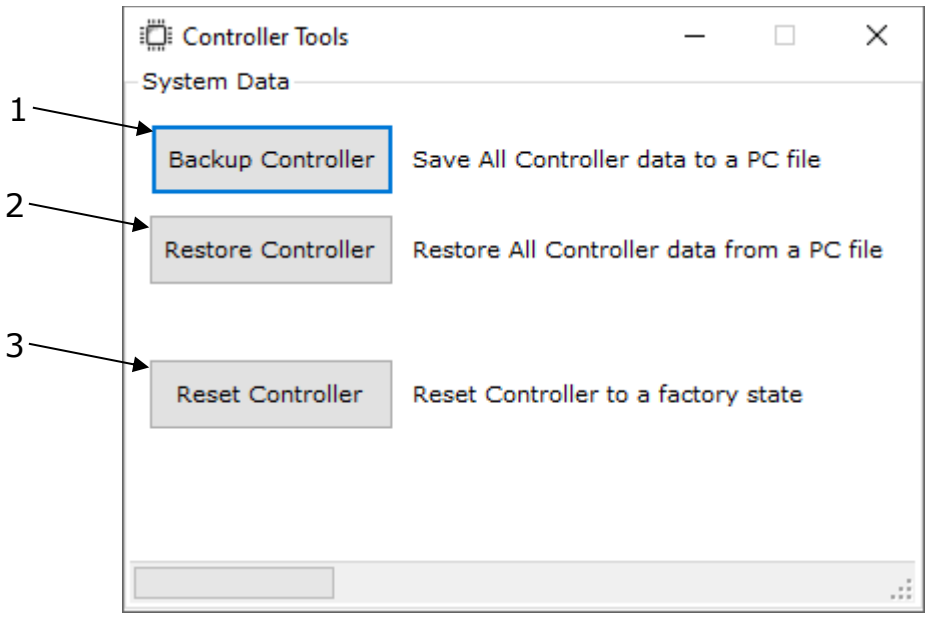
	Imports points from a file saved in the *.lc format.
	Saves changes of the points to the project directory. Note: This action does not send points to the robot's memory
	Exports points to a *.lc file.
	Sends the currently selected points (Joint or Transformation) to the robot's memory
	Downloada currently selected points (Joint or Transformation) from the robot's memory
	Choice

astorinoIDE allows you to add comments to points (Description), as well as manual editing of points. The manually modified point is highlighted in red and requires saving or sending to the robot to make changes.



1	#P1	24.408	-33.060	112.644	-22.517	62.166	-29.221	0.000	place
2	#P2	-30.195	-34.893	121.410	111.211	28.075	-21.601	0.000	put
3	#P3	-12.720	25.497	124.446	136.479	81.016	-73.568	0.000	

14 Controller window

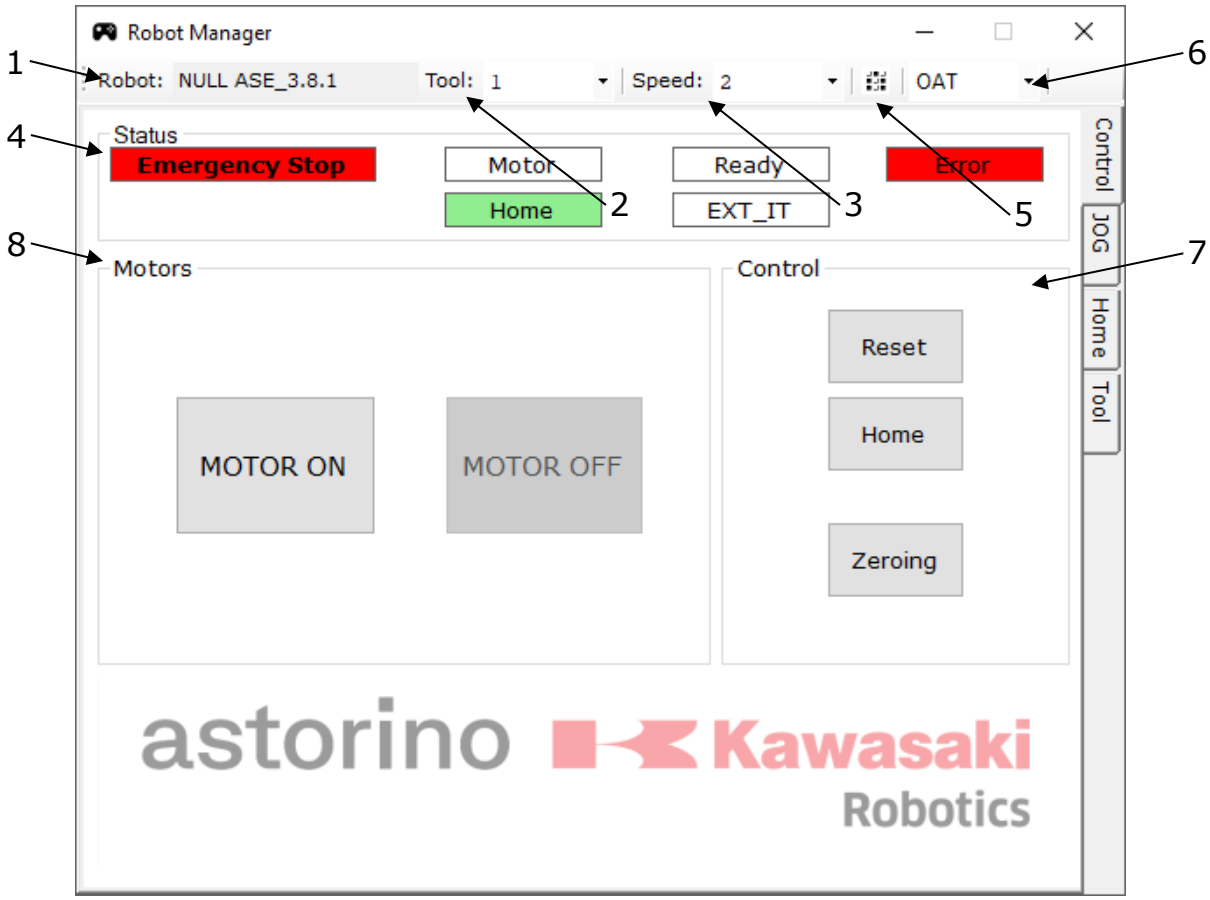


1	Creates a backup copy of the robot controller, saves a *.as file, which contains all data from the robot's memory
2	Loads a backup copy file into the robot's memory
3	Resets the controller to the factory state

15 Robot Manager window

The Robot Manager window is only available after connecting to the robot. It allows you to manually (Teach mode) move the arm, as well as change the HOME position, determine the TOOL data, as well as teach points.

