NEURA ROBOTICS

MAiRA®

Multi-Sensing Intelligent
Robotic Assistant

Software Manual

Edition: v2.1 / 12-2024

Software Version: sw v4.19.1 / ai v3.0.0

Document No.: 100001000014

TABLE OF CONTENTS

1 ABOUT THIS MANUAL

This is the manufacturers' original manual describing the software operation of the product. Please read this manual and make sure you understand the functionality and performance of the product before attempting to use it. Keep this manual in a safe place where it will be available for reference during operation.

1.1 Applicable product models

This manual describes the software operation of the MAiRA cognitive robot (all models).

1.2 Intended audience

This manual is tailored to robotic integrators responsible for setting up and maintaining robotic systems in industrial environments. Intended readers have a foundational knowledge of robotics, factory automation, mechanics and electronics, e.g.:

- Robotic integrators
- Automation engineers
- Maintenance technicians with robotics experience
- System engineers in industrial automation
- Factory automation specialists

The manual provides clear instructions and illustrations to support the effective integration and operation of the robotic system and/or device.

1.3 Further information

Further information can be found on the NEURA Robotics PartnerHub:

https://qrco.de/NEURA_PartnerHub

NEURA PartnerHub

Technical resources in one place.

For setting up your robot, simply scan the QR-code to access quick start guides, user manuals, CAD-files and other technical documents.

1.4 Units

All units are metric unless otherwise noted.

1.5 Safety warnings

⚠ DANGER
Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
Nature or type of hazard
Possible injury or damage
How to avoid such injury or damage

⚠ WARNING
Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION
Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury.

ATTENTION
Indicates a potentially hazardous situation which, if not avoided, could result in damage to or destruction of the product.

1.6 General indications

🗐 Prerequisite	Indicates a prerequisite that must be satisfied before the following action is performed. Example: 🗐 Your teach pendant is powered on
► Action	Indicates a single action. Example: ► Power on the robot
➔ Result	Indicates the result of an action. Example: ➔ The robot is powered on
Bold text	Indicates user interface elements and menu items. Example: ► Tap **Robot**
i Info	Indicates a reference to detailed, supplementary or further information. Example: i Please refer to the robot's software manual for detailed information.

2 ABOUT THE MAIRA SOFTWARE

2.1 Basic principles

The software on MAiRA consists of the following three main components.

Low-level control: This component is responsible for all movements of the robot. It also provides all types of controllers from joint control up to advanced torque control. The low-level control of the robot system ensures the motion properties based on hard real-time control.

High-level control: This component is responsible to read and analyze the sensor data from the robot and decision making on the higher level. The planned decisions will be transferred and executed by the low-level control system. All advanced applications are planned in this component.

Graphical User Interface (GUI): The User interacts with the robot system via the Graphical User Interface. The User Interface offers functionality for commissioning, programming, monitoring and managing the robot system.

All the commands and sensor data go through the GUI to get transferred between low-level and high-level control. The GUI is also responsible for giving a full and real-time update to the user. The user can have full control on any functionality of the robot through the GUI.

2.2 Controlling the MAiRA robot

The Teach Pendant is the tablet on which the graphical user interface (GUI) is shown but not run. All software components including low-level control, high-level control and the GUI are running on computers inside the control box.

The Teach pendant is only one way to access the GUI. If another external PC or tablet is safely connected to the network and the admin user provides all the safety passwords, the output of the GUI can be visualized on that PC to control the robot.

On the GUI, the user can put the robot in **simulation mode** or **real mode**. In simulation, almost all robot functionalities are available, and the user can try any program in simulation before running it on the real robot. In simulation, the visualized robot on the GUI will move completely based on the given command while the real robot will not move or get any command. Note that the sensor data in the simulation mode is read from the sensors on the real robot.

2.3 About Artificial Intelligence (AI)

Artificial Intelligence (AI) is a super simplified version of intelligence that is currently hugely advancing due to novel hardware and software technologies. AI tries to give the agent (here the robot) the ability to perceive the environment human-like by using all types of sensory information including but not limited to microphone arrays and 3D cameras.

MAiRA is equipped with all type of on-board sensors and AI plays a major role to use this information in a similar way as human beings. This makes MAiRA the first cognitive robot to understand the environment, for example by seeing or listening. The unique feature of MAiRA is that all AI calculations and data handling are done and kept inside the control box and there is no need to use cloud computation. Voice Control is one of the AI based functions which makes MAiRA a cognitive manipulator. The user can fully control the robot using the GUI. Most of the GUI functionalities are available to be controlled through voice commanding. Multiple languages are available now, and more are being added.

3 VIRTUAL MACHINE FOR OFFLINE ROBOT SIMULATION

Please follow these instructions to set up a virtual machine (VM) for offline simulation of a MAiRA robot.

The VM is primarily intended to familiarize users with the robot’s graphical user interface (GUI). It can be used to create and run simple programs without being connected to a real robot. Hence there is no real robot connected, not all features of the GUI can be used to their full extent. The user experience is highly hardware dependent as the use of virtualization software is very resource demanding (see system requirements). The use of weak hardware (e.g. CPU performance) and/or false network settings (e.g. firewall configuration) may cause impairments.

The virtual machine is delivered as an image for the open-source virtualization software Oracle VM VirtualBox. Running the VM with other virtualization software could pose problems.

These instructions were created using the Windows 10 operating system (Ubuntu 18/20 can also be used).

3.1 System requirements

System part	Requirement
Hard drive space	At least 30 GB of free space
Computer Memory	At least 10 GB of RAM
Computer Processor	At least 8-12 cores

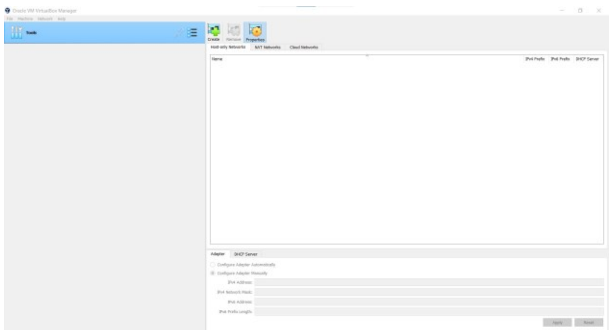
3.2 Installation

- To obtain the virtual machine, visit the link below and download the **MAiRA Virtual Machine** (***.ova**) image file.

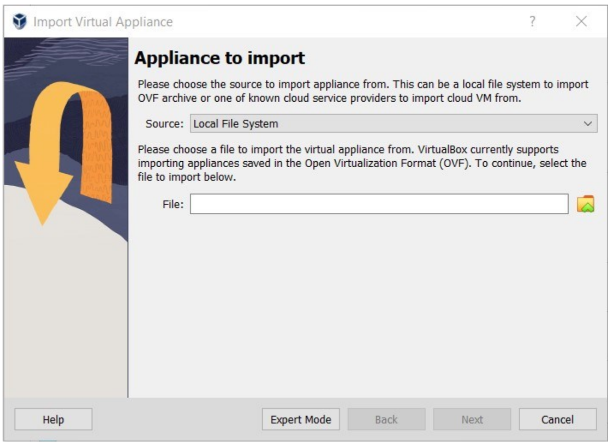
  https://neura-robotics.com/partner-hub

To access the NEURA Robotics PartnerHub, you might need credentials provided by your NEURA Robotics representative.

- Visit https://www.virtualbox.org/ to download and install the **VirtualBox** virtualization software for your operating system.
- Please use the default settings for installation.
- After installation, start **VirtualBox**.
- You now need to import the virtual machine image.
- In **VirtualBox**, click **File > Import Appliance** to import a virtual machine image.



► In the **Appliance to Import** dialog, click the **Select File** button.



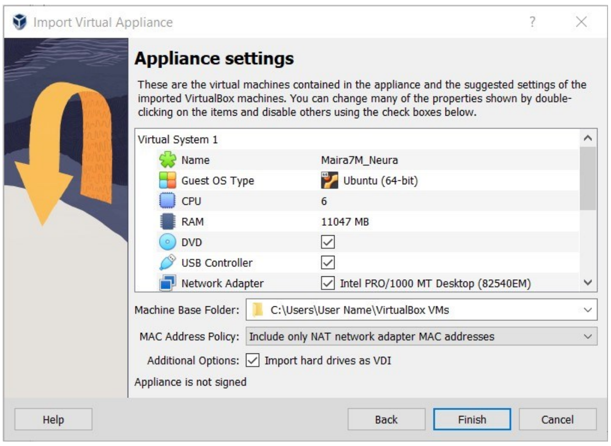
► Use the **File Explorer** dialog to browse to the folder where you just saved the ***.ova** file you downloaded from the NEURA Robotics PartnerHub.

► Select the ***.ova** file and click the **Open** button.

➔ The path to your virtual machine image file is then shown in the **File:** textbox.

► Click the **Next** button and the **Appliance settings** dialog will show up.

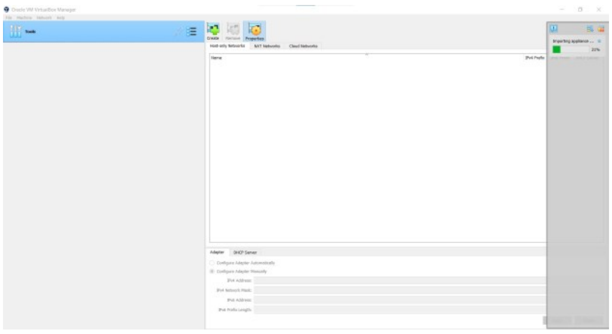
In the **Appliance settings** dialog, your virtual machine can be configured.



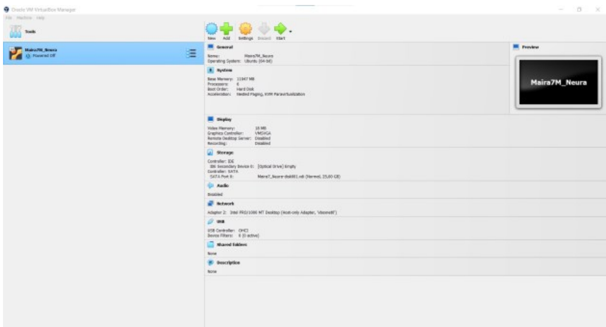
It is highly recommended that you keep the default settings.

► Click the **Finish** button to import the virtual machine image.

► Wait until the import of the virtual machine image is finished (progress is shown in the popup section on the right side of the VirtualBox).



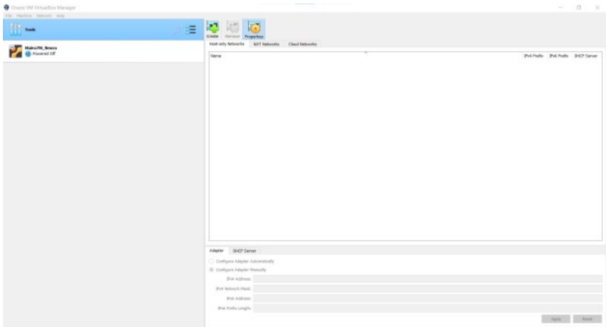
➔ After the import has been finished, the main view of the **VirtualBox Manager** shows your newly installed robot VM.



➔ The installation part is finished.

🗐 Before you can use the VM for robot simulation, you must configure the virtual network adapter by following paragraph 3.3.

3.3 Network configuration

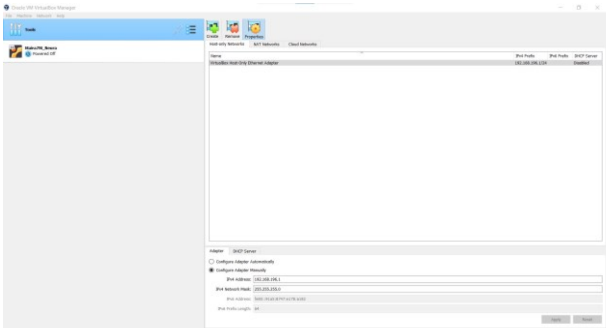


🗐 Before you can use the VM for robot simulation, you must create and configure a virtual network adapter by following the next steps.

► In the main view of the **VirtualBox Manager**, click on **File > Tools Network Manager** .

➔ The **Network Manager** will open.

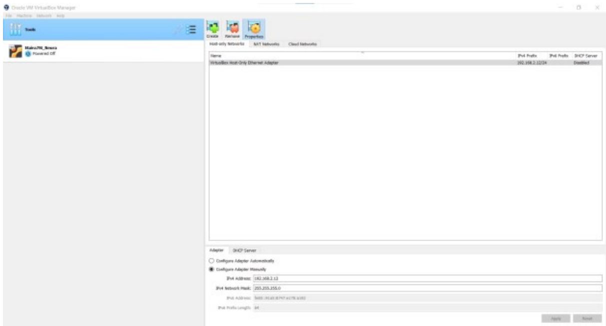
► Click on the **Create** button to create a new network adapter.



➔ An Adapter will be created, the list will show a **VirtualBox Host-Only Ethernet Adapter**.

🗐 This virtual network adapter will be configured in the next steps.

► Select the **VirtualBox Host-Only Ethernet Adapter** and click on the Properties button.

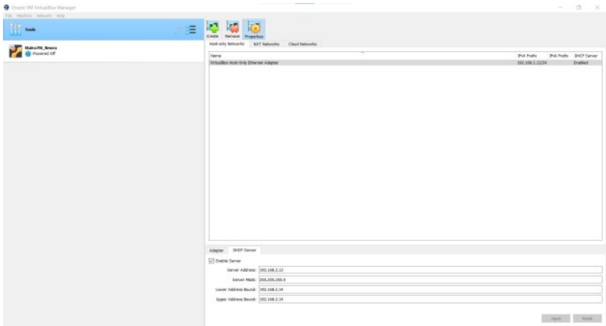


► Click the **Adapter** tab and check **Configure Adapter Manually**.

► Enter the **IPV4 Address** as well as the **IPV4 Network Mask** accordingly:

Setting	Value
IPv4 Address	192.168.2.12
IPv4 Network Mask	255.255.255.0

► Click the **Apply** button and switch to the **DHCP Server** tab.



► In the **DHCP Server** tab, check **Enable Server** and enter the **Server Address**, the **Server Mask** as well as the **Address Bounds** accordingly:

Setting	Value
Server address	192.168.2.13
Server mask	255.255.255.0
Lower address bound	192.168.2.14
Upper address bound	192.168.2.14

► Check the **Enable Server** checkbox.

► Click the **Apply** button and close the **Properties** dialog with the **Close** button.

The newly created network adapter must now be allocated to your virtual machine to enable a connection between your computer and the robot simulation.

► In the main view of the **VirtualBox Manager** on the left-hand topside, right-click on your virtual machine (e.g.: Maira7M_Neura) and click on the **Settings** button.

➔ The **Settings** dialog for your virtual machine will appear.

► On the left-hand side of the **Settings** dialog click on **Network**.

► Open the tab for **Adapter 1** and ensure that **Enable Network Adapter** is unchecked.

► Switch to the tab for **Adapter 2** and check **Enable Network Adapter**.

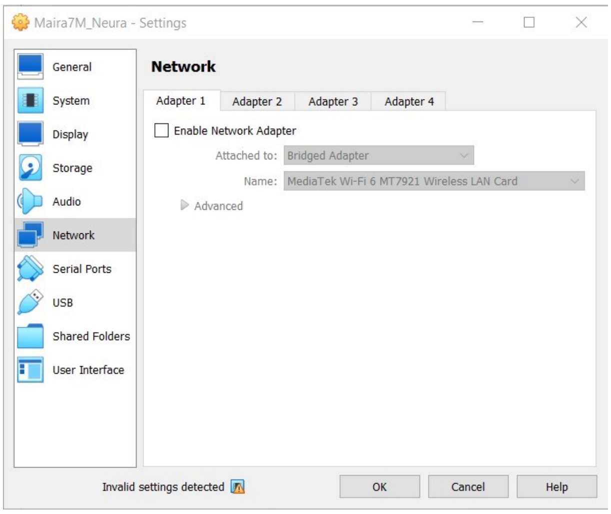
► Use the drop-down menus to configure the adapter accordingly:

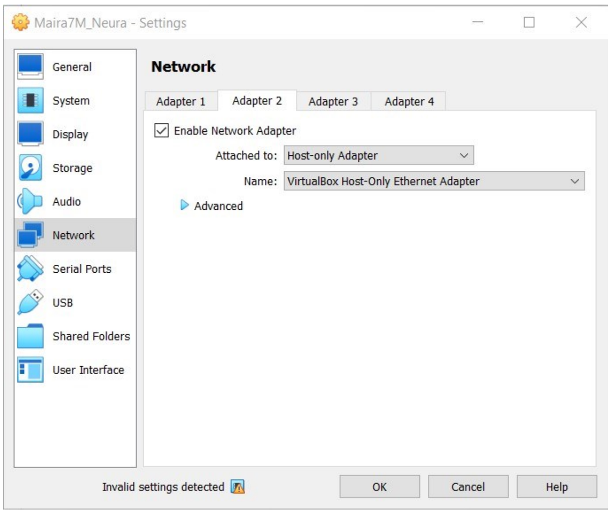
Setting	Value
Attached to	Host-only Adapter
Name	VirtualBox Host-Only Ethernet Adapter

► Click the **OK** button to save your configuration.