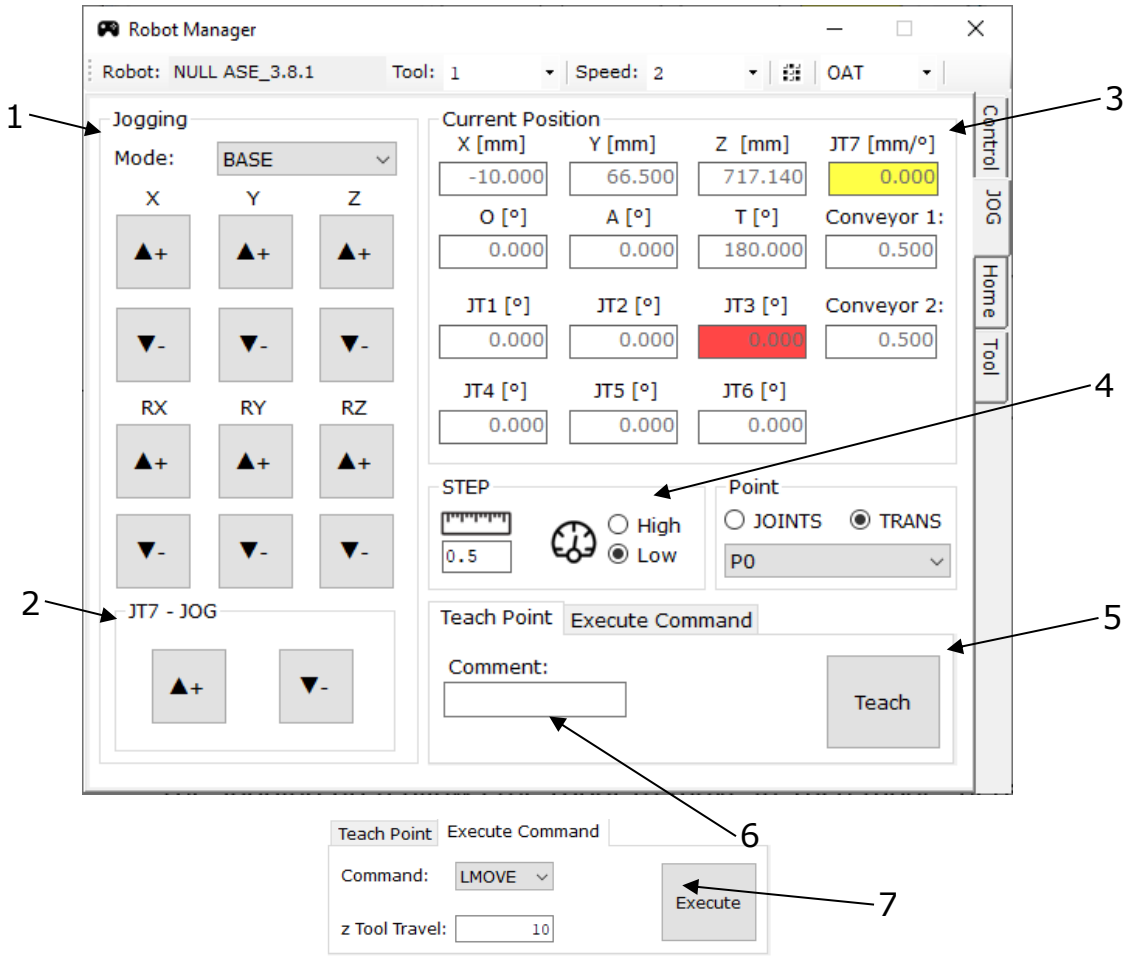
15.1 Control tab



1	Shows the serial number of the robot and the current firmware
2	Shows and allows you to change the currently active TOOL number
3	Shows and allows you to change the current robot speed in Teach mode
4	Current robot status
5	Opens the robot I/O view window
6	Allows you to switch between displaying the current orientation in OAT (ZYZ) or RPY (XYZ) angles
7	This area allows you to reset the error, get to the HOME position and start the reset procedure.
8	This area allows you to turn the drives on or off

15.2 JOG Card

The JOG card allows you to control the robot in manual mode, and also shows the current position of the arm.

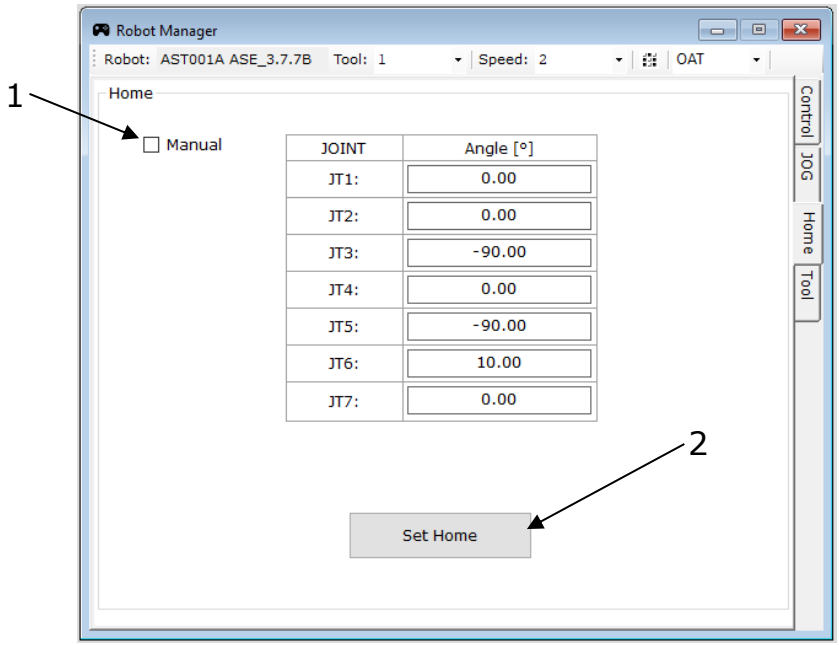


1	The Jogging area allows the robot to move in Tach mode, here you can also change the interpolation method (BASE, TOOL, JOINT, CONV)
2	JT7-JOG area allows manual control of axis 7 (linear track)
3	Current position of the arm, yellow indicates a position close to the maximum range, red indicates the maximum position of an axis.
4	Step motion settings
5	Points teaching area, clicking the Teach button saves the current arm position as the point selected in the list under Point. The TeachPoint area also allows you to add comments to point data.
6	The Point area allows you to select the point you want to save or the point you want to move to in Teach mode
7	The Execute Motion area allows you to perform one of the movements (LMOVE, JMOVE, LAPPRO, JAPPRO, JUMP) to the point selected in the Point area

15.3 Home tab

The Home tab allows you to modify your home position. Clicking the [Set Home] button (2) saves the current position of the arm as the home position.

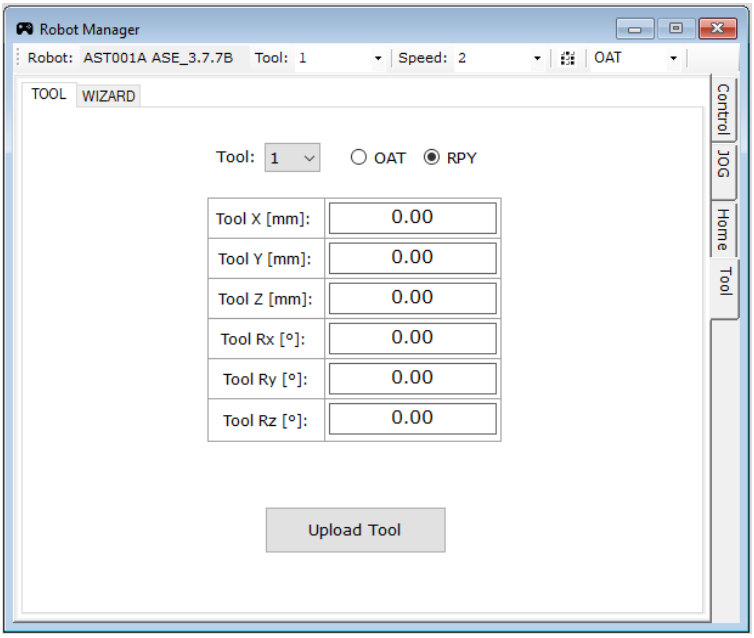
Turning on Manual mode (1) allows you to manually edit the home position.



15.4 Tool tab

15.4.1 Tool tab

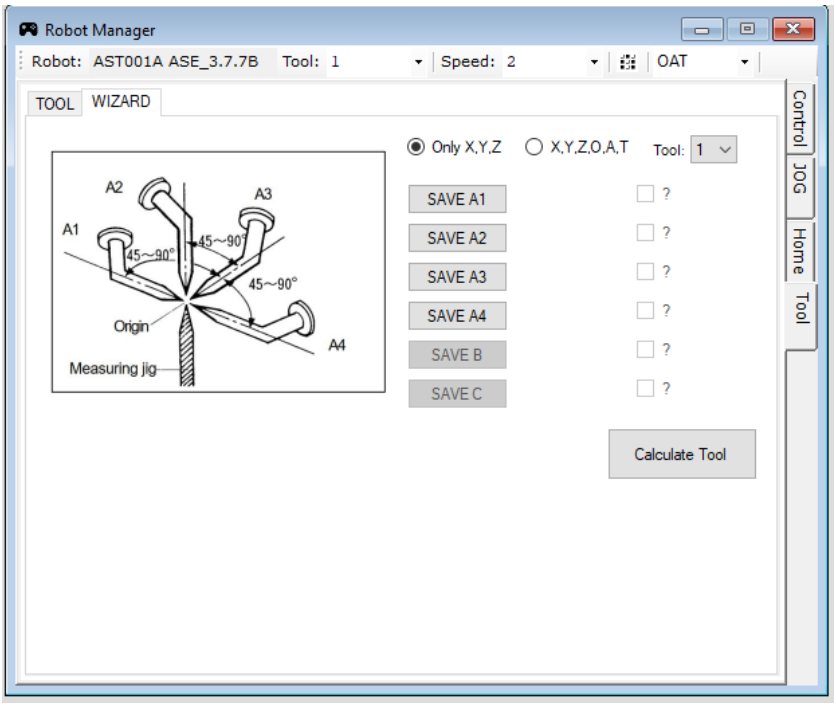
Allows manual modification of the coordinate system of the tool (Tool)



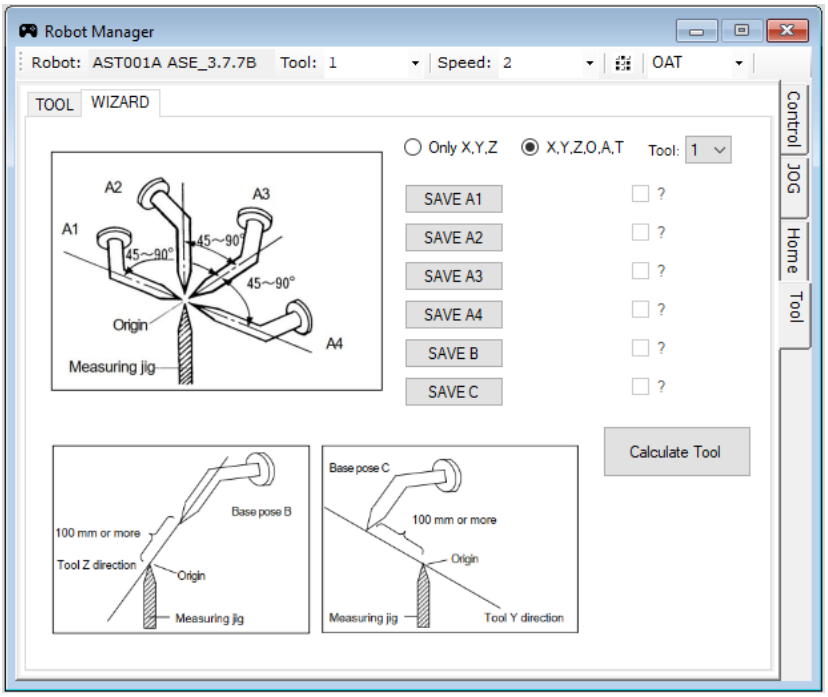
15.4.2 WIZARD tab

The WIZARD tab allows you to automatically calculate the coordinate lay-out of the tool using the 4-ro or 6-point method.

4-point method (XYZ calculation):

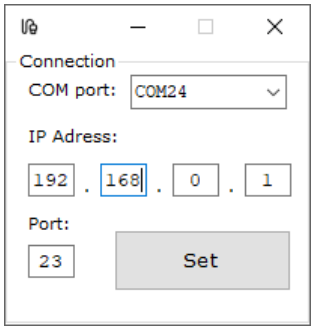


6-point method (calculation of XYZ and OAT):



15.5 PC to Robot Communication window

This window allows you to select the COM port to which the robot is connected. And changing the settings of the IP address under which the astorino is located.



Warning!

The COM port is detected automatically, no need to change it manually. If more than one robot is connected to the computer, it allows you to choose which unit you want to connect to.

15.6 Preferences window

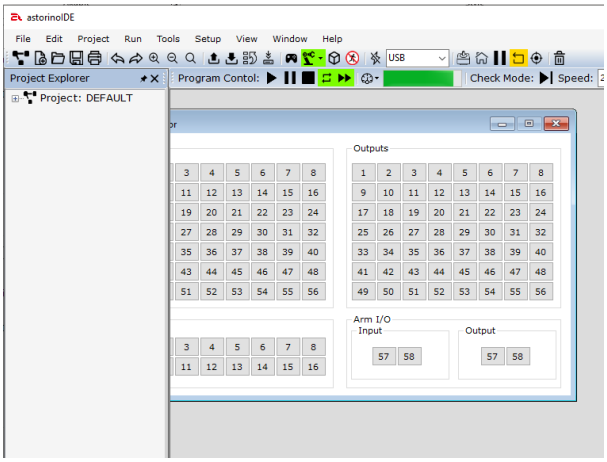
The Preferences window allows you to change astorinoIDE settings. Mainly changes in the functionality of auxiliary windows (Robot Manager, IO Monitor, Visualization, Points)

It allows you to change the visibility of the window from MDI Window to Dialog and vice versa.

The settings are saved on the computer and remembered even after the program is turned off

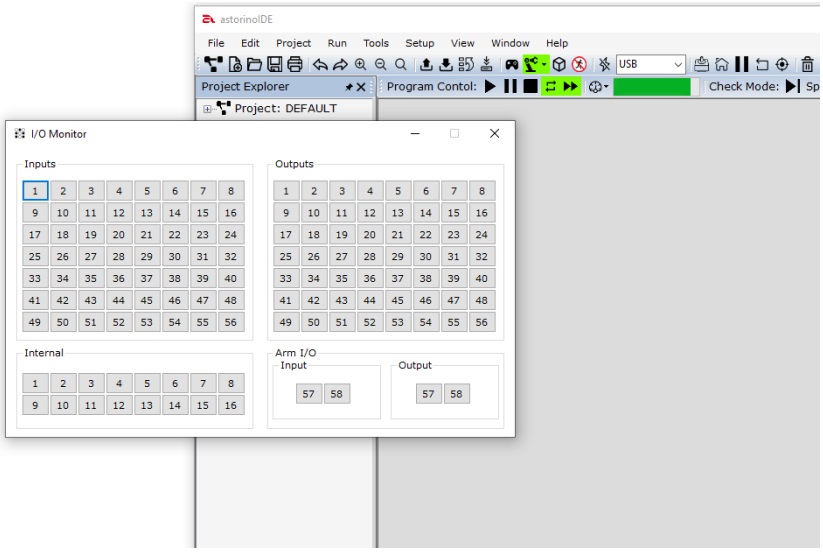
- **Okna typu MDI Window**

MDI windows are windows that are "inside" in the main area and cannot be separated from the main astorinoIDE window