➔ The network configuration is now complete.

Paragraph 3.5 explains how to start the virtual machine and connect to the robot simulation.



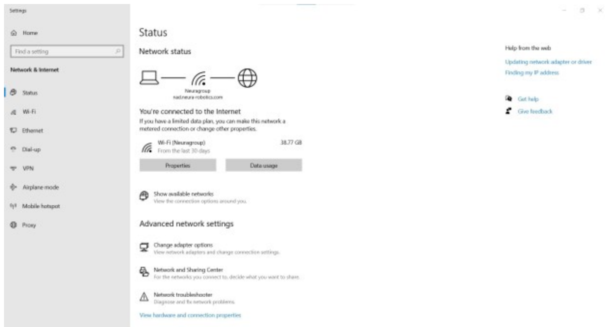


3.4 Troubleshooting

🗐 If you experience problems with the **VirtualBox Host-Only Ethernet Adapter**, please follow this quick procedure which often fixes issues.

► To reset the **VirtualBox Host-Only Ethernet Adapter**, right-click on the **Windows logo** in the bottom-left corner of your Desktop and click on **Network Connections**.

► In the upcoming dialog right under **Advanced network settings**, click on **Change adapter options**.

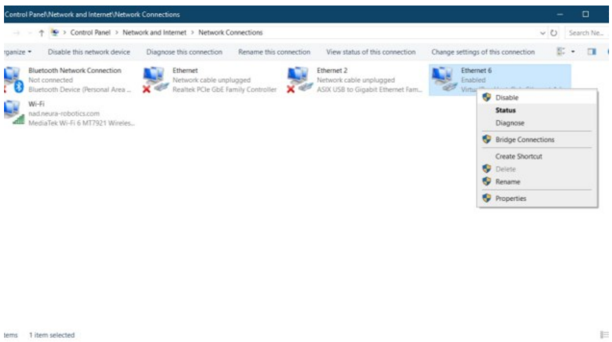


► In the **Network Connections** dialog, right-click on the **VirtualBox Host-Only Network** ethernet adapter and click on **Disable**.

➔ This will deactivate the adapter.

► Repeat the above step, but this time click on **Enable**.

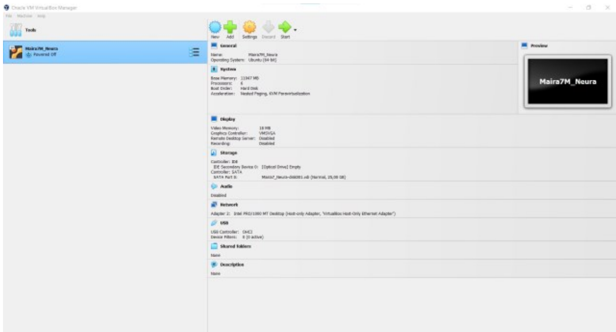
➔ The **VirtualBox Host-Only Ethernet Adapter** has now been reset.



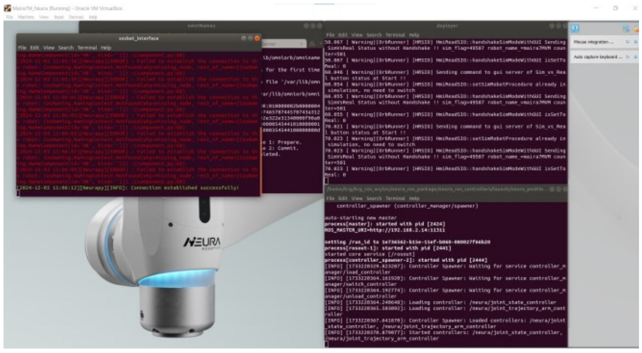
► Please check if your problems with the adapter persist.

3.5 Usage

To use the simulation software, you need to start the virtual machine and connect to the Graphical User Interface (GUI) of the robot with your browser.

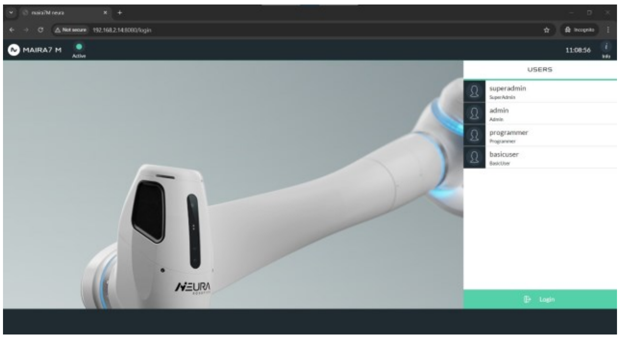


- ► To start the virtual machine, select it in on the left-hand topside in the main view of the **VirtualBox Manager**.
- ► Click the **Start** button to boot the virtual machine containing the robot simulation.
- ► Booting the virtual machine may take some time, so please wait for at least 3 minutes.



- ➔ After the Ubuntu based virtual machine has booted, some terminal windows will pop up with commands being executed.
- ➔ Afterwards the VM is ready for usage.

  Be aware, that you cannot interact with the VM (your mouse cursor will show as a lock icon which can be moved, but you cannot click anything).



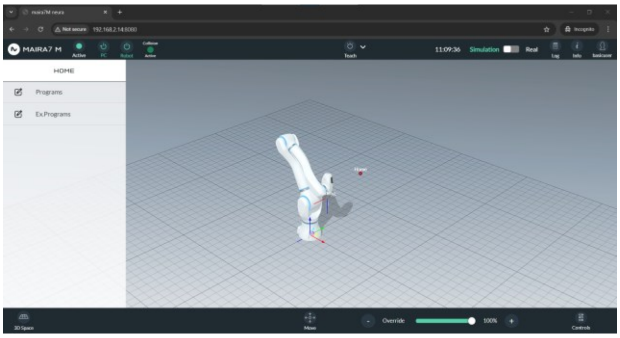
- ➔ To interact with the simulation software, open your browser (**Chrome** or **Edge** are recommended), enter the URL to the robot GUI into the address bar of your browser and load the URL:

Setting	Value
URL	http://192.168.2.14:8080

- ➔ If you experience connection issues, try resetting the **VirtualBox Host-Only Ethernet Adapter** as described in paragraph 3.4.
- ➔ The robot GUI will be shown in your browser tab.
- ► Log in to one of the predefined user profiles.

► Now you can start using the NEURA Robotics simulation software and program your virtual robot.

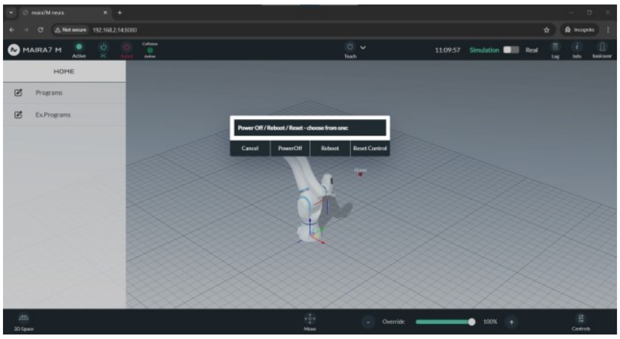
i For instructions on how to do so, please refer to the corresponding topics for the real robot within this manual.



► To close the VM/robot simulation, click on **Robot** in the top bar of the simulation GUI and power of the robot. Then click on **PC** followed by the **PowerOff** button.

➔ This will properly shut down the VM.

► Afterwards you can close the browser window and the **VirtualBox** software.



4 PYTHON API (NEURAPY)

NeuraPy is an easy to use, extendable, robot agnostic Python API (application programming interface) to the NEURA Robotics control software written for application developers. All the functionalities and static parameters of a robot[1] can be accessed as methods and attributes of an instantiation of the **Robot** class.

This paragraph describes the installation process using the Windows 10 operating system (Ubuntu 18/20 can also be used). For further information and code examples, please refer to the **NeuraPy** API documentation.

The software and documentation can be found in the NEURA Robotics PartnerHub by visiting https://neura-robotics.com/partner-hub.
To access the PartnerHub, you will need credentials provided by your NEURA Robotics Business Partner or Project Manager.

There is also a NeuraPyAI version (extended functionality) available for MAiRA robots.

4.1 Installation

- ► Visit the link below to download **Python 3.8** on your Windows machine (Python versions >= 3.6 should work):

  https://www.python.org/ftp/python/3.8.9/python-3.8.9-amd64.exe

- ► Start the installation program, check the **Add Python 3.8 to PATH**[2] box and proceed by clicking **Install Now**.
- ► Close the setup program once the installation is finished.
- ► To install the **NeuraPy** Python API, visit https://neura-robotics.com/partner-hub and download the **Python SDK** zip file for the robot.

🗐 To download the **Python SDK,** you will need credentials provided by your NEURA Robotics Business Partner or Project Manager.

[1] The NeuraPy Python API can be used with real robots as well as with the NEURA Robotics simulation software (see paragraph 2)

[2] By adding programs or libraries to PATH (equals adding them to the Environment Variables in Windows) you can use them from any directory when using the command line without the need to work from their specific directory.

► Extract the downloaded **Python SDK** zip file into a folder of your choice, e.g.: C:\Users\Your Username\NEURA Robotics\Python API.

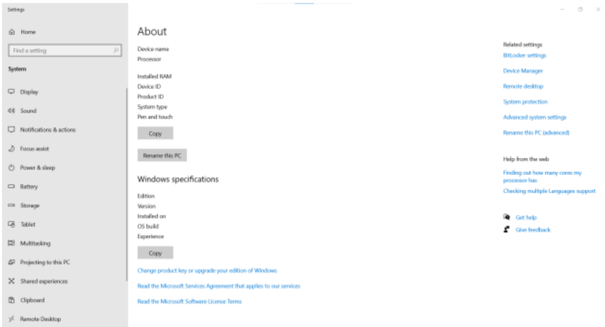
➔ Extracting the zip file will create a new folder called **neurapy** containing a file called **robot.py**.

► For ease of use, the path to the **neurapy** folder should now be added to the **Environment Variables** of your system[2].

► To add the **Environment Variable** for your **NeuraPy** copy, right-click on the **Windows logo** in the bottom-left corner of your Desktop and click on **System**.

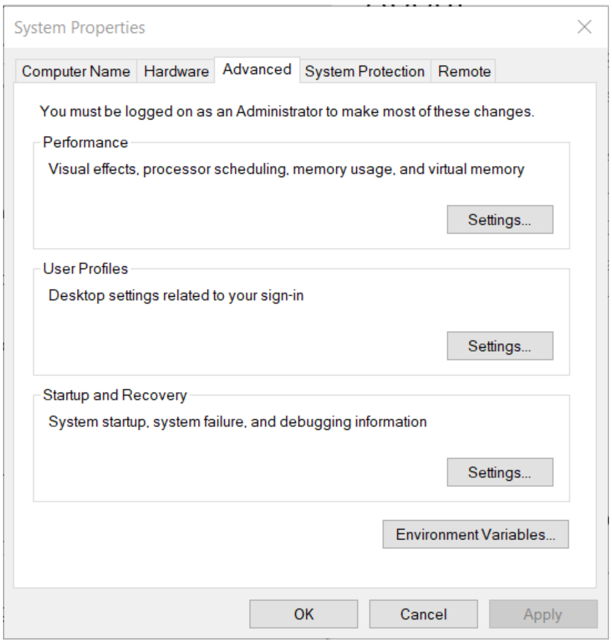
➔ The **About** dialog will open.

► In the **About** dialog, click on **Advanced system settings**.

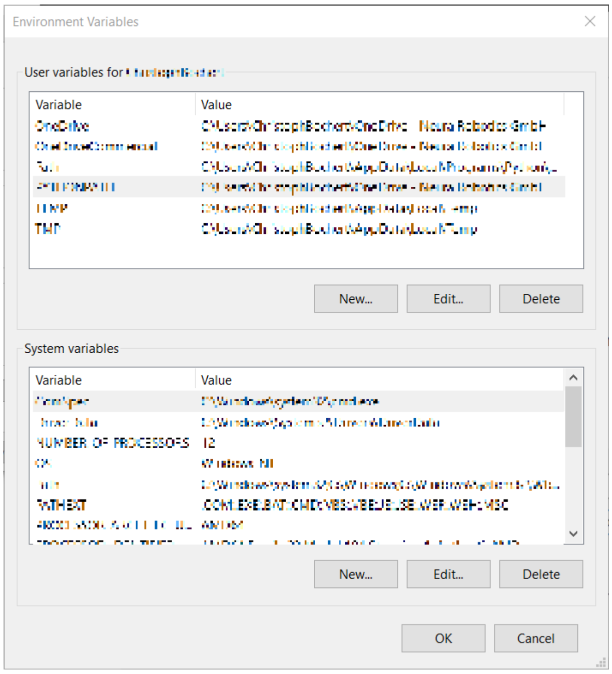


► In the **Advanced system settings** dialog, click on **Environment Variables**.

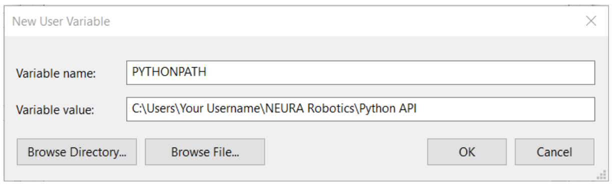
➔ The **Environment Variables** dialog will show with your current variables.



► To define a new **User variable**, please click the **New** button in the top-half of the **Environment Variables** dialog.



► In the upcoming dialog, define the new **User Variable** accordingly:



Setting	Value
Variable name	PYTHONPATH
Variable value	Path to your copy of the **neurapy** folder, e.g.: C:\Users\Your Username\NEURA Robotics\Python API

► Click the **OK** button to save the new **User Variable**.

► Click the **OK** buttons within **the Environment Variables** and the **Advanced system settings** dialogs to exit out of the configuration process.

► Close the **About** dialog with the **X** button in the top-right corner.

➔ They **NeuraPy** API is now installed and ready to be used.

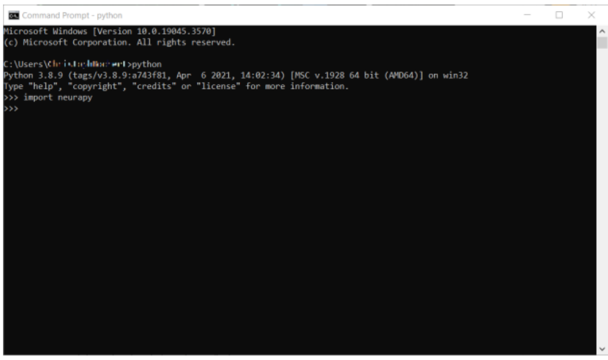
► Please follow the steps in paragraph 4.2 to ensure proper installation.

4.2 Installation testing

► To verify a proper installation of the **NeuraPy** Python API, open a windows command prompt by pressing the **Windows Key** on your keyboard, typing in **cmd** and pressing the **Enter** key.

► Type in **python** and press the **Enter** key.

➔ The **Python command line interpreter** should come up.



► Type in **import neurapy** and press the **Enter** key.

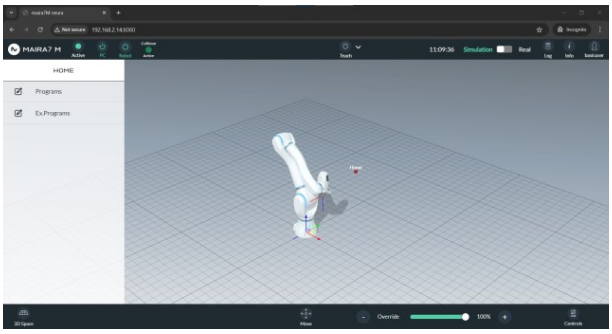
➔ The **NeuraPy** module has now been imported and is ready to be used with the Python interpreter.

If you get a **ModuleNotFoundError** warning, there is a high probability that you didn't configure your **Environment Variables** properly.

► Please repeat the configuration steps above, to verify a proper variable definition.

The Value for the **PYTHONPATH** variable must contain the path to the **neurapy** folder, not the **robot.py** file inside the folder.

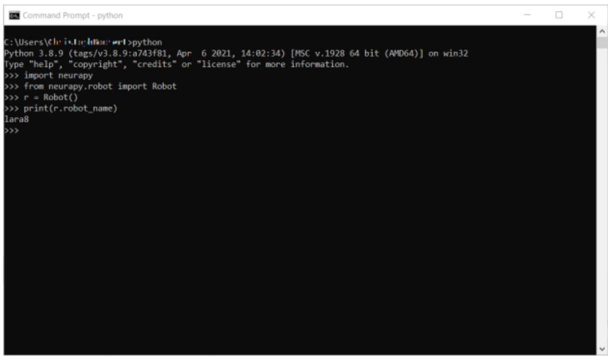
To verify the functionality of the **NeuraPy** installation and for the first steps, the use of the NEURA Robotics simulation software is recommended.



► If not done yet, please follow paragraph 3.2 to install, configure and start the virtual machine containing the robot simulation software.

🗐 The software must be up and running for further API testing.

🗐 Considering the NEURA simulation software and the **Python interpreter** with imported **NeuraPy** API are running, you can now instantiate an object of the **Robot** class.



For initial testing, we will only access a static property of the robot without performing any motion.

For further API usage, please refer to the **NeuraPy** API documentation.

► To access the documentation, please visit https://neura-robotics.com/partner-hub.

🗐 You will need credentials provided by your NEURA Robotics Business Partner or Project Manager.

► Please type in the following commands:
from neurapy.robot import Robot
r = Robot()

print(r.robot_name)

➔ With the above commands, you imported the **Robot** class from the **NeuraPy** API, instantiated an object of the **Robot** class and printed the static **robot_name** property to the command line.

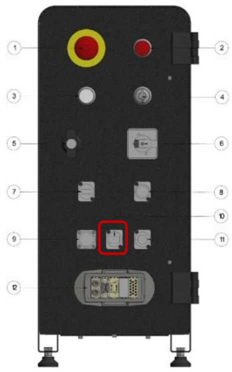
➔ If your **NeuraPy** setup is working properly, the command line should show the **robot_name** property, in this example: **lara8**[3].

➔ The **NeuraPy** Python API is now ready to be used.

4.3 Network configuration

► To use the **NeuraPy** API for controlling a real robot via Ethernet connection, please follow these steps regarding the network configuration.

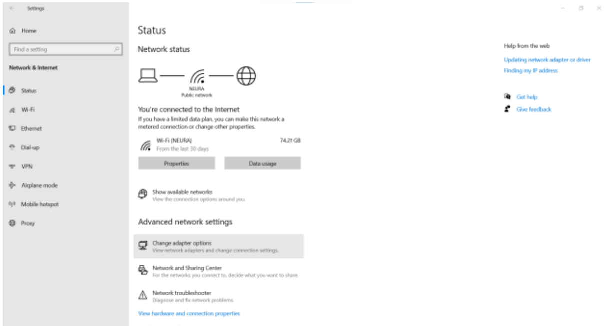
► To connect an Ethernet cable from your computer to the robot controller, please use the Ethernet port in the lower center of the MAiRA control box (10).



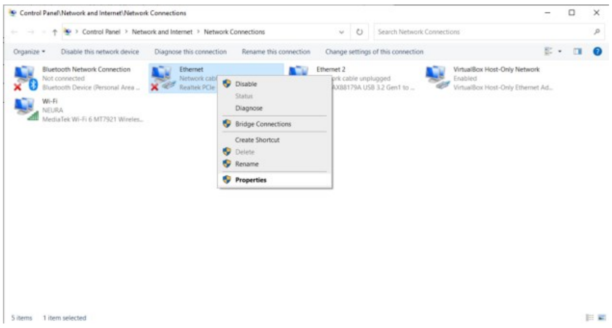
► To configure your network settings, right-click on the **Windows logo** in the bottom-left corner of your Desktop and click on **Network Connections**.

[3] Example for the LARA robot, works the same for the MAiRA robot.

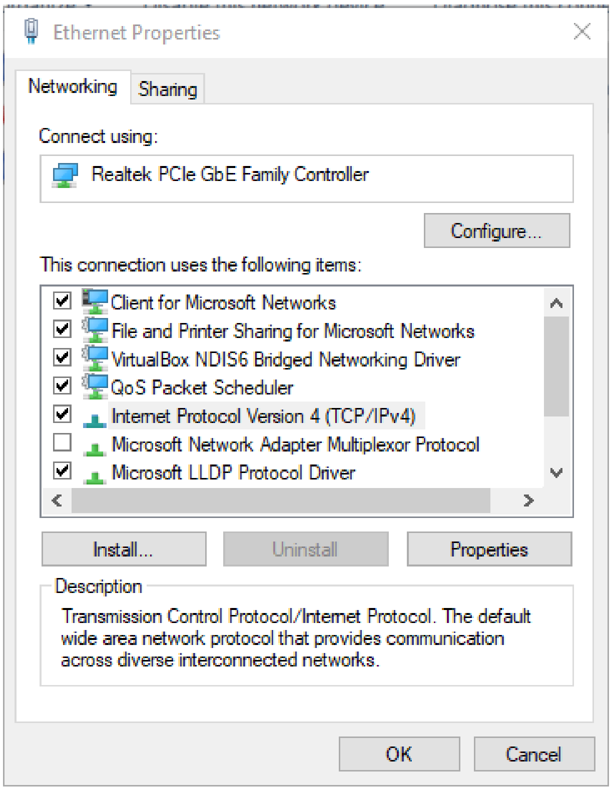
- In the upcoming dialog right under **Advanced network settings**, click on **Change adapter options**.



- In the **Network Connections** dialog, right-click on your ethernet adapter and click on **Properties**.



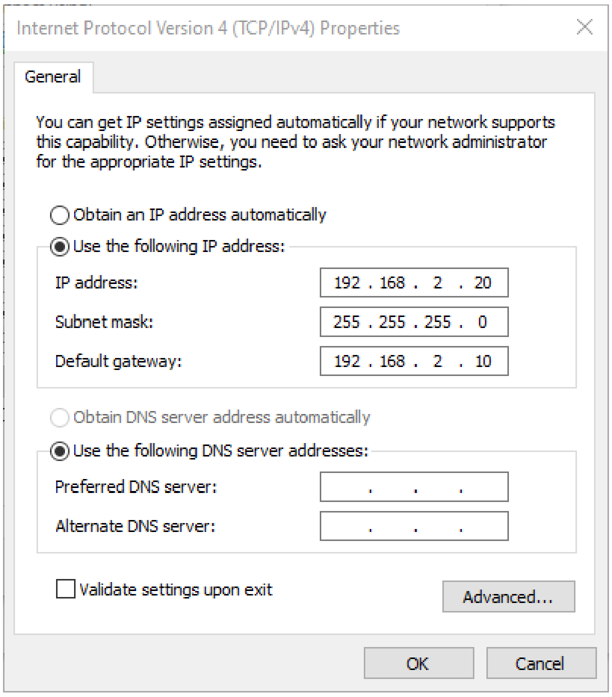
- Within the Ethernet Properties dialog select **Internet Protocol Version 4 (TCP/IPv4)** and click on the **Properties** button.



- In the **IPv4 settings** dialog, check the option field to **Use the following IP address** and configure the IP accordingly:

Setting	Value
IP address	192.168.2.20
Subnet mask	255.255.255.0
Default gateway	192.168.2.10

- Exit out of the configuration windows by clicking the respective **OK** or **X** buttons.



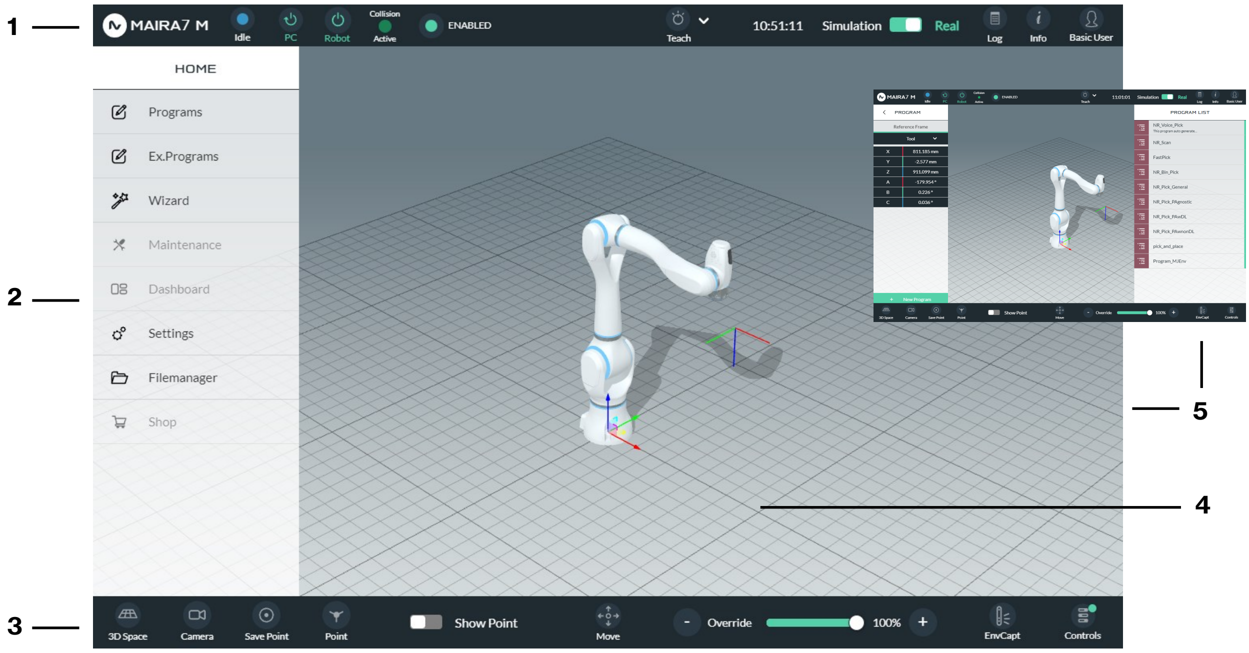
➔ Your network adapter is now configured to use the NeuraPy API for controlling a real robot via Ethernet connection.

5 ROS INTEGRATION

The Robot Operating System (ROS) is a popular Middleware for controlling robots in research and experimental applications. NEURA provides an integration of its products into the ROS ecosystem and a ROS node. Please contact NEURA support if your applications require integrating NEURA Robots into the Robot Operating System. NEURA will provide a Quick Start Guide to enable the control of NEURA robots from the Robot Operating System.

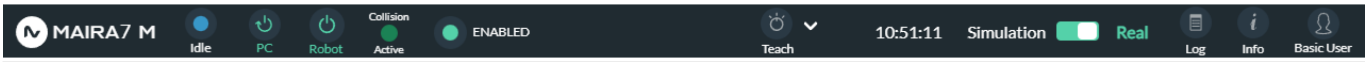
For installing ROS on your computer please refer to the official page: https://wiki.ros.org/ROS/Installation.

6 MAIN PAGE OVERVIEW



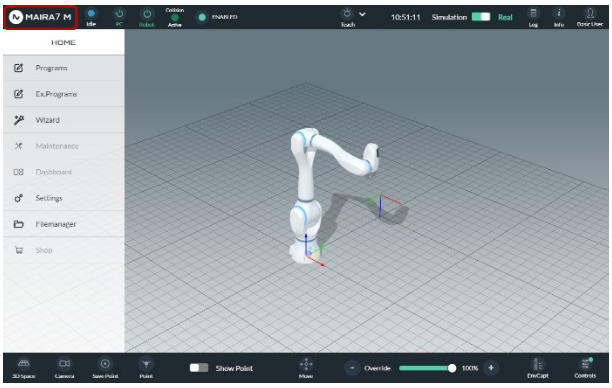
No.	Area	Meaning
1	Top bar	Contains the current robot status, buttons to switch on/off PC and robot, select the robot modes (Teach, Automatic, Semi-Automatic), switch between Simulation and Real mode, open the Log file, show additional info (software version, serial number, ...) and select the user profile.
2	Menu pane	Contains all the operating menus. • Programs • Ex.Programs (External Programs) • Wizard • Maintenance (greyed out, future release) • Dashboard (greyed out, future release) • Settings • Filemanager • Shop (greyed out, future release)
3	Bottom bar	Contains buttons to select a display mode, save, edit, and show points, move the robot, and open the controls.
4	Main window	Shows the robot and the parameters of a selected settings menu.
5	Settings pane	Depending on the selected menu or function, a window appears on the right side of the screen, providing further details and settings (e.g. Program List).

7 TOP BAR ELEMENTS



Element	Meaning	Selections
MAIRA7 M	Robot Model	Reload the GUI
Active	Robot Status	Color indicates robot status Idle — **Blue** — **Idle** Robot stopped Active — **Green** — **Active** Robot moving Emergency Stop — **Red** — **Emergency Stop** Robot stopped immediately
PC	PC operation	**Power Off**: Switch off PC **Reboot**: Reboot PC **Reset Control**: Reset robot control without restarting PC
Robot	Robot operation	Robot — **Red** — **Robot is off** Press to Power On Robot — **Green** — **Robot is on** Press to Power Off
Collision	Collision Status	Collision — **Green** — **Collision detection feature is active** Collision — **Red** — **Collision has been detected**
READY_TO_POWER_ON ENABLED	READY_TO_POWER_ON / ENABLED	READY_TO_POWER_ON — **Orange** — Robot is ready to be powered on (48V power supply must be enabled via the **Enable** button on the control cabinet) ENABLED — **Green** — Robot has already been powered on
Teach	Mode - Teach Indicates the operating mode	**Teach:** Allows manual control of the Robot **Semi-Automatic:** Allows to run a program **Automatic:** Enables the sending of commands from a PC
11:25:23	Time	Displays System Time
Simulation Real	Simulation/Real Select operating mode	**Simulation:** simulate Robot handling **Real:** operate robot in real world
Log	Log	Open log file with warnings and errors
Info	Info	Open Information Center
Basic User	Basic User	Select User Profile **Log out** to switch user profile **Basic User:** with restricted rights **Admin:** with full rights

7.1 Reload the GUI

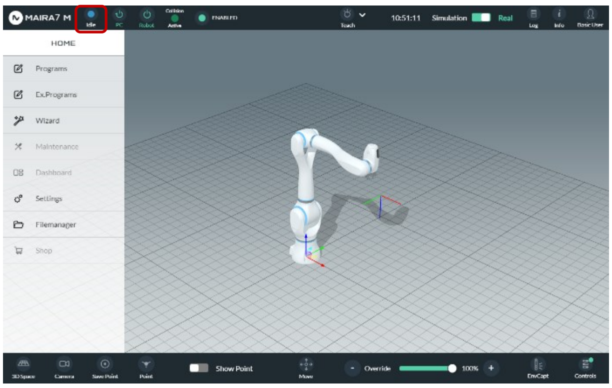


If the user experiences issues with the graphical user interface, it can be reloaded.

1. Press the **MAiRA logo** in the top left corner.
2. Reset the GUI by pressing **Reload**.

NOTICE	It can take up to 30 seconds to refresh the GUI. If the user interface is still not visible, restart the teach pendant.

7.2 Robot Status

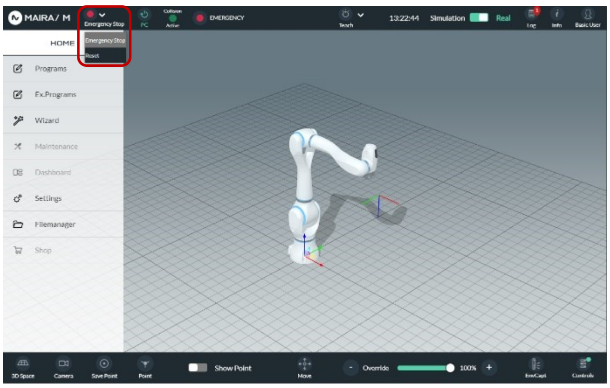


The user can see the robot status by looking at the **colored indicator** in the top left corner.

Colors are defined as follows:

Idle	Blue	Idle: Robot stopped
Active	Green	Active: Robot moving
Emergency Stop	Red	Emergency Stop: Robot stopped immediately

NOTICE	If there has been an emergency stop, there will be a dropdown menu, accessible via the **Status indicator** icon to reset the emergency stop and restart the robot.