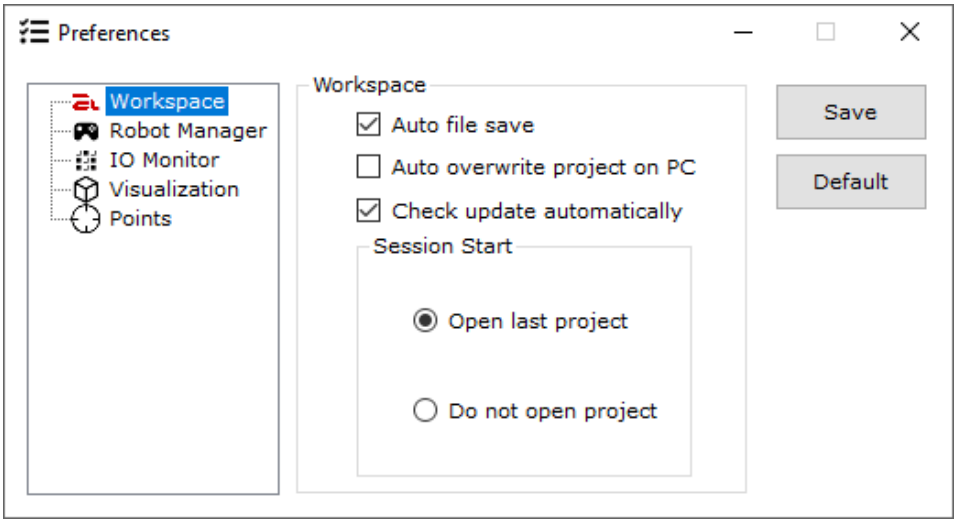


- **Dialog windows**

Dialog windows are windows that open as additional windows of the astorinoIDE program, this allows you to freely place such windows on the user's screen.



15.6.1 Workspace

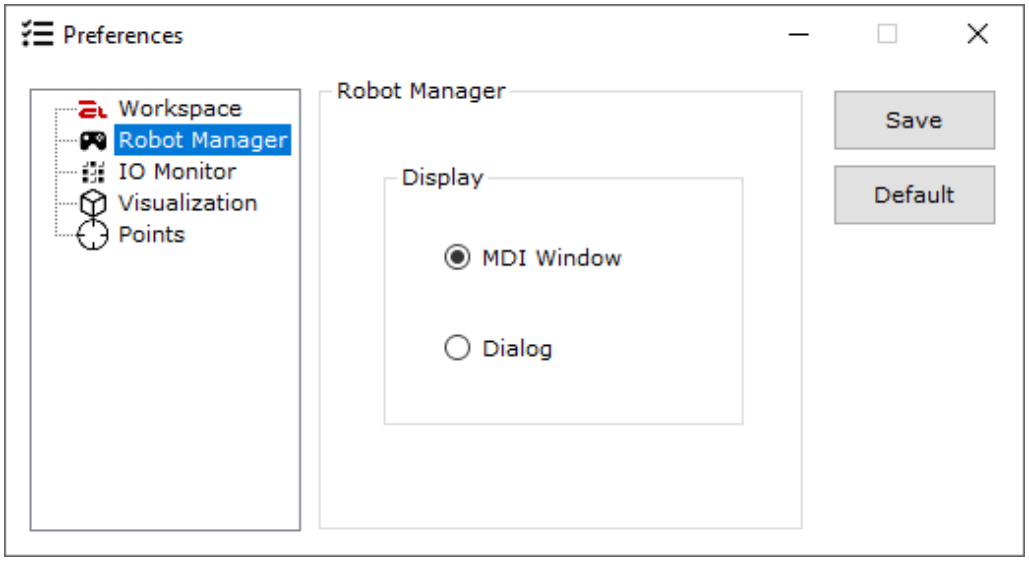


The Workspace area allows you to change system settings such as:

- Automatic saving of files when the program window is closed
- Automatic overwriting of the project on the PC when connected to the robot (the synchronization window does not appear)
- Check for a new software version after startup.
- Choose how to open the environment. We can choose whether the astorinoIDE has or does not open the last used project at startup.

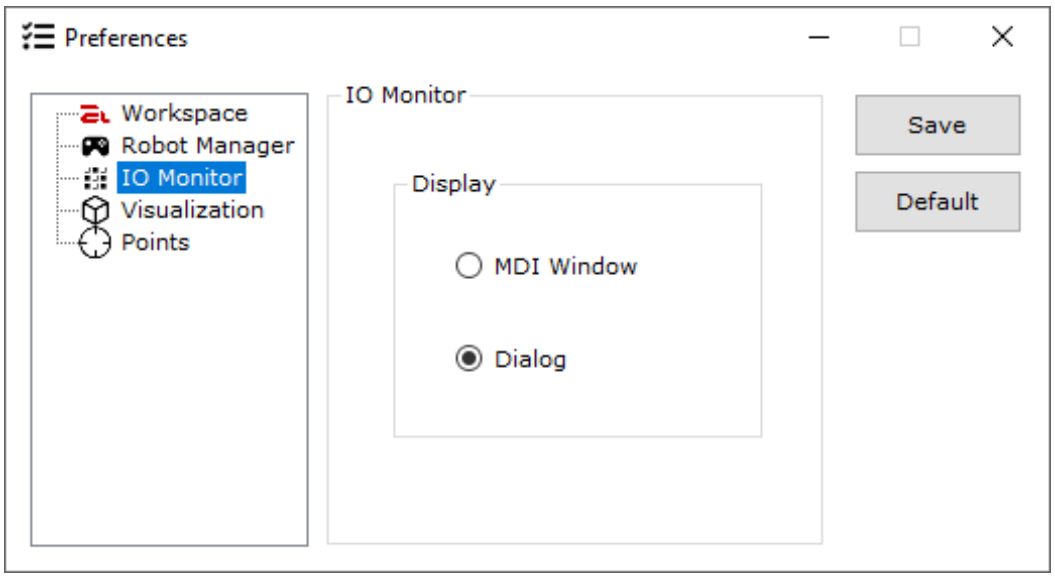
15.6.2 Robot Manager

It allows you to choose how to show the Robot Manager window. Choose whether the window should be displayed as an MDI Window or a Dialog window.