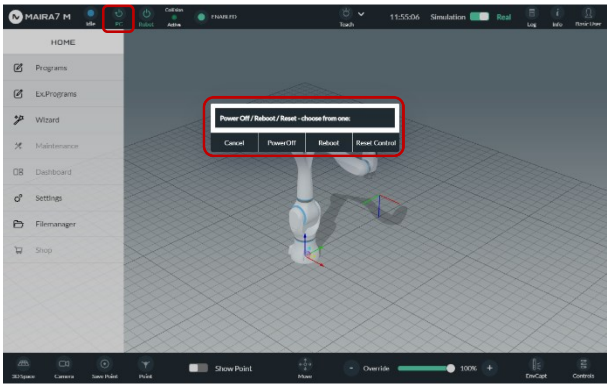


To reset an Emergency Stop:

1. The situation causing the Emergency Stop must be resolved to safely reset the robot.
2. Unlock the actuated **E-Stop switch**.
3. Press the **Enable** button on the controller cabinet.
4. Press the red **Robot Status** icon to open the **Emergency Stop dropdown** menu.
5. Press **Reset.**
6. **Confirm** with yes.

7.3 Power Off/Reboot PC

The user can power off, reboot, or reset the robot controller by pressing the **PC** button in the top left corner.



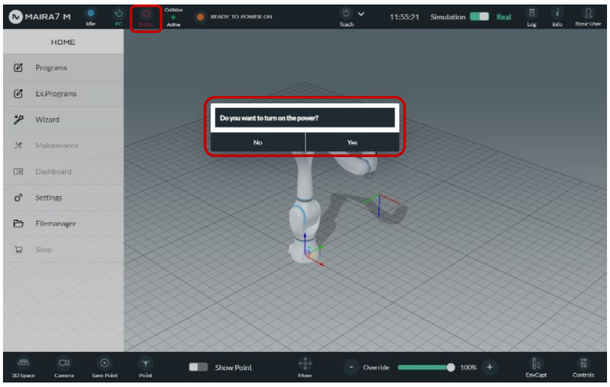
The user can:

1. Shut down the controller PC by pressing **PowerOff**.
2. Reboot the controller PC by pressing **Reboot**.
3. Reset the robot control without restarting the PC by pressing **Reset Control**.

NOTICE	Powering off the robot controller will also power off the robot causing a clicking sound when the brakes get engaged. Rebooting the controller PC might take a short while, please wait for the GUI to be reloaded on the Teach Pendant.

7.4 Power On/Off Robot

The user can power on, respectively power off the robot by pressing **Robot** in the top left corner.



If the **Robot** button is green, the robot is already powered on and can be powered off.

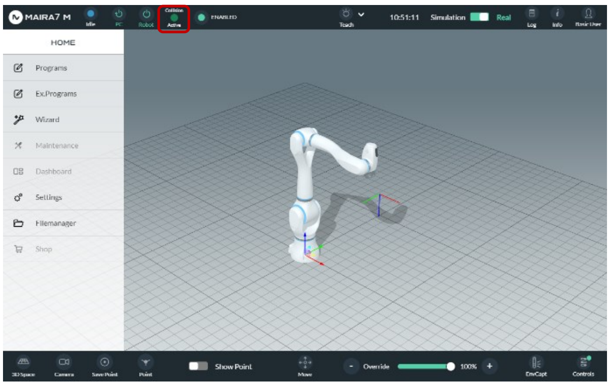
If the button is red, the robot is powered off and can be powered on.
Powering on or off the robot:

1. Press the **Robot** button to power the robot on or off.
2. **Confirm** with yes.

NOTICE	When powering on or off the robot you will hear the brakes releasing or engaging with a clicking sound.

7.5 Collision Status

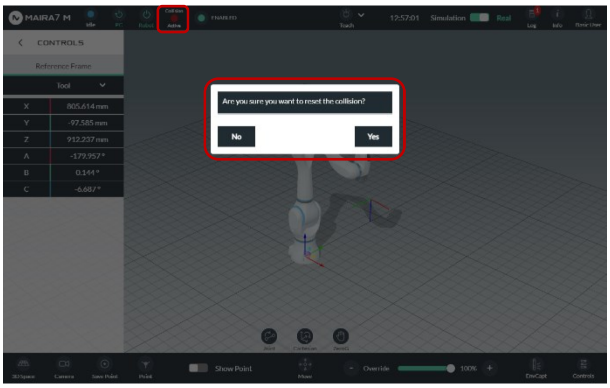
The pulsing green **Collision** symbol in the top left corner indicates that the collision detection feature is active (default setting).



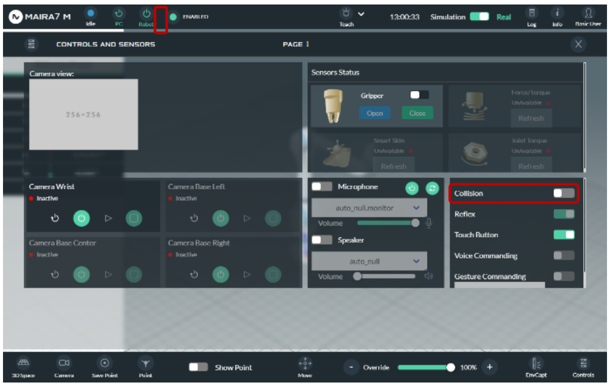
If a collision has been detected, the **Collison** symbol color turns red.

To reset a collision detection (collision detection pauses program execution):

1. Remove the physical cause of the collision if there is one.
2. Press the red **Collision** symbol to reset the collision detection.
3. **Confirm** with yes.



> **⚠ CAUTION**
>
> Collision detection should always be active. But if the robot should be used in a traditional "non" collaborative way, the collision detection can be turned off via the **Controls** settings (administrator privileges necessary).
>
> If the Collision detection has been deactivated, no **Collision** symbol will be shown in the top left corner of the Top bar.



Collision detection automatically pauses program execution.

- ▶ To resume program execution, press the **Play** button
- ▶ To cancel program execution, press the **Stop** button and exit the **Program Launcher**

7.6 Operation Modes

The user can select 3 different operation modes for the robot.

Mode	Description
Teach	Allows manual control of the robot.
SemiAutomatic	Allows continuous or discrete "step-by-step" program execution (see 11.3).
Automatic	Allows running programs automatically. Additionally enables external control of the robot, e.g.: sending commands from a PC.

The current mode is shown underneath the **Mode** icon in the middle section of the Top bar.

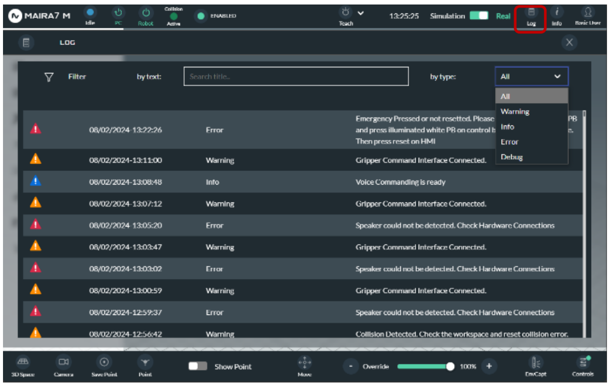
To switch modes:

1. Press the **Mode** icon showing the current mode.
2. **Select** the desired mode from the dropdown menu.

The operation mode is also important for the different safety states of the robot.

7.7 Log File

By pressing **Log** in the top right corner, the user can access the log file of the robot.



The user can search and filter through the history of log entries containing possible Warning, Info, Error and Debug messages.

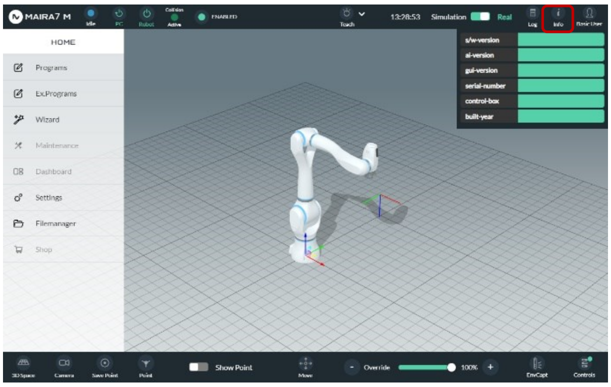
To browse the log file:

1. Press **Log.**
2. Scroll through the list of log entries.
3. Or search for a keyword by typing it into the search box.
4. Or filter the log file by type of message.

7.8 Robot System Information

By pressing **Info** in the top right corner, the user can display some information about the robot system:

Info	Description
s/w-version	Version number of the installed robot control software.
ai-version	Version number of the installed robot AI software.
gui-version	Version number of the installed robot GUI software.
serial-number	Serial number of the robot.
control-box	Serial number of the robot control box
built-year	Year the robot was built.



7.9 User Login/Logout

By pressing the **User** icon in the top right corner, the user can switch accounts.

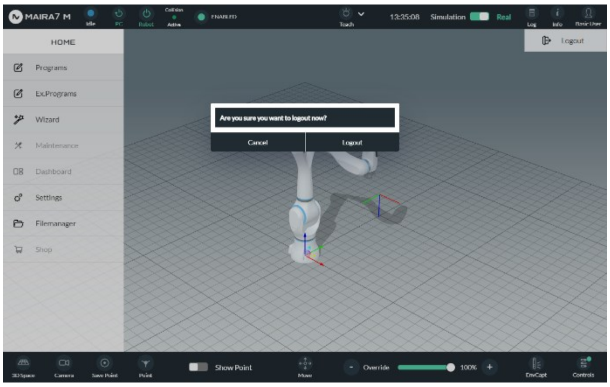
To log out the current user / switch to a different user:

1. Press the **User** icon in the top right corner, then press the appearing **Logout** button to log out the current user.
2. Confirm with pressing **Logout**.
3. After the current user has been logged out, the user is presented with the login screen from where a different user can start the login procedure.

To log in with a different user:

1. Select the user from the list on the right side of the screen.
2. Enter the user Password into the text box.
3. Press the **Login** button in the bottom right corner.

NOTICE	If the tablet display goes black after logging out a user (screensaver), please press the tablet power button shortly, to bring the login screen back up.



7.9.1 User profiles

From software version v4.19.1 you can use the robot with predefined[4] user profiles holding distinct levels of access rights. The user profiles are aimed at different user skills or tasks respectively. The user profiles build upon each other, e.g. the programmer profile includes the basic user privileges. The table below gives an overview of the eligible user profiles.

User profile	Access rights / operator skills	Explanation
basicuser	+	The basic user profile has minimal access rights. It is intended for running existing programs and requires the least user training. The basic user profile is tailored to operators.
programmer	++	The programmer profile lets you create new or edit existing programs. It is aimed at programmers and requires appropriate training.
admin	+++	The admin profile lets you change functional settings. It is intended for e.g. system integrators with expertise in robotics.
superadmin	++++	The superadmin profile has all access rights. It lets you change safety relevant settings. This user profile requires the highest user qualification and is aimed at safety or service technicians.

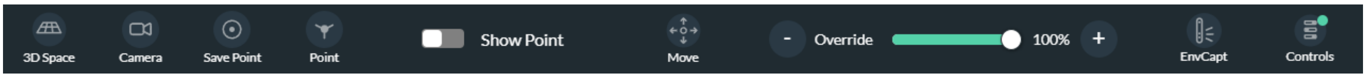
[4] Complete user management to be implemented in future software versions (add and modify users, change passwords, grant rights).

For detailed access rights, see the table below:

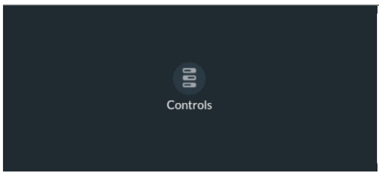
GUI Features	superadmin	admin	programmer	basicuser
General	✓	✓		
Camera Calibration	✓	✓	✓	
Tools	✓	✓	✓	
Global	✓	✓	✓	
• Point	✓	✓	✓	
• Recorded Path	✓	✓	✓	
User	✓	✓		
• View	✓	✓		
• Edit	✓	✓		
• Delete	✓	✓		
HMI Buttons	✓	✓	✓	
Robot	✓			
• Joint	✓	✓		
o Encoder	✓			
o Limits	✓	✓		
o Max Speed	✓	✓		
• Cartesian	✓	✓		
o Workspace	✓	✓		
o Speed	✓	✓		
• Robot Parameters	✓	✓		
o ZeroG	✓	✓		
o Collision	✓	✓		
o Axis Lock	✓	✓		
• Gravity Vector	✓	✓		
• Brakes	✓	✓		

AI	✓	✓		
• Voice	✓	✓	✓	
○ Language Settings	✓	✓		
○ Voice Command	✓	✓		
• Object Detection	✓	✓		
• Pose Estimation	✓	✓	✓	
IO Settings	✓	✓	✓	
Robot Information	✓	✓	✓	
Import/Export	✓	✓		
Software	✓	✓	✓	
Programs	✓	✓	✓	✓
Create/ Delete Programs	✓	✓	✓	
External Sources	✓	✓	✓	
External Programs	✓	✓	✓	✓
Upload / Delete Programs	✓	✓	✓	
Robot Calibration	✓	✓		
DPI	✓	✓		
Interfaces	✓	✓	✓	
Tool Calibration	✓	✓	✓	
Factory Restore	✓			
Control	✓	✓		

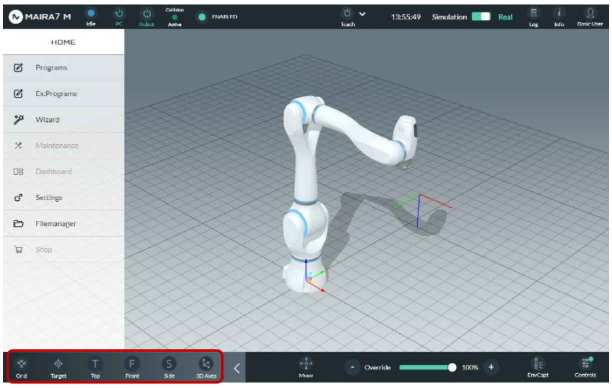
8 BOTTOM BAR ELEMENTS



Item	Meaning	Details / more information
3D Space	Click to expand the menu. Change the view settings of the 3D scene visualization.	**Grid:** switch grid on/off. **Target:** select target view. **Top/Front/Side:** select view direction. **3D axis:** show/hide 3D axis.
Camera	Display camera images in the GUI.	**Wrist**: display the wrist / head camera image. **Left**: display the left (base) camera image (optional camera). **Center:** display the center (base) camera image (optional camera). **Right**: display the right (base) camera image (optional camera).
Save Point	Save a new point.	Save a new point at the current TCP position. Point is stored as joint and cartesian position.
Point	Edit and save a point.	
Show Point	Show / hide points in the 3D scene visualization.	If there are too many points in the scene, it is recommended to turn this option off
Move	Select the move type and set the move parameters.	**Joint:** move each axis. **Cartesian:** move tool in 3D space. **ZeroG:** move robot manually.
- Override 100% +	Scale the speed of the motion of the robot system from 0 – 100 %.	This function is used to test the execution of robot motions with reduced speeds without changing the nominal speed parameters of the program.
EnvCapt	Capture Environment / load a scanned model.	Capture the environment and save as a 3D image or load an existing scan to be visualized in the 3D GUI space.

	Control settings	Activate/deactivate Collision detection, HMI Buttons, External controls, Reflex mode; select between Joint and Torque mode.

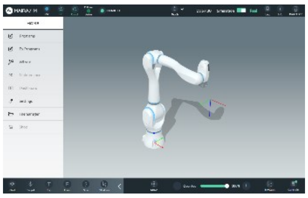
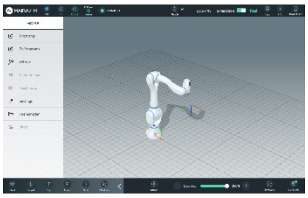
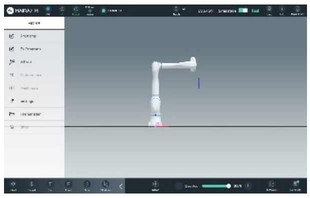
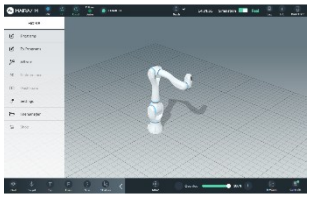
8.1 3D Space



By pressing the **3D Space** button, the user can toggle different view options or switch between predefined views.

To use the **3D Space** functions:

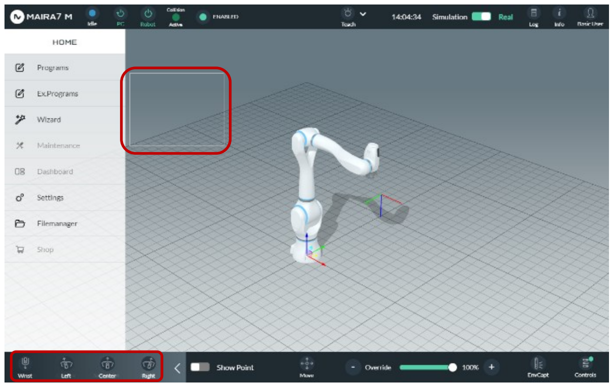
- ▶ Press the **3D Space** button
- ▶ Press the button with the desired function (see table below) in the **3D Space** submenu

Button	Function	
Grid	Show or hide the grid within the main view.	 →Grid turned off
Target	Select Target view.	 →Target view selected
Top	Switch view direction to Top view.	 →Top view selected
Front	Switch view direction to front view.	 →Front view selected
Side	Switch view direction to side view.	 →Side view selected
3D Axes	Show or hide coordinate axes within the main view.	 →3D axes turned off

8.2 Camera

By pressing the **Camera** button, the user can activate/deactivate camera feeds depending on hardware configuration.

- ► Press the **Camera** button
- ► Press on a camera position, e.g. Wrist (default camera) to activate the camera feed
- ➔ Camera feed will be displayed in the rectangular box if available
- ► Press the same camera position again to deactivate the camera feed or press on another position to switch between camera feeds



NOTICE	Camera configuration depends on your hardware setup.

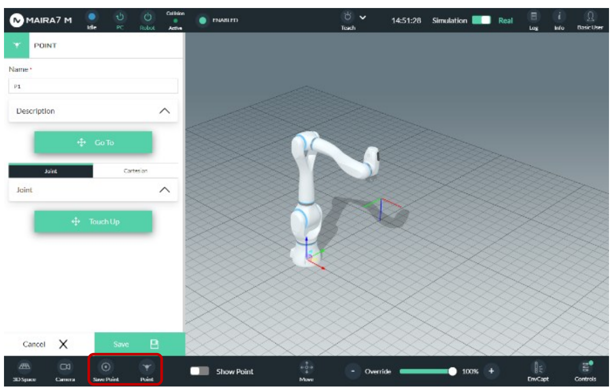
8.3 Point and Save Point

Save Point immediately saves a point at the current robot position. Both the joint configuration and the position / orientation of the robot's TCP are saved. The points will be named automatically by numbering them consecutively (e.g. P1, P2, Pn).

Point allows you to save a point by providing additional information. You can enter a point name and optional description or specify joint or cartesian coordinates.

Save/edit a point by pressing **Point:**

1. Edit **Name** if wanted and add a **Description** if needed.
2. Change coordinates in **Joint** or **Cartesian** mode if wanted. When changing the **Point,** the operator can choose to adjust the joint angles of the point, and subsequently update the cartesian coordinates to match the new joint angles, or vice versa.
3. **Save** stores the point.



Go To

1. Pressing the green **Go To** button initiates a joint motion to the specified joint angles.
2. Continue pressing until the joint position is reached.

Touch Up

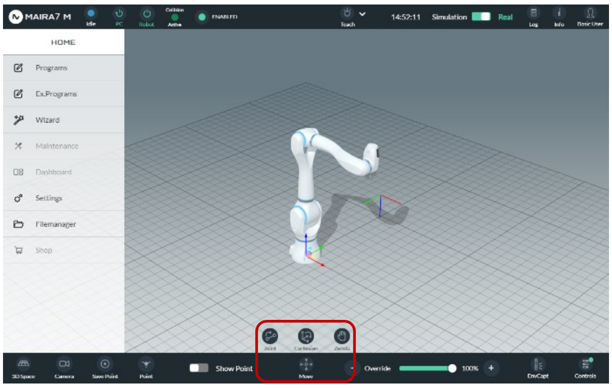
Pressing the **Touch Up** button will overwrite the previously stored values of the point with the current robot configuration.

8.4 Move

Select the move type and set the move parameters. In all move types on the left-hand side the coordinates of the current position are shown.

► Change Reference Frame if needed.

NOTICE	If a tool is selected in the “General settings” the software automatically adapts to the mass of the tool. Be careful with ZeroG if the wrong tool mass is selected. The robot might drift away.



8.4.1 Joint mode

The joint mode enables the user to move separately every single joint of the robot.

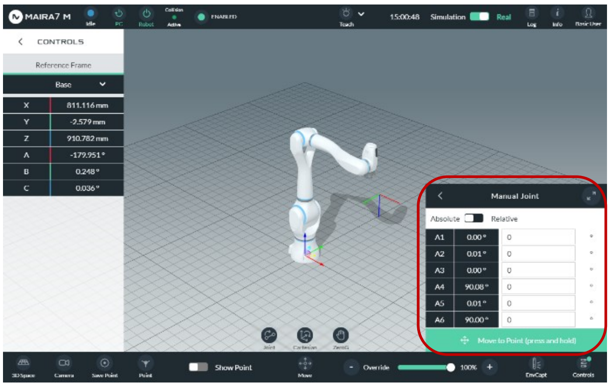
► Use the sliders to move each joint individually.

►Angular Velocity: toggle between °/s and rad/s. The entry field allows you to set the maximum Angular Velocity (limit: 30 °/s)

Click the icon to open the **JOG MODE** menu (see chapter 8.5).

Manual Joint

► Select **Manual** button on the right-hand side to move robot manually.



1. Select if angles are **Absolute** or **Relative**

Absolute	The absolute position is based on the actual position of the robot in the reference frame.
Relative	Relative is the measurement of how far the robot will move relative to its starting joint position.

2. Type in the wanted angles [°].

3. Tap and hold the **Move to Point** button to move the robot. (Note: keep finger steadily on the button for the robot to move)

8.4.2 Cartesian mode

Move the robot in the cartesian space.

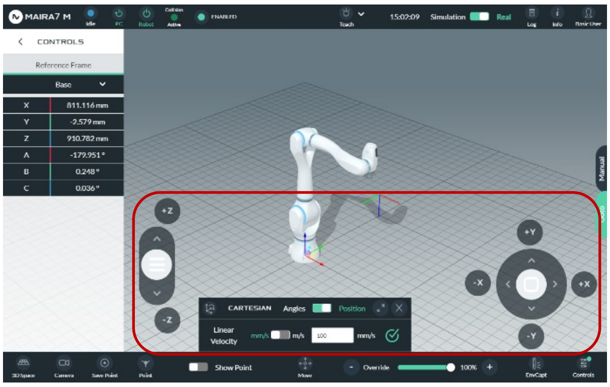
Choose an option:

Position

Move the robot on a linear path in the xyz axis.

► Choose mm/s or m/s.

► Set maximum Linear Velocity.



Angles

Rotate the TCP around the xyz axis.

► Choose °/s or rad/s.

► Set maximum rotation speed.

► **A B** and **C** specify the angular velocity around the x-, y-, and z-axes in the specified **Reference Frame**.

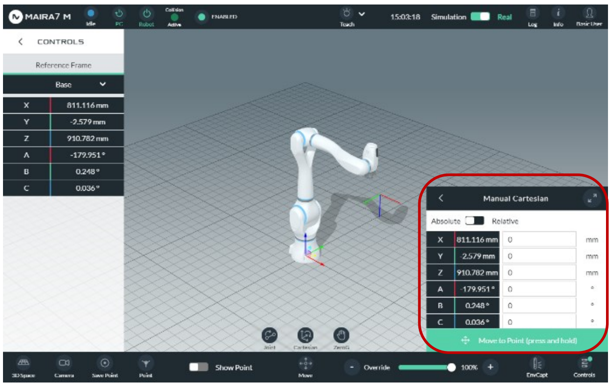
Click icon to open the **JOG MODE** menu (see chapter 8.5).

Manual Cartesian

► Select **Manual** button on the right-hand side to move robot manually.

1. Select if angles are **Absolute** or **Relative**

Absolute	The robot will move to exactly the position that is written into the fields. Example: X= 100 mm, Y = 300 mm, Z =150 mm Robot will try to reach exactly this point.
Relative	The robot will move the number of millimeters that are written into the fields. The robot will move relative to its starting cartesian position in base coordinates. Example: Robot is at position: X= 100 mm, Y = 300 mm, Z =150 mm Operator types into the fields X = 200 mm, Y = 0 mm, Z = 50 mm The robot will end up in cartesian position: X= 300 mm, Y= 300 mm, Z=200 mm



2. Type in the wanted coordinates [mm].
3. Tap and hold the **Move to Point** button to move the robot.

8.4.3 ZeroG mode

In **ZeroG** mode, you can move the robot manually by hand guiding it. Additionally, it's possible to record a movement and play it back inside a program.

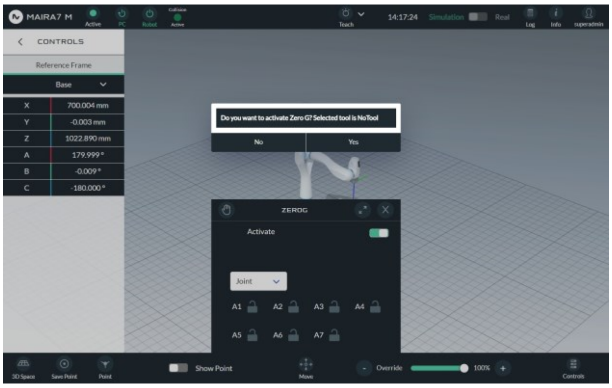
⚠ WARNING

The robot may behave unexpectedly as it needs to know the weight and other parameters of a tool mounted to its flange. Depending on the misinformation, the robot may drift or drop.

- ► Before turning on **ZeroG** mode, verify that:
  - a. DPI has been performed properly (see 12.1)
  - b. The tool is properly set up (see 15.2 Tools)
  - c. The gravity vector is properly set up (see 15.9.4)
- ► When activating **ZeroG**, keep your hands on the robot to prevent it from unexpectedly drifting/dropping.
- ► Prepare to deactivate **ZeroG** or to press the E-stop in case unwanted motion occurs.

► To hand guide the robot, activate ZeroG

NOTICE	Depending on your safety configuration, it might be necessary to keep the yellow dead man switch on the Teach Pendant pressed in middle position to use **ZeroG** mode. Keep it pressed before activating **ZeroG** and as long as moving the robot in **ZeroG**.



NOTICE	**ZeroG** can also be activated/deactivated by using the HMI buttons on the robot's head (if HMI buttons are activated and configured accordingly, see 15.8).

To record a movement:

🗐 ZeroG must be activated

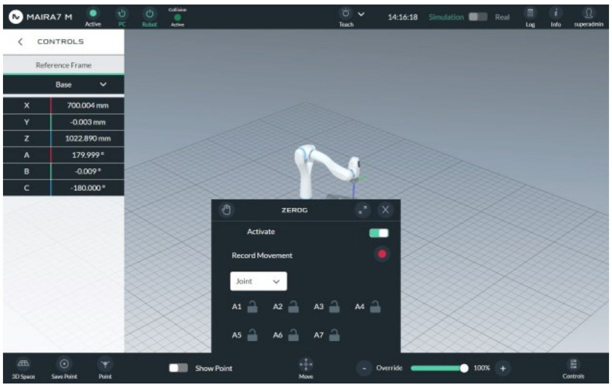
► Select **Record Movement** to start recording..

► Select **Stop** to stop the recording.

► Once stopped, select **Edit** to edit name for recorded path.

► **Save** recorded path.

► To execute the recorded path, create a new program with a **MoveRecordedPath.**



NOTICE	If ZeroG is active, jogging or starting a program is not possible.

8.4.3.1 Joint and cartesian axes lock

When you teach the robot in Zero G mode, you can restrict the robot's movements by locking one or multiple axes.

► Tap Move > ZeroG

► Select the type of axes lock from the dropdown menu:

- Joint: select joint to lock any of the robot's axes A1, A2, A3, A4, A5, A6, A7 depending on your type of robot.
- Cartesian: select cartesian to lock any of the robot's translational X, Y, Z or rotational A, B, C movement directions.

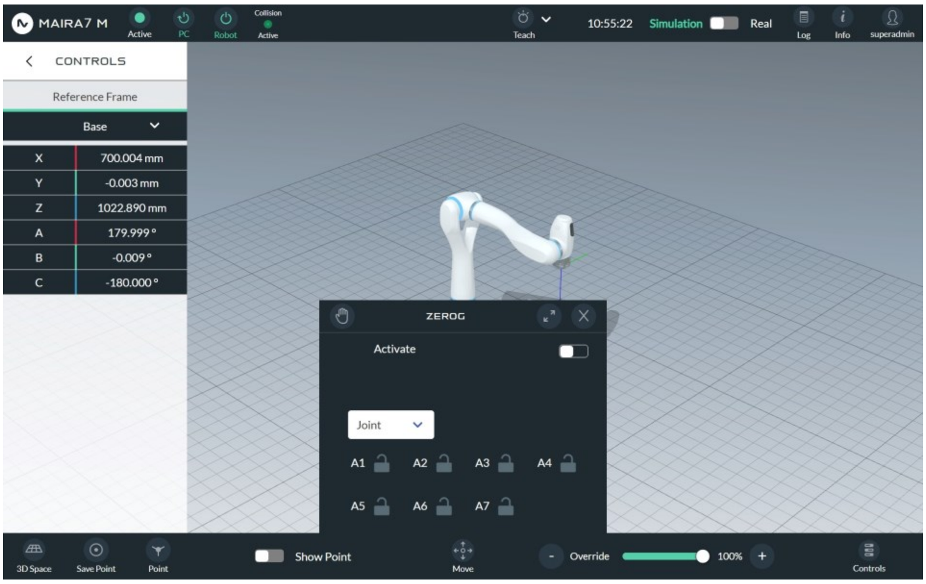
► Tap the open lock icon of one or multiple axes to lock them for movement.

► Tap the closed lock icon to unlock locked axes.

► Activate ZeroG to move the robot following your axes lock restrictions

Note on locking axes:

- You can only use one type of axes lock, either joint or cartesian.
- You can lock any combination of axes for the selected type of axes lock.
- You can lock axes with ZeroG activated, but only when the robot is not moving.



8.5 Jog mode

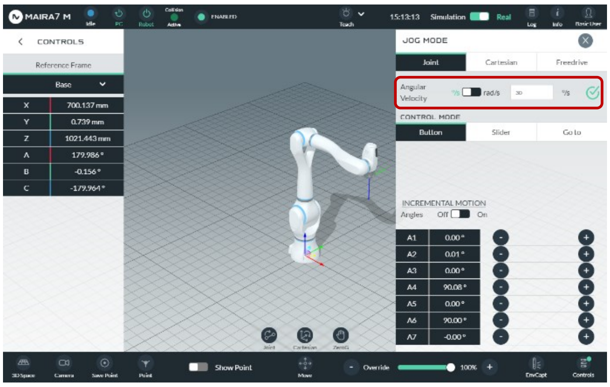
From the **Move** menus **Joint**, **Cartesian** and **ZeroG** the **Jog mode** can be accessed. In the Jog mode further options are available for each Move type.

How to access Jog mode

1. Open one of the Move modes in the bottom bar.
2. Select the Jog mode button (see red rectangle). The Jog mode window opens on the right-hand side of the main window.

8.5.1 Joint

8.5.1.1 Angular Velocity



► Toggle between °/s and rad/s. The entry field allows you to set the maximum Angular Velocity (limit: 30 °/s).

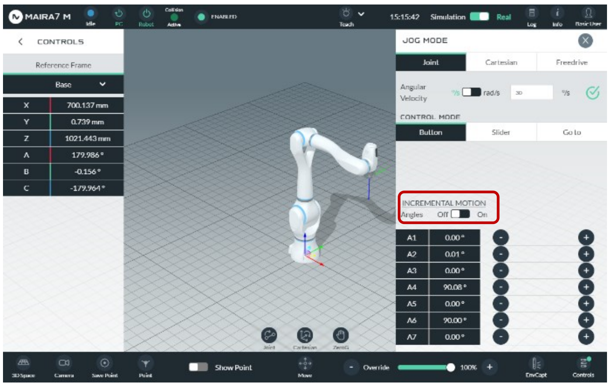
8.5.1.2 Button

INCREMENTAL MOTION deactivated (**Angles** Off)

► Use the buttons to move each joint individually.

► Joints continue moving as long as buttons are pressed or till reaching joint range limits.

► Motions will be performed with the specified angular velocity (see 8.4.1.1) considering globally set speed limits.

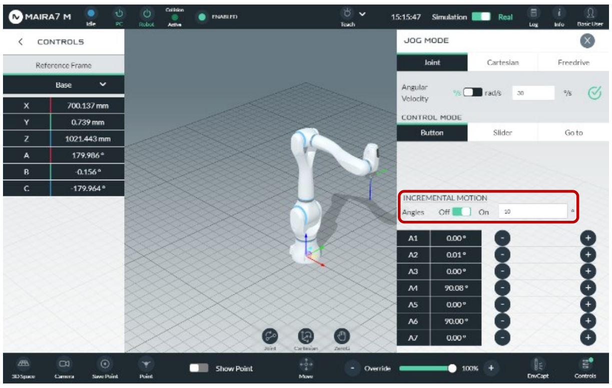


INCREMENTAL MOTION activated (**Angles** On)

► Specify the joint angle increment by which the robot should move (degree or rad, depending on the setting for angular velocity, see 8.5.1.1).