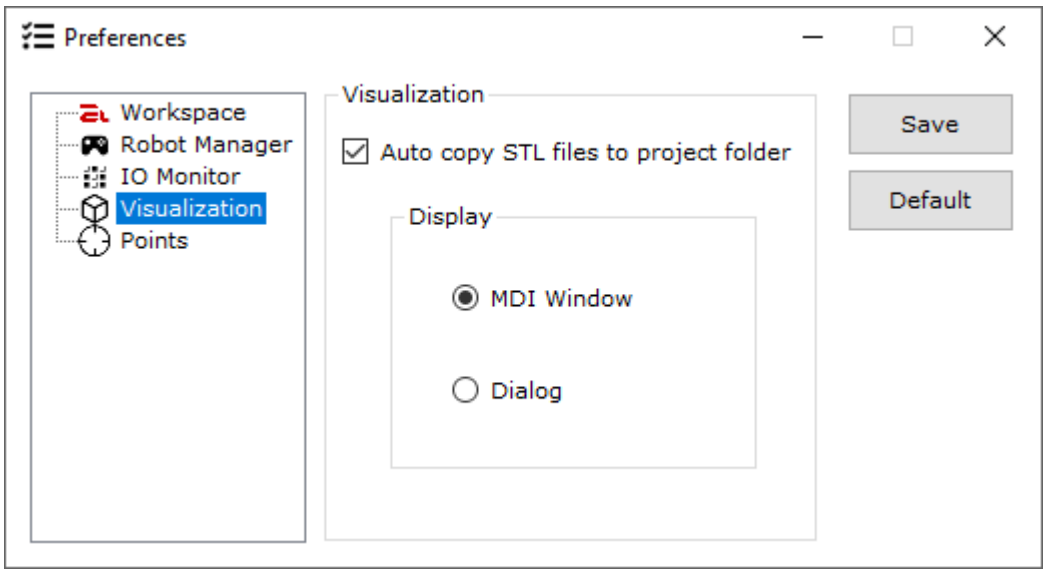
15.6.3 IO Monitor

It allows you to choose how to show the IO Monitor window. Choose whether the window should be displayed as an MDI Window or a Dialog window.



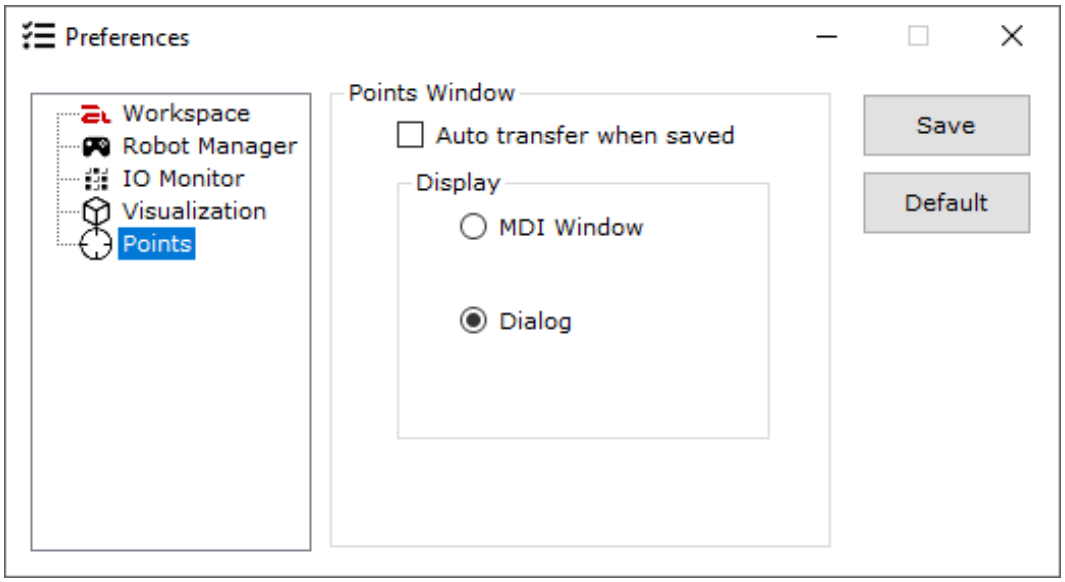
15.6.4 Visualization

Allows you to choose how to show the Visualization window. Choose whether the window should be displayed as an MDI Window or a Dialog window. And also whether *.stl files are to be automatically copied to the project folder.



15.6.5 Points

Allows you to choose how show the Points window. Choose whether the window should be displayed as an MDI Window or a Dialog window. It also allows you to set whether points are to be automatically sent to the robot after saving or not.

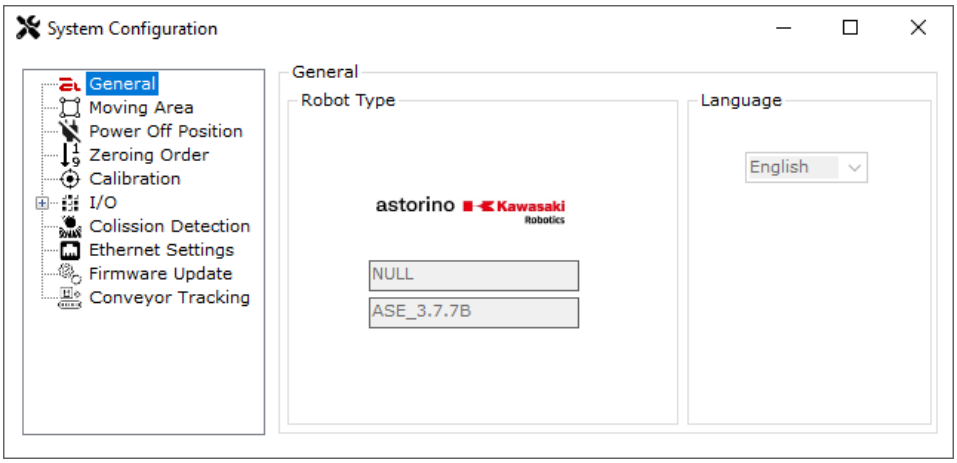


16 System Configuration window

System configuration allows you to view and modify the system settings of the robot.

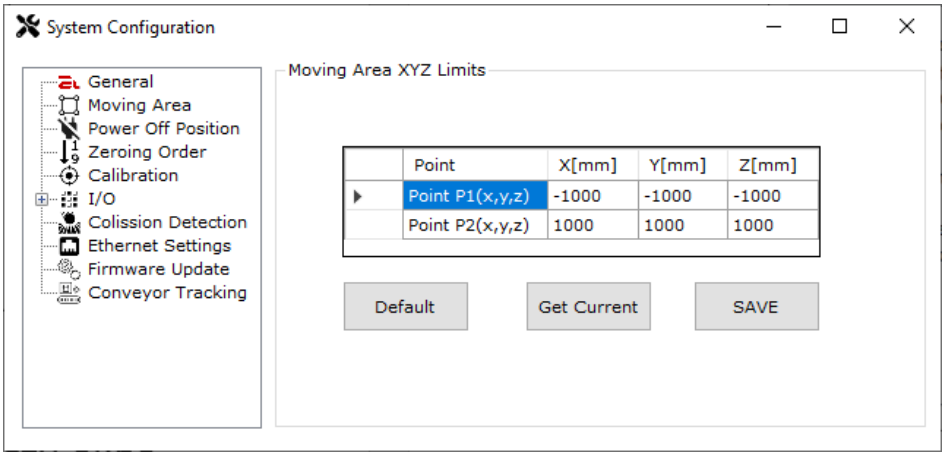
16.1 General

Shows the serial number and firmware version currently uploaded to the robot



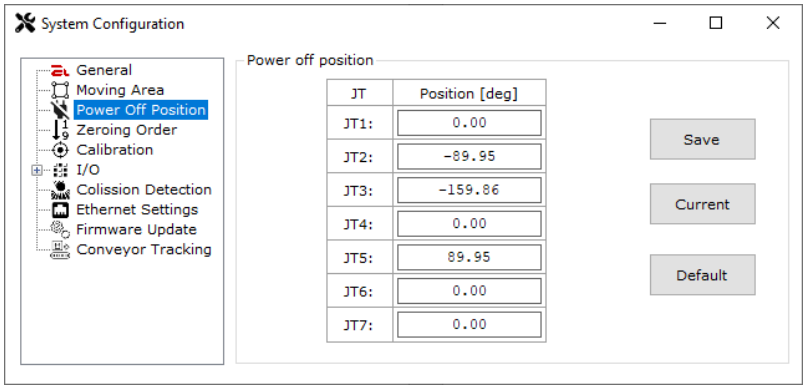
16.2 Moving Area

It allows you to modify the working area of the astorino robot. Point P1 is the point defining the minimum range of work in the axes X, Y, Z, and point P2 is the point defining the maximum range of work in the axes X, Y, Z