►Keeping a button pressed will move the respective joint until the specified increment is reached (or a joint limit is exceeded). Releasing the button press during motion will immediately stop the joint for safety reasons.

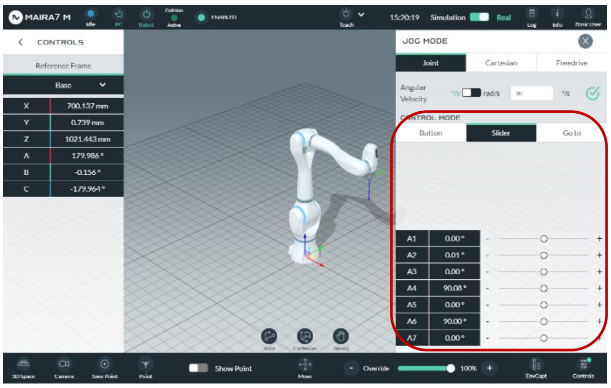
► Motions will be performed with the specified angular velocity (see 8.5.1.1) considering globally set speed limits.



8.5.1.3 Slider

► Use the sliders to move each joint individually.

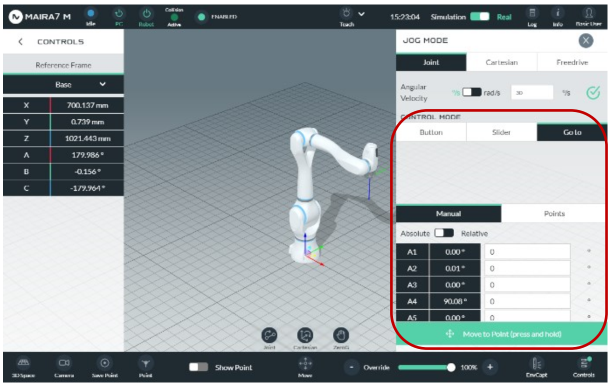
► The motion will be performed with the specified maximum velocity in each direction.



8.5.1.4 Go to

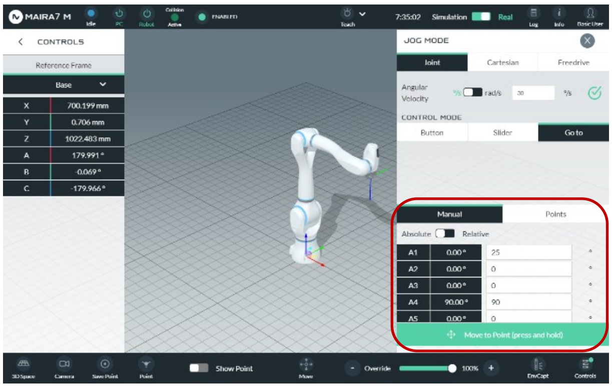
Manual

1. Set movement to **Absolute** or **Relative** mode.
2. Enter a value for the joint to be moved.
3. Press and hold the **Move to Point** button to move the robot.
4. Monitor the movement of the robot carefully during the execution of the movement and stop the motion by releasing the **Move to Point** button whenever a human, a machine is in danger, or if the robot motion does not conform to your expectation.
5. Note that not all points can be reached, e.g., when the point lies outside the joint range.



Example for absolute manual movement (move A1 by + 25 °):

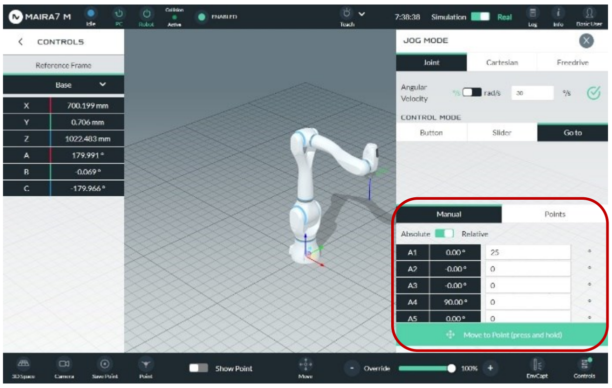
1. Set movement to **Absolute** mode.
2. Type the current position values for the joints not to be moved into the corresponding fields.
3. Enter the position value for A1 = 25 °
4. Press and hold the **Move to Point** button to move the robot. In this example A1 will end up at exactly 25°.
5. Monitor the movement of the robot carefully during the execution of the movement and stop the motion by releasing the **Move to Point** button whenever a human, a machine is in danger, or if the robot motion does not conform to your expectation.



NOTICE	Note that not all points can be reached, e.g., when the point lies outside the joint range.

Example for relative manual movement (move A1 by + 25 °):

1. Set movement to **Relative** mode.
2. Enter the relative movement value for A1. In this case 25 °.
3. The values for the other axes remain at 0 °, they should not be moved in this example; A1 current position is 35°.

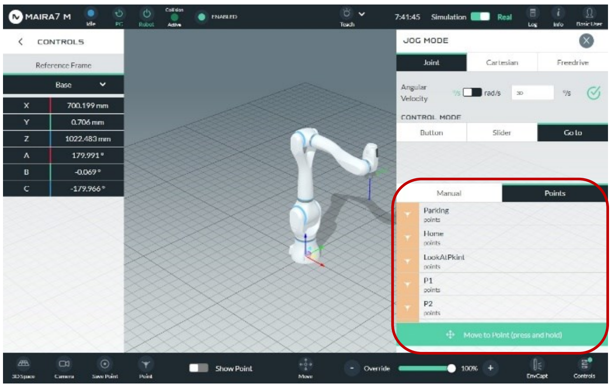


4. Press and hold the **Move to Point** button to move the robot. When reached, the new position of A1 is 60°.
5. Monitor the movement of the robot carefully during the execution of the movement and stop the motion by releasing the **Move to Point** button whenever a human, a machine is in danger, or if the robot motion does not conform to your expectation.

NOTICE	Note that not all points can be reached, e.g., when the point lies outside the joint range.

Points

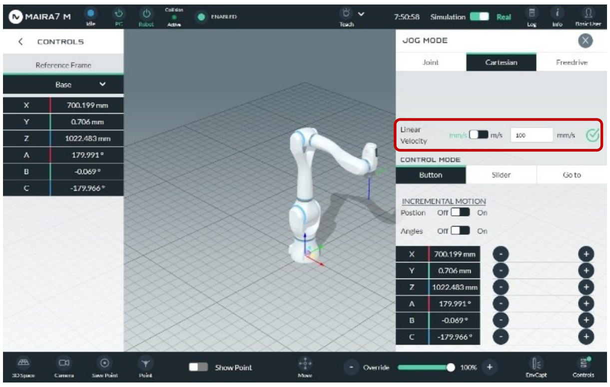
1. Select a point from the list.
2. Press and hold the **Move to Point** button to move the robot to the selected point.
3. The robot will go to the specified point joint coordinates.
4. Monitor the movement of the robot carefully during the execution of the movement and stop the motion by releasing the **Move to Point** button whenever a human, a machine is in danger, or if the robot motion does not conform to your expectation.



8.5.2 Cartesian

8.5.2.1 Linear Velocity

► Toggle between mm/s and m/s. The entry field allows you to set the maximum Linear Velocity (limit: 250mm/s).

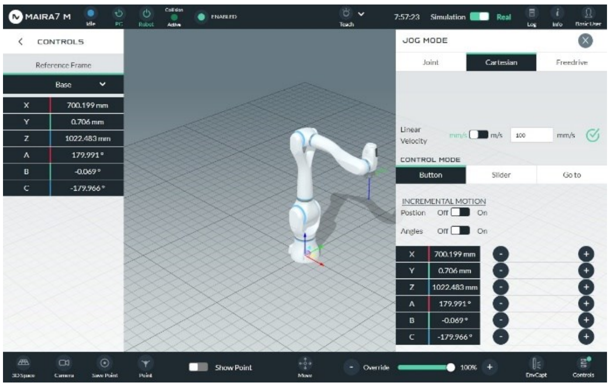


8.5.2.2 Button

INCREMENTAL MOTION deactivated (**Position** Off, **Angles** Off)

► Use the buttons to move the robot's TCP in any direction.

► Pressing the buttons will increase or decrease the TCP position. The parameters **X Y Z** are the translatory position of the TCP (x-, y-, z-coordinates), the parameters **A B C** are the **roll pitch and yaw angles** (rotation around the x-, y-, z-axes) of the current TCP configuration.



► Note that the values are given in the Reference Frame that is currently selected (Base/ World/ Tool).

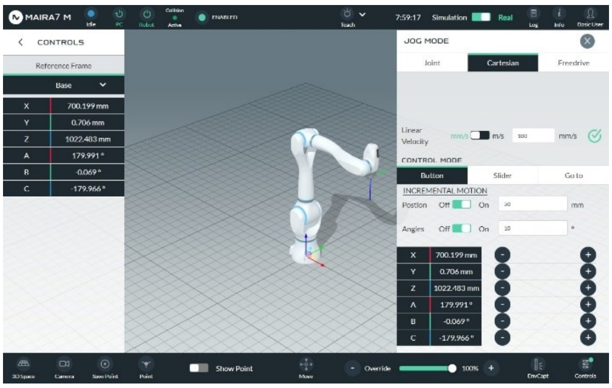
► Motions will be performed with the specified linear velocity (see 8.5.2.1) considering globally set speed limits.

► Note that the **orientation changes are** not interpreted as time derivatives of the roll pitch yaw angles A B C, but as **angular velocities around the x-, y-, and z-axes** of the selected coordinate frame.

INCREMENTAL MOTION activated (**Position** On and/or **Angles** On)

► Toggle **Position** On for the incremental translatory motion (x-, y-, z-coordinates) with the X, Y, Z buttons.

► Toggle **Angles** On for the incremental orientation change (rotation about the x-, y-, z-axes) with the A, B, C buttons.



► Specify the **Position** and or **Angle** increment by which the robot should move (mm or m, depending on the setting for linear velocity, see 8.5.2.1).

►Keeping a button pressed will change the respective position or angle value until the specified increment is reached (or a cartesian limit is exceeded). Releasing the button press during motion will immediately stop the motion for safety reasons.

NOTICE	**Position** On setting only affects the first three "coordinate buttons" X, Y, Z, the **Angles** On setting only affects the last three "orientation buttons" A, B, C. Simultaneous use of modes is possible.

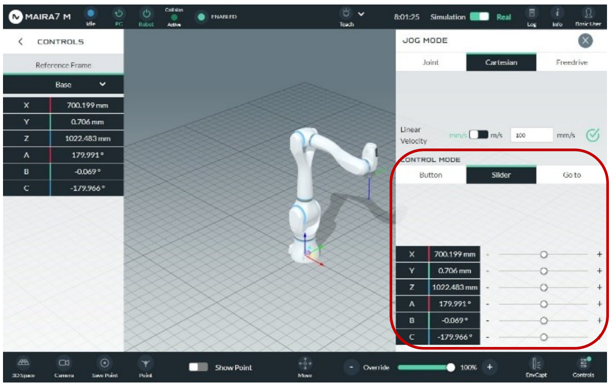
► Note that the values are given in the Reference Frame that is currently selected (Base/ World/ Tool).

► Motions will be performed with the specified linear velocity (see 8.5.2.1) considering globally set speed limits.

Additional remark to the angular increments A, B, C (roll, pitch, yaw): In most configurations, specifying an increment in B (or C), will affect all three angles. This is related to the non-uniqueness of the roll pitch yaw convention: several sets of A, B, C can represent the same orientation. The performed motion always corresponds to the desired one, however, a different set of A, B, C values is shown.

8.5.2.3 Slider

Move the robot's TCP in any direction incrementally. See the explanations above.

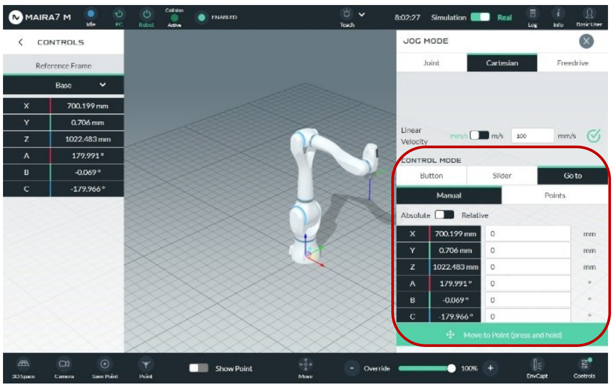


8.5.2.4 Go to

Manual

1. Set movement to **Absolute** or **Relative** mode.
2. Enter a value for the robot movement.
3. Press and hold the **Move to Point** button to move the robot.

   ► Note that not all points can be reached, e.g., due to maximum workspace constraints or singular configurations.



► Note that the motion generated is a joint motion that ends in the specified TCP cartesian configuration. Please watch the workspace during the motion.

Example for absolute manual movement (move + 25 mm in the direction of the x-axis):

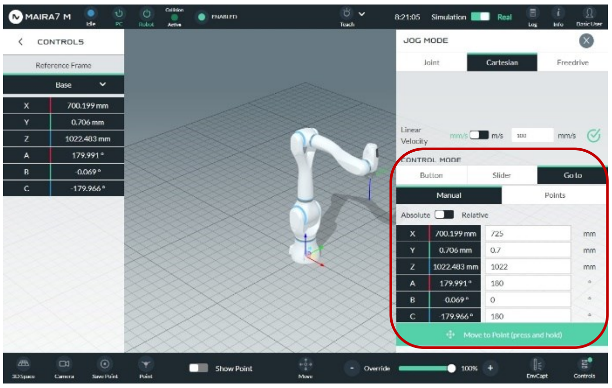
1. Set movement to **Absolute** mode.

2. Type the current values for the coordinates not to be moved into the corresponding fields.

3. Enter the new value for x + 25 mm (in this case approx. 325 mm + 25 mm = 350 mm).

4. Press and hold the **Move to Point** button to move the robot.

5. Monitor the movement of the robot carefully during the execution of the movement and stop the motion by releasing the **Move to Point** button whenever a human, a machine is in danger, or if the robot motion does not conform to your expectation.



NOTICE	Note that not all points can be reached, e.g., when the point lies outside the joint range.

Example for relative manual movement (move + 25 mm in the direction of the x-axis):

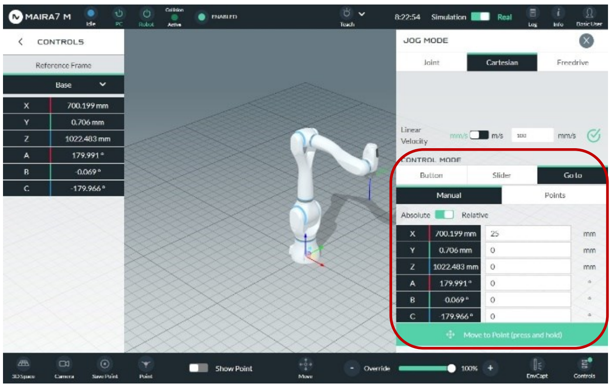
1. Set movement to **Relative** mode.

2. Enter the relative movement value for the x-axis. In this case 25 mm.

3. The values for the other axes remain at 0 mm, because they should not be moved in this example.

4. Press and hold the **Move to Point** button to move the robot.

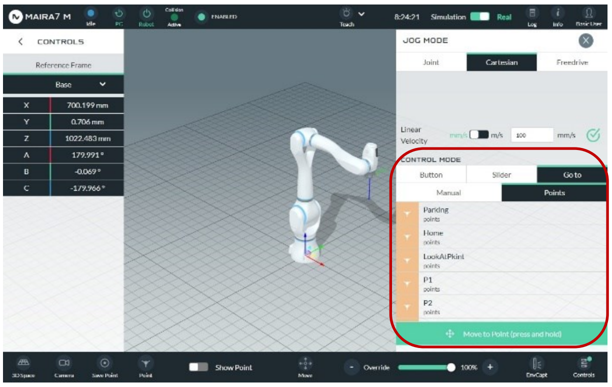
5. Monitor the movement of the robot carefully during the execution of the movement and stop the motion by releasing the **Move to Point** button whenever a human, a machine is in danger, or if the robot motion does not conform to your expectation.



NOTICE	Note that not all points can be reached, e.g., when the point lies outside the joint range.