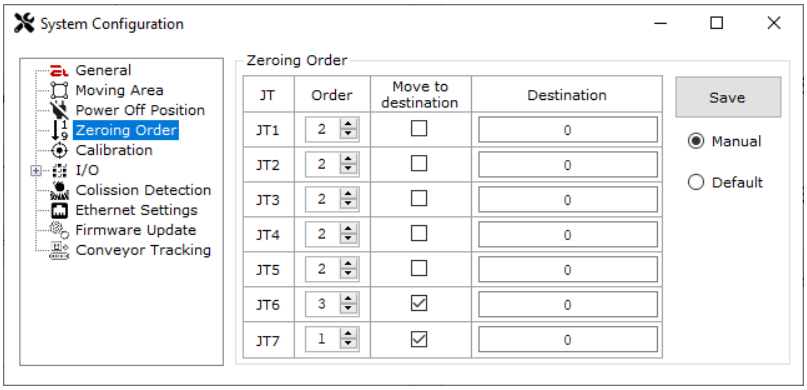
16.3 Power Off Position

This area allows to define the turn off position of the drives. Power Off Position is the position to which the robot will go after switching off the drives, if the zeroing process has been completed.



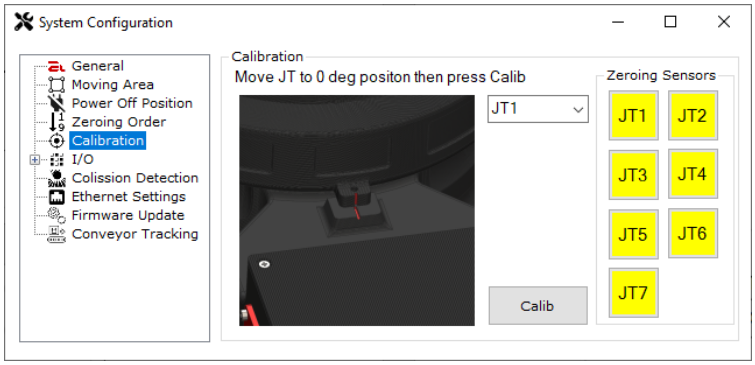
16.4 Zeroing Order

This area allows to define the process of zeroing the axis. It allows you to select the order of zeroing, as well as select whether the axis is to go to the set position (Destination) after resetting.



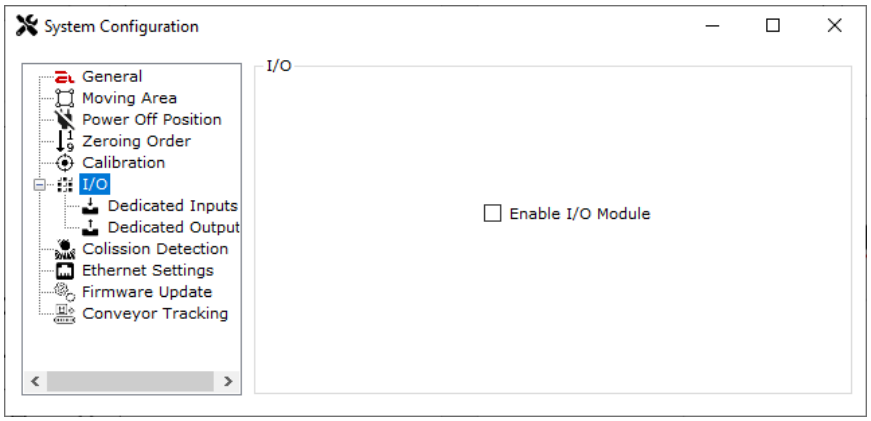
16.5 Calibration

This area allows axis calibration. Calibration of the axis is only necessary in the event of a failure of the SD card in the robot controller or replacement of the printed main body parts.

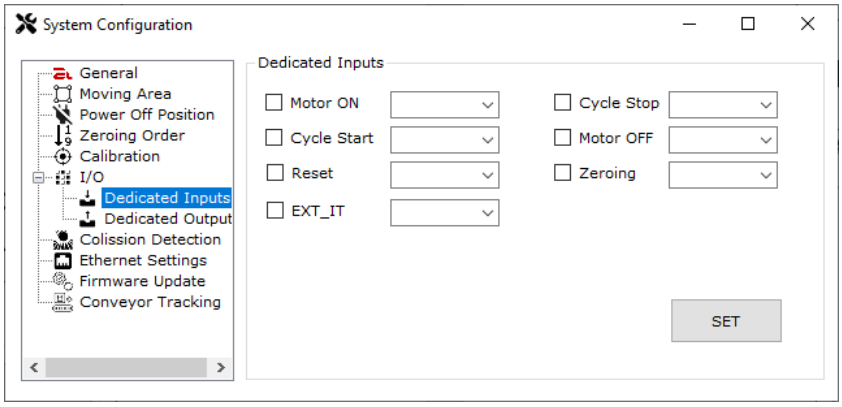


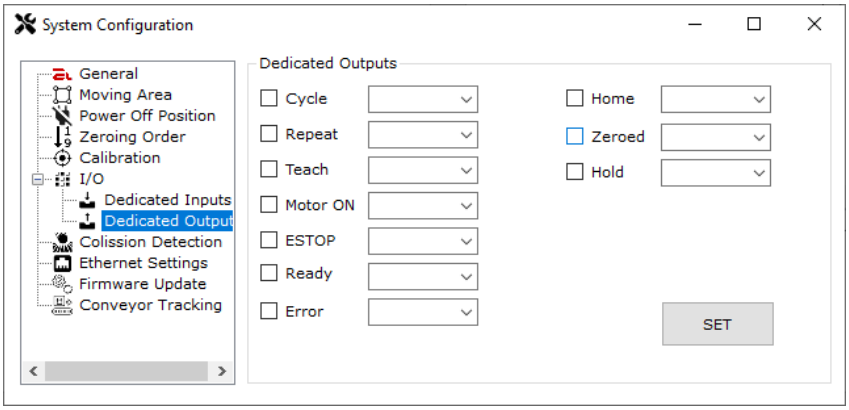
16.6 I/O

This area allows you to deactivate the I/O module or reactivate it.