Points

1. Select a point from the list.
2. Press and hold the **Move to Point** button to move the robot to the selected point.
3. The robot will go to the specified cartesian coordinates of the point and consider the specified **tool**.
4. Monitor the movement of the robot carefully during the execution of the movement and stop the motion by releasing the **Move to Point** button whenever a human, a machine is in danger, or if the robot motion does not conform to your expectation.



8.5.2.5 TCP Align

The align feature lets you orient the robots tool center point (TCP) at its current position relative to a point selected from the points list.

To align the robot's TCP with a point:

► Select the type of alignment

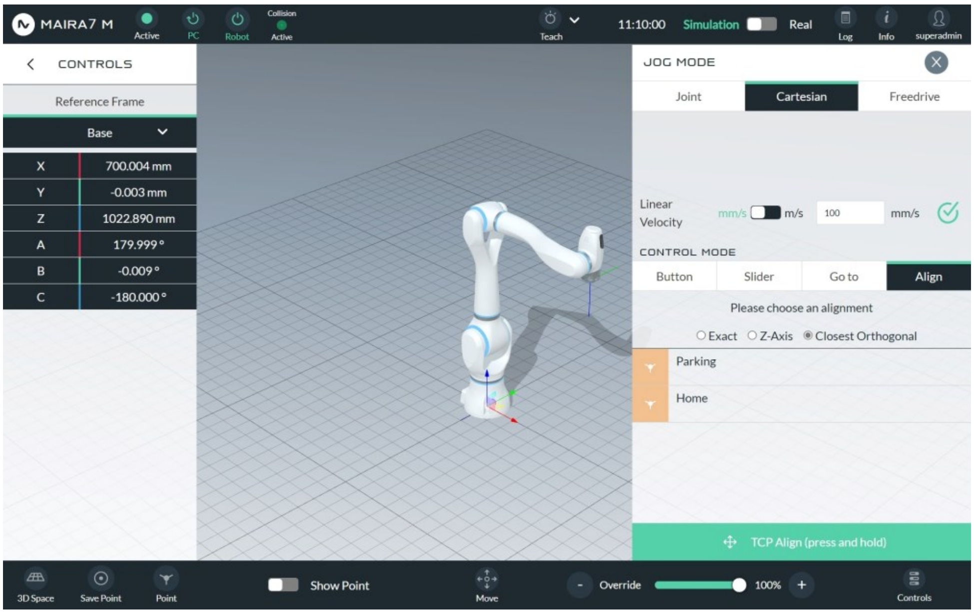
- Exact: select the exact alignment option to orient the robot's TCP at its current position into the pose of the selected point (all axes will be exactly aligned).
- Z-Axis: select the Z-axis alignment option to orient the robot's TCP at its current position into the Z-direction of the selected point (Z-axis will be exactly aligned, X- and Y- axes will be aligned orthogonally closest).
- Closest Orthogonal: select the closest orthogonal alignment option to orient the robot's TCP at its current position into the closest orthogonal orientation regarding the selected point and its reference frame (any axis of the robot's TCP can be aligned to any axis of the point's reference frame, e.g. the Z-axis of the TCP to the X-axis of the point's reference frame. The position of the TCP remains steady, only the orientation is adjusted).
- The TCP and the selected point coordinate system are displayed in the robot visualization to help with spatial understanding.

► Select a point from the points list

► Tap and hold **TCP Align** to align the robot's TCP

- The robot's movement can be paused by releasing **TCP Align**
- Hold **TCP Align** until the robot stops moving to complete the alignment

➔ The robot's TCP is oriented relative to the selected point and alignment option.



8.5.3 Freedrive

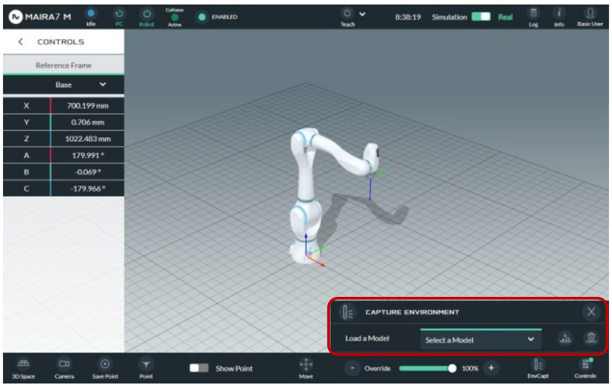
The Zero G mode for hand guiding the robot can also be accessed via the Freedrive tab in the Jog Mode menu (see 8.4.3 for details on how to use Zero G).

8.6 EnvCapt

Capture the actual environment or load an existing scan to be visualized in the 3D GUI space.

To visualize an existing scan:

- Press the **EnvCapt** button in the lower right of the bottom menu bar
- Select an existing model from the **Load a Model** dropdown menu
- Press the **Scan/Load** button to visualize the selected model in the 3D GUI space
- Press the **Wastebin** button to stop displaying the model in the 3D GUI space (model will just be hidden, not deleted)



To scan the actual environment:

- Press the **EnvCapt** button in the lower right of the bottom menu bar
- Select **Current View** from the **Load a Model** dropdown menu
- Press the **Scan/Load** button to scan the environment and visualize the newly captured image in the 3D GUI space

- Press the **Wastebin** button to stop displaying the model in the 3D GUI space (model will just be hidden, not deleted)

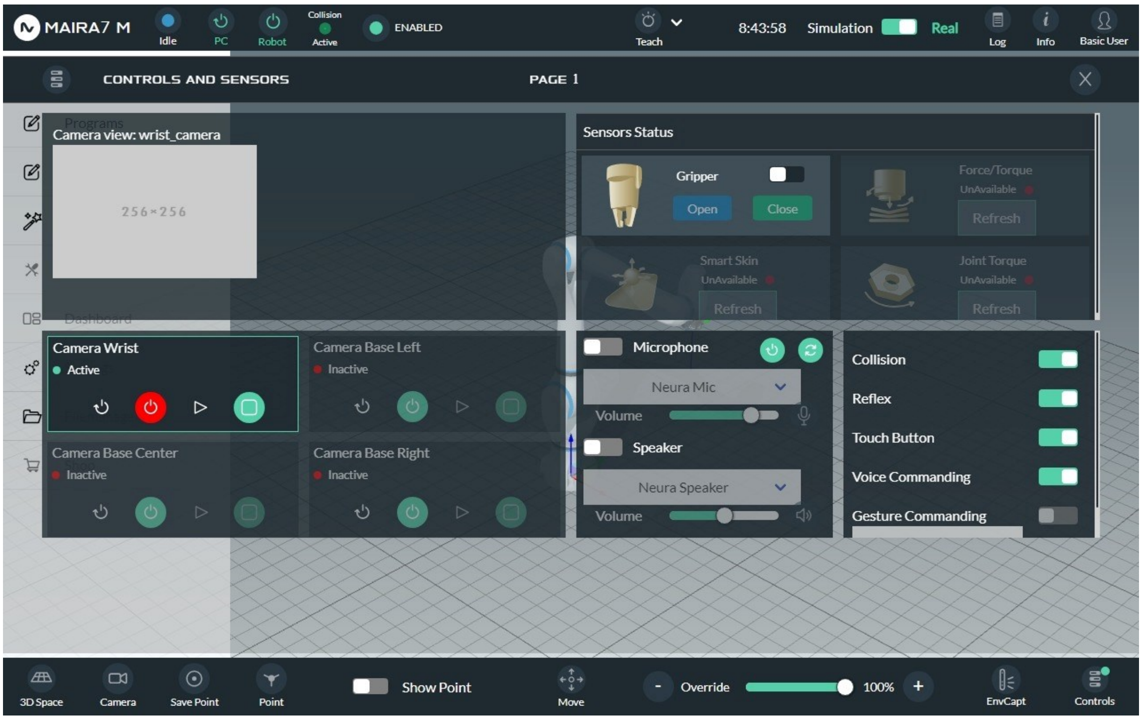

Scan/Load


Tapping on the dust bin does not delete the model, but only hides it.

NOTICE	Please remember to delete obsolete models via the **Filemanger** in the **HOME** menu.

8.7 Controls

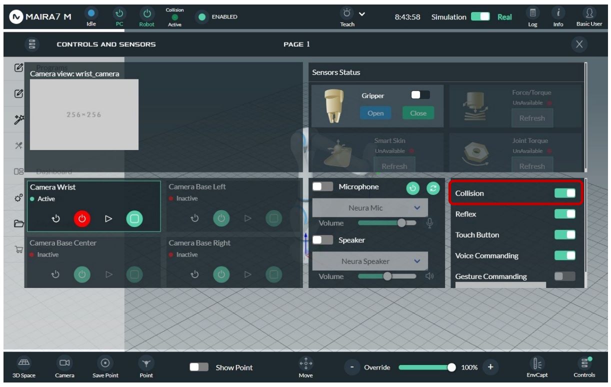
This section explains the functionality of the Controls menu.

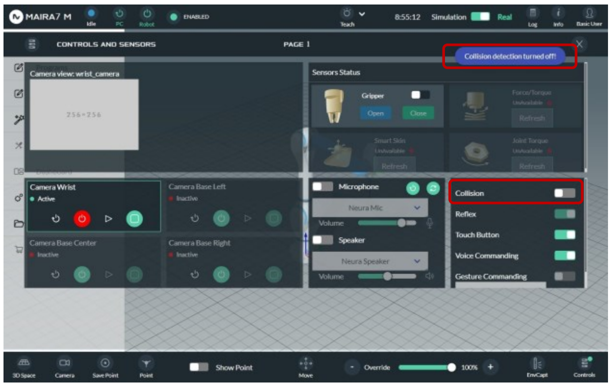


Menu	Meaning	Details / more information
Camera view	Shows a live picture of the selected and available (installed) camera(s).	
Camera Wrist / Base Left / Base Center / Base Right	Camera status and controls.	Camera Active (green) Camera Inactive (red) Restart camera. Start camera (green) Stop camera (red) Show camera image. Hide camera image.
Sensor Status	Currents status of sensors: • **Gripper:** activate and open/close gripper • **Force torque:** sensor measuring the force torque (optional) • **Smart Skin:** legacy feature • **Joint torque:** Sensor measuring the joint torque (optional)	• Status **Active (green)**: sensor active • Status **Error / UnAvailable (red)**: sensor failed or not installed • **Refresh**: refresh status display Note: Gripper will also operate even when in simulation mode.
Microphone	Operate the microphones.	▶ Switch on/off microphone with slider. ▶ Select a microphone from the dropdown menu. ▶ Adjust Volume with slider. ▶ Restart microphone/speaker with ▶ Refresh microphone/speaker list with
Speaker	Operate the speakers.	▶ Switch on/off speaker with slider. ▶ Select a speaker from the dropdown menu. ▶ Change volume with slider.
Voice Commanding	Switch on/off voice command operation.	▶ For more information see chapter 16.
Gesture Commanding	Switch on/off gesture command operation and select camera for gesture recognition.	
Touch Button	Switch on/off touch buttons.	
Collision Off - On	Switch on/off collision detection.	

8.7.1 Collision Off - On

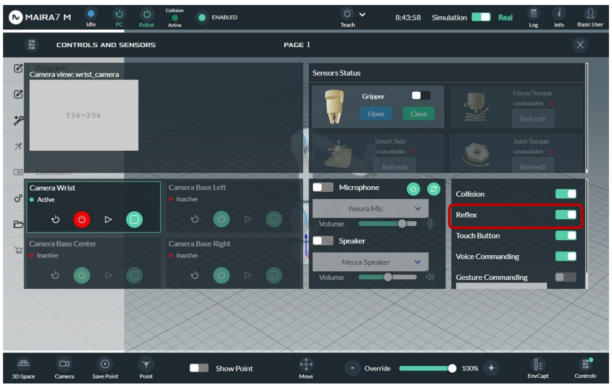
- The password protected option allows you to deactivate the collision detection.
- Without collision detection, the robot cannot be used in collaborative mode anymore and needs special safety precautions.
- When powering on the robot, collision detection is always turned on.
- Collision detection allows safe collaborative work between operator and Cobot in a shared workspace. The Cobot constantly monitors the forces from the environment and stops its motion when the specified collision forces are exceeded.
- Note that the sensitivity of the collision detection depends on the speed of the motion, and the distance from the robot base. A risk assessment always needs to be done when installing the Cobot.





8.7.2 Reflex

- The Reflex toggle allows the operator to choose the reaction strategy after a collision.
- The Reflex option results in a behavior where the robot bounces back after a collision is detected. After a collision no static contact forces, e.g., clamping forces remain.
- If the Reflex option is deactivated, the robot will only stop its motion after a collision. In this scenario, contact forces can remain.
- The Reflex option is recommended.
- After a collision, the operator needs to confirm that the workspace is free of collision object by manually resetting a collision. Collision detection automatically pauses program execution.



► To resume program execution, press the **Play** button

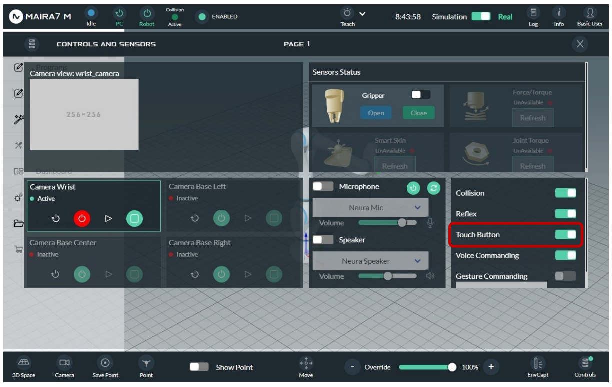
► To cancel program execution, press the **Stop** button and exit the **Program Launcher**

8.7.3 Touch Button (HMI Buttons)

► Select the switch to activate / deactivate the Touch Buttons (HMI Buttons) on the robot head.

► For changing the function of the buttons read chapter 15.8. You can deactivate the buttons if they are not used at all.

► In the picture to the right the Touch Buttons are powered off.



9 POWER ON/OFF THE ROBOT

This paragraph shortly explains powering the robot ON after mechanical and electrical installation and OFF after usage.

For a more detailed description, including information about mechanical and electrical installation, please refer to the MAiRA Quick Start Guide and User Manual.

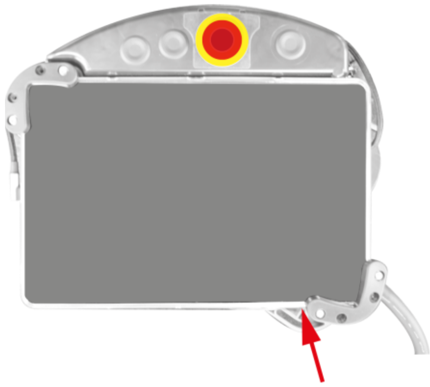
9.1 Power on the robot

- Mechanical and electrical installation must be completed.
  (refer to the MAiRA user manual).

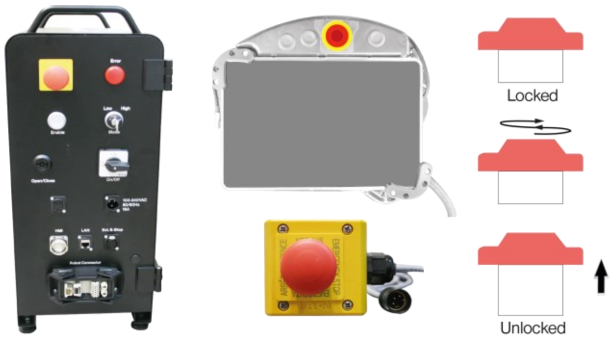
► Switch on power on the robot control box.

- Ensure the Teach Pendant is charging.
  If the charging symbol does not appear, wait 5 minutes to initially charge.

► Turn on the Teach Pendant by pressing the small **Power Button** at the bottom right of the device for more than 10 seconds.



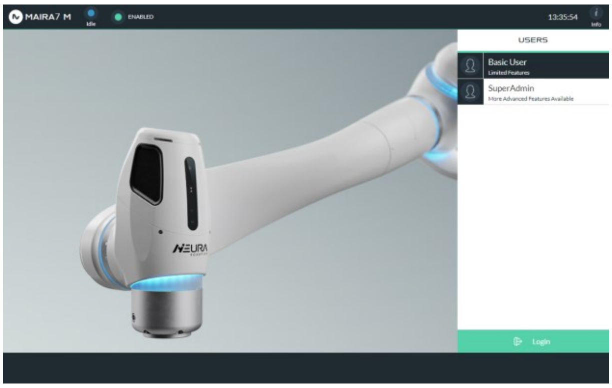
NOTICE	Ensure the external **Emergency Switch** and the **Emergency Switches** on the Teach Pendant and the Control Cabinet and unlocked.



▶ Press the **Enable** button on the control cabinet once.

➔ The button should be **constantly lit**.
🟠 READY_TO_POWER_ON will be shown in the top bar.