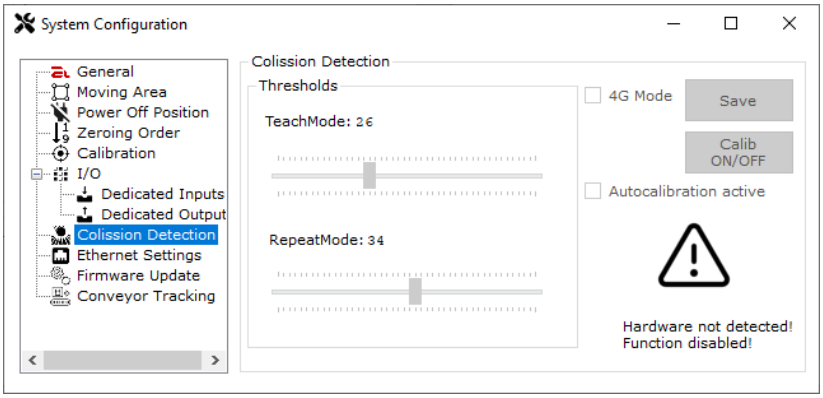
It also allows you to dedicate the dedicated inputs or outputs to perform a predefined function, such as Motor ON





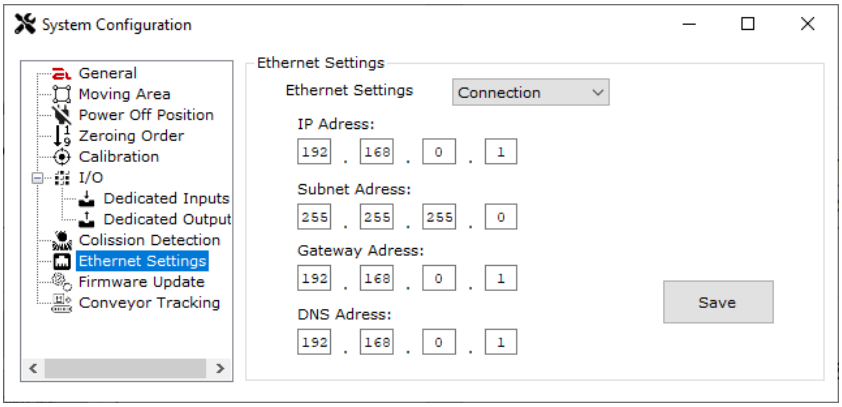
16.7 Colission Detection

This area allows manipulator automatic determination of collision detection thresholds. It also shows that if the unit is equipped with a sensor that allows impact detection.



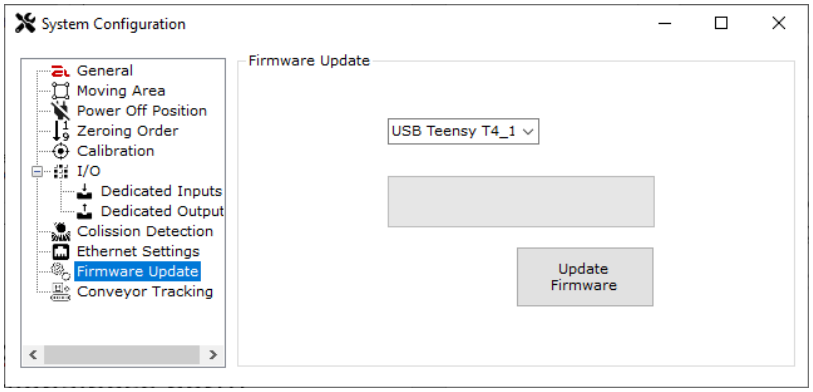
16.8 Ethernet Settings

This area allows you to change the settings of the Ethernet port located in the robot. You can change the network addresses as well as the functionality of the Ethernet port.



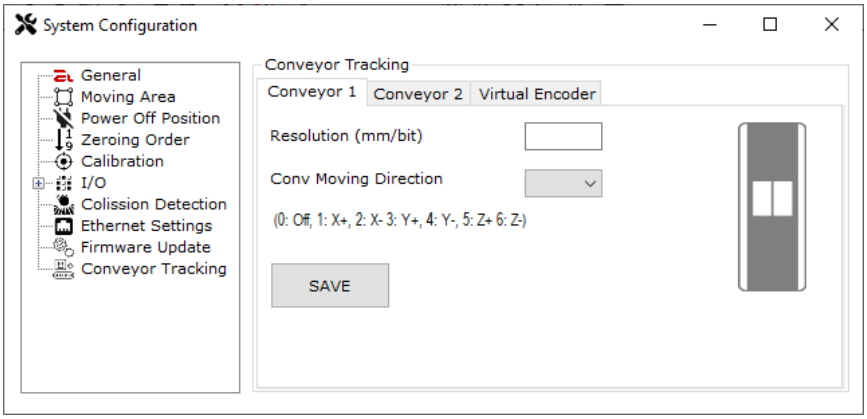
16.9 Firmware Update

This area allows updating the firmware in the robot.

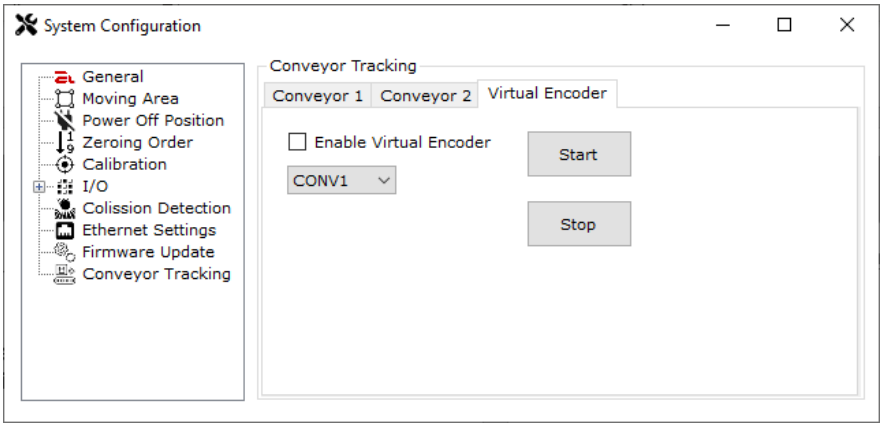


16.10 Conveyor Tracking

This area allows you to modify the conveyor settings, we can set the resolution of the conveyor (mm / bit), as well as the direction of cooperation with the robot.

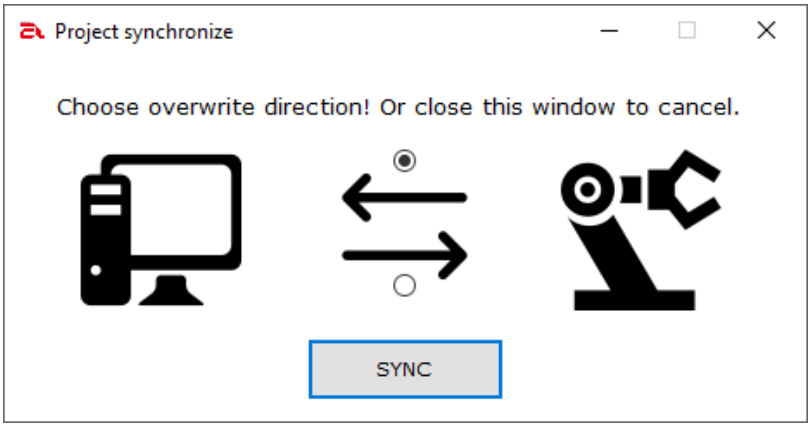


Here you can also turn on or off a virtual encoder, which can be used to simulate applications with a conveyor belt.



17 Synchronization window

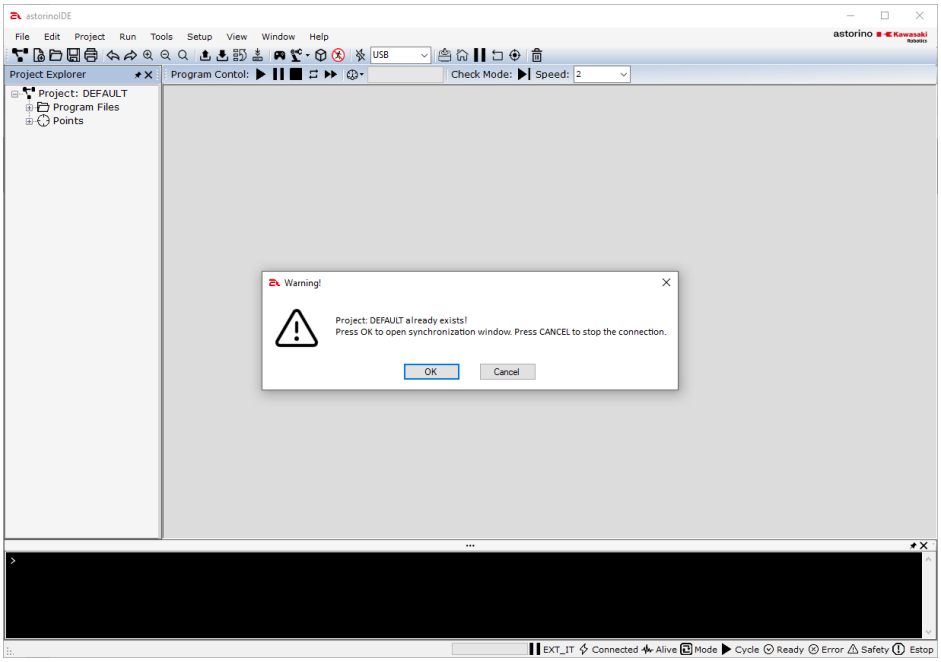
The synchronization window appears when astorinoIDE is connected to the robot. Allows you to choose the direction in which the project data is synchronized.



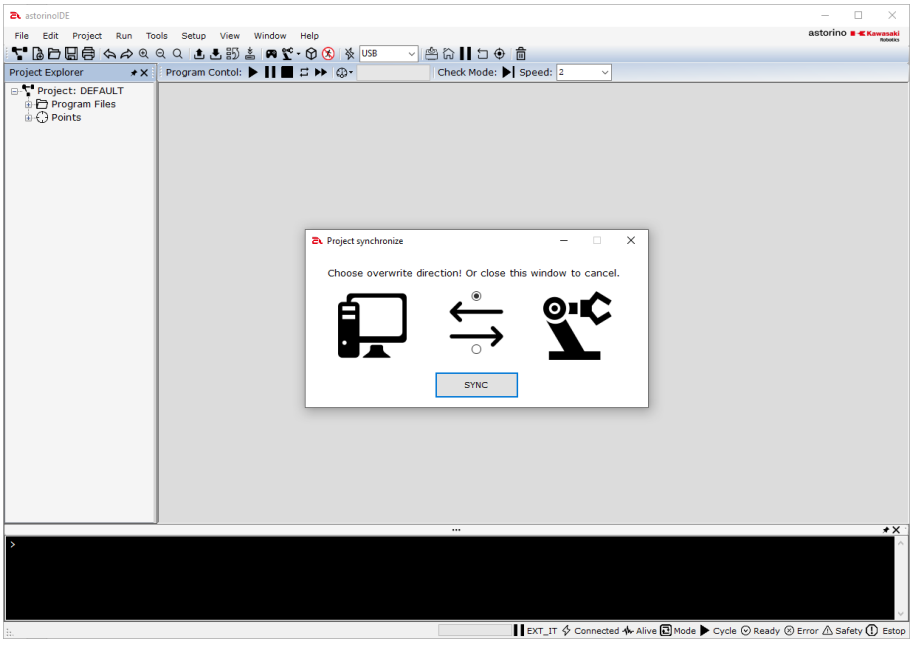
- Synchronization from the robot to the computer overwrites any project data on the computer.
- Synchronization from the computer to the robot overwrites any project data on the robot.

18 Connect and work with your environment

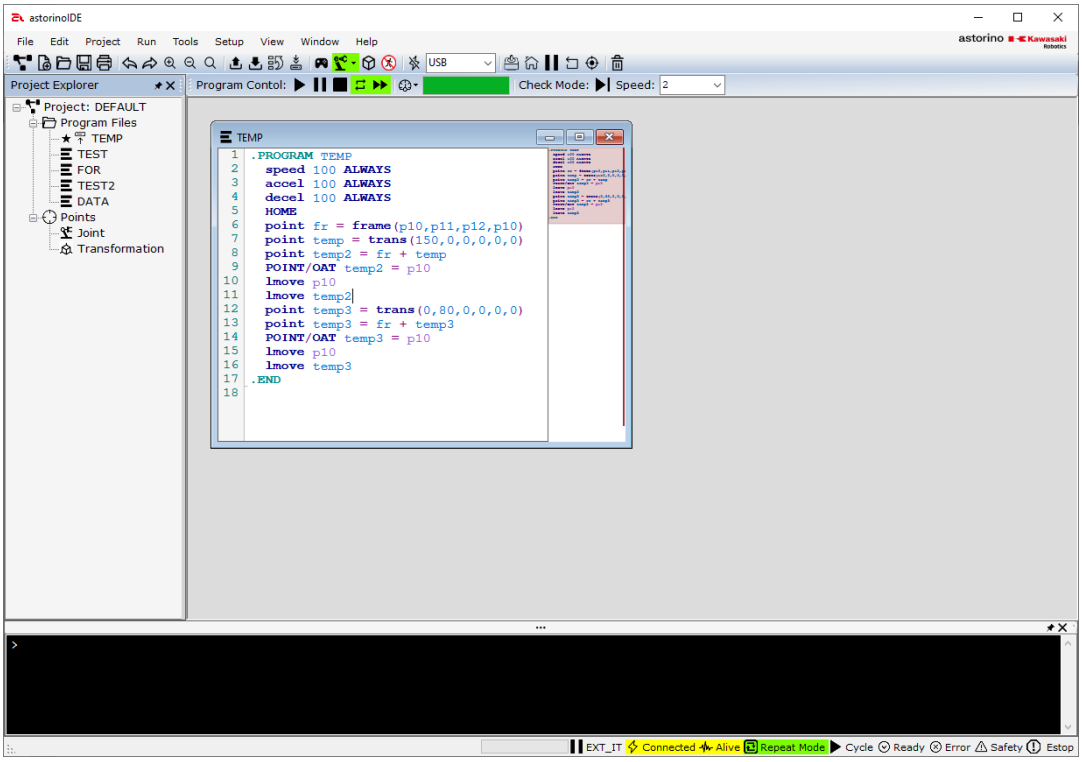
After clicking the [Connect] button, a synchronization window may appear. The program, after finding a project with the same name as the one saved on the robot on the disk in the computer, will open a warning window and ask whether to continue the connection.



After clicking [OK], the synchronization window opens. Select the direction and then the [SYNC] button, closing this window with the 'x' button stops synchronization. The robot will be connected to the program, but the data on the computer and on the robot may differ.



After the synchronization is completed, the selected program that is prepared to be turned on (located in the robot's RAM) will open in the main area



We can start writing programs and using the robot.

19 Manufacturer information

Kawasaki Robotics Astorino
USER MANUAL ASTORINOIDE

January 2024: 3rd edition

Publication: ASTOR and Kawasaki Robotics GmbH