Till software version v4.19.0:



🗐 The **Basic User** will be logged in automatically and the User Interface should come up on the Teach Pendant after approximately 30 seconds.

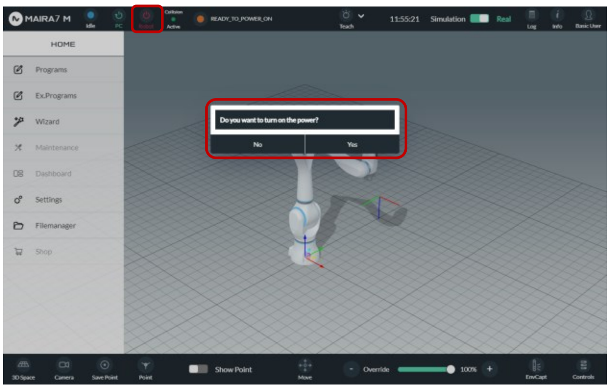
NOTICE	Sometimes it can take up to 2 minutes until the User Interface comes up. If it still doesn't appear, turn the tablet display off and back on again by quickly clicking the tablet power button.

From software version v4.19.1:

The automatic login of the basic user profile was removed, instead a login screen appears providing different user profiles to select for login.

▶ Login with one of the predefined user profiles (see 7.9.1)

▶ Switch the robot in **Real** mode, if the robot is in **Simulation** mode in the top right corner of the robot GUI.



▶ Power on the robot by tapping the **Robot** button in the top left corner of the robot GUI.

▶ Confirm powering on the robot with **Yes**.

➔ Your MAiRA robot is now **ready to use**.

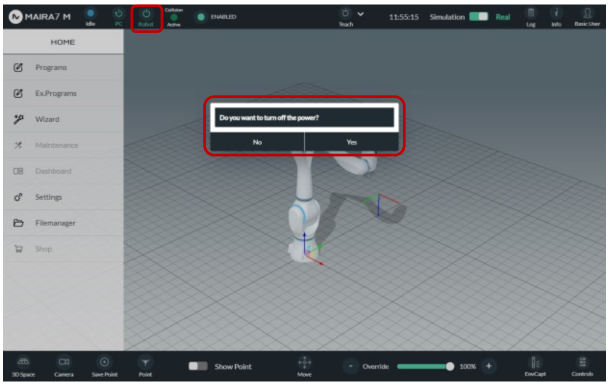
🟢 ENABLED will be shown in the top bar.

⚠ WARNING
Check if the robot's workspace is free. Always be prepared to press the **Emergency Stop** switch while the robot is switched on in case a dangerous situation occurs. When powering on the robot you will hear the breaks releasing with a clicking sound.

9.2 Power off the robot

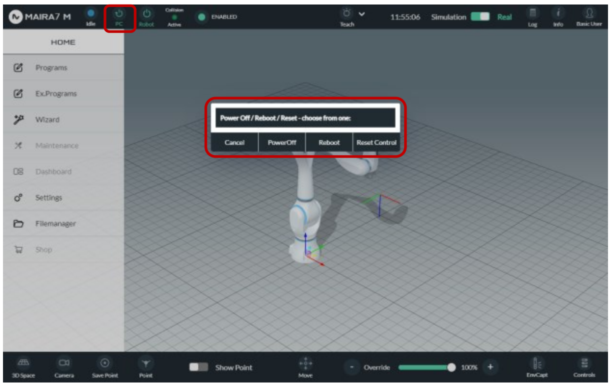
► Power off the robot by tapping the **Robot** button in the top left corner of the robot GUI.

► Confirm powering off the robot with **Yes**.



NOTICE	If you plan on disassembling the robot and repacking it into its original shipping box, move it to the **Parking** point (see chapter 10.1) stored within the robot GUI before powering it off. This ensures the correct robot pose to fit inside the box if the parking point has not been changed in the meantime.

► Shut down the robot controller PC by pressing **PC** in the top left corner of the robot GUI and selecting **PowerOff.**



🗐 Wait for the control PC to shut down properly. This should be completed after 10 seconds.

► Switch off power on the robot control box.

➔ Your MAiRA robot is now powered off.

10 PREDEFINED POINTS

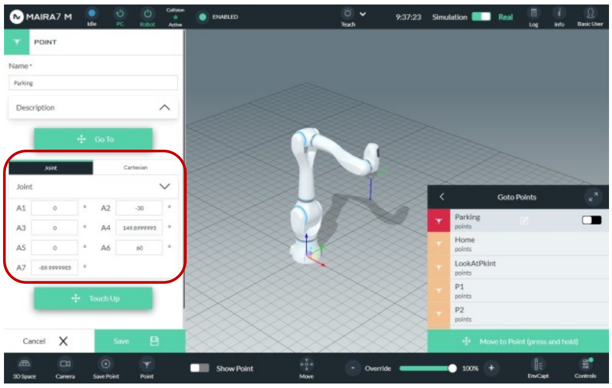
The MAiRA robot software comes with two predefined points.

10.1 Park Position

The **Park** position is used for packaging the robot in its shipping box. Therefore, the robot pose matches the cutouts in the foam box insert.

NOTICE	If you plan on disassembling the robot and repacking it into its original shipping box, move it to this point before powering it off.

If you changed the **Park** position by accident, the original joint angles can be found in the table below.



Parking point joint angles for MAiRA 6 S / M / L:

Joint	Degree	Joint	Degree
A1	0	A2	-30
A3	149.9	A4	0
A5	60	A6	-90

Parking point joint angles for MAiRA 7 S / M / L:

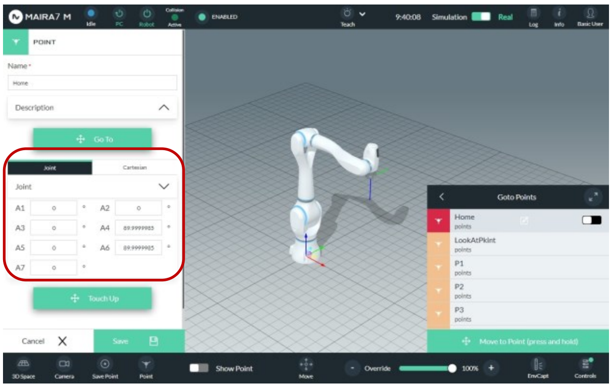
Joint	Degree	Joint	Degree
A1	0	A2	-30
A3	0	A4	149.9
A5	0	A6	60
A7	-90		

10.2 Home Position

The **Home position** is used as a general starting point for using the robot.

If you have just unpacked and installed the robot, move it to this point to start working with it.

If you changed the **Home position** by accident, the original joint angles can be found in the table below.



Home point joint angles for MAiRA 6 S / M / L:

Joint	Degree	Joint	Degree
A1	0	A2	0
A3	90	A4	0
A5	90	A6	0

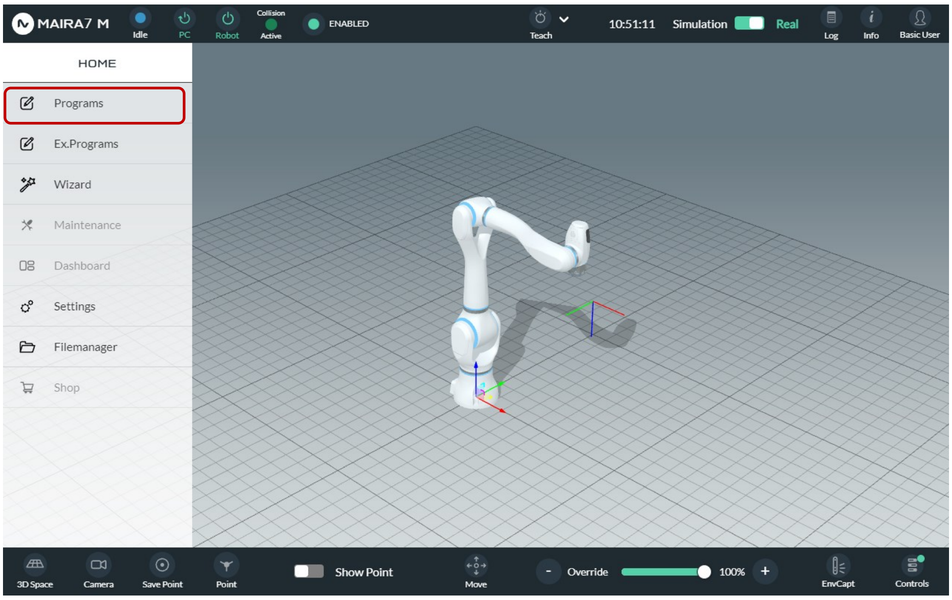
Home point joint angles for MAiRA 7 S / M / L:

Joint	Degree	Joint	Degree
A1	0	A2	0
A3	0	A4	90
A5	0	A6	90
A7	0		

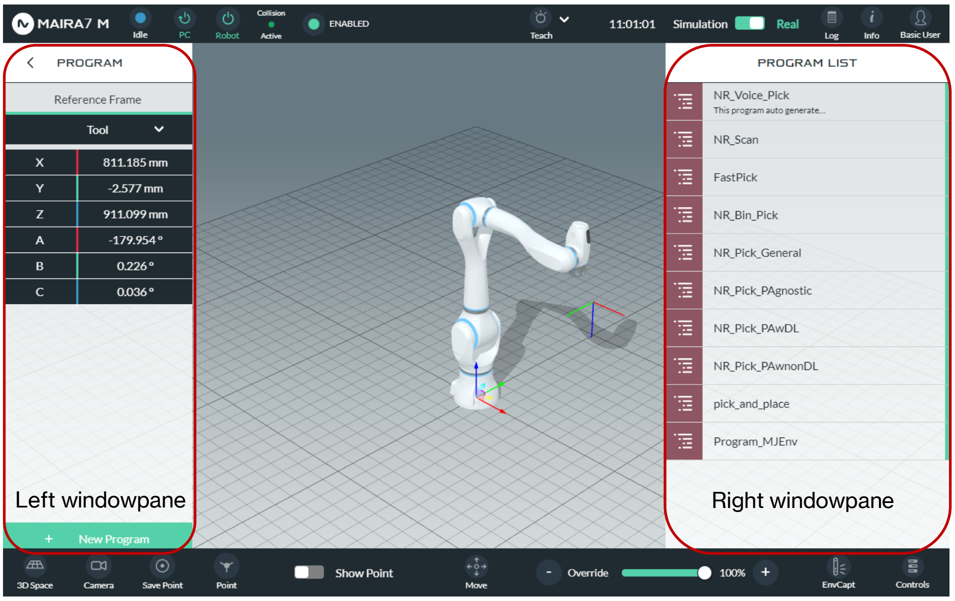
11 PROGRAM MENU

In the Program menu you can create, edit, and run a program.

► Click the **Programs** button to open the menu.



11.1 Left windowpane

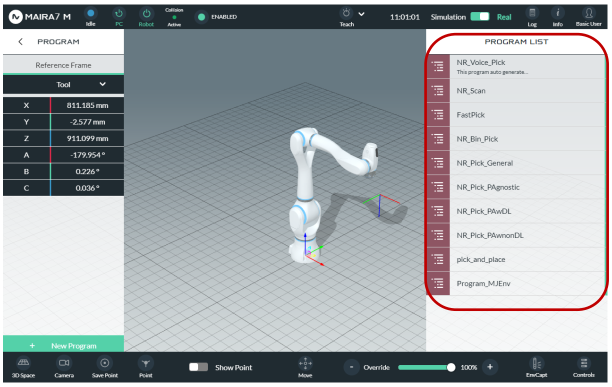


Entry	Meaning	Selection
Reference Frame	Select a Reference Frame for Jog-Mode. Cartesian motions will be performed in the selected coordinate system.	**World:** When the robot is operated mobile, a reference frame outside the robot can be defined in the GUI. When the robot is operated stationary, World corresponds to Base. **Base:** Select the base of the robot as reference frame. **Tool:** Select the tool as the reference frame. • **User:** Create a reference frame with the wizard reference frame and select it here.
X Y Z A B C	Robot coordinates.	• **Robot coordinates base/ reference on the selected reference frame.**
New Program	Click to create a new program.	**1.** Click to create a new program. **2.** Select the editor: **a. Tree Editor:** Create a program based on a tree structure. **b. Code Editor:** Not available yet (create programs with Python scripts). Use the Python / ROS API or run as an external program instead.

11.2 Right windowpane

Entry	Meaning	Selection
PROGRAM LIST	Already created programs are listed here.	► Click on an entry to open, edit, copy, delete or start a program.

11.3 Using a program

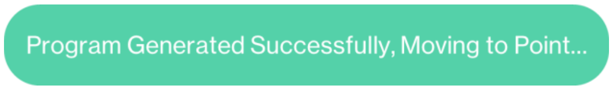


To use an existing program:

1. Select the program from the **Program List**.

2. Click the **Play** button.

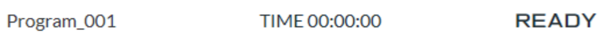
3. The program will be generated for running.
4. Successful program generation will be indicated by a green popup message.



5. The **Launcher** navbar in the top section of the GUI can then be used for controlling the program.

In the left section:

- The name of the program is shown.
- The elapsed running time of the whole program is shown.
- The program status is shown (READY, INITIAL for initializing, PAUSED, STOPPED or ERROR).



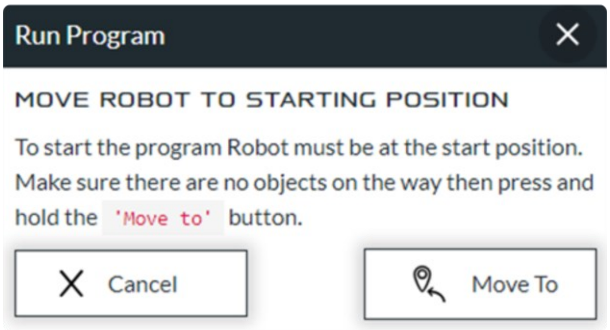
In the middle section:

- The program can be started with the **Play** button.
- The program can be paused with the **Pause** button.
- The program can be stopped with the **Stop** button.

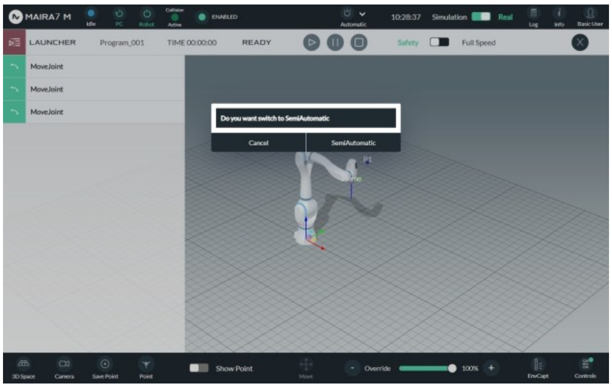
In the right section:

- The speed for running the program can be set.
  - **Safety**: the program runs at safety speed.
  - **Full Speed**: the program runs at the defined speeds for the individual motion commands and with the actual **Override** settings.
  - With the **X** button, you can exit the program execution.

If you start a program and the robot is not yet at the starting position of the program, the robot will tell you that it needs to move to the starting position of the program.



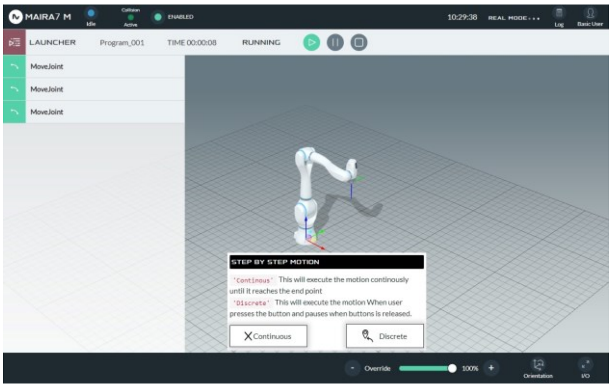
6. Press and hold the **Move To** button until the robot has reached the starting point of the program.
7. Click the **Play** button to start the program execution.
8. After the initial program position is reached you will be asked to switch to SemiAutomatic mode.
9. Depending on the mode, program execution can be performed differently.
    a. In **Automatic** mode, programs with all their instructions will be executed automatically.
    b. In **SemiAutomatic** mode, you can execute programs step-by-step, instruction for instruction.
10. Press **SemiAutomatic** to switch to SemiAutomatic mode.
    a. If **Cancel** is pressed, mode will be switched to Automatic.
    b. You can also use the top bar dropdown menu to switch modes (see 7.6).



NOTICE	Once switched to **SemiAutomatic** mode, you cannot switch to **Automatic** mode until the end of program execution.

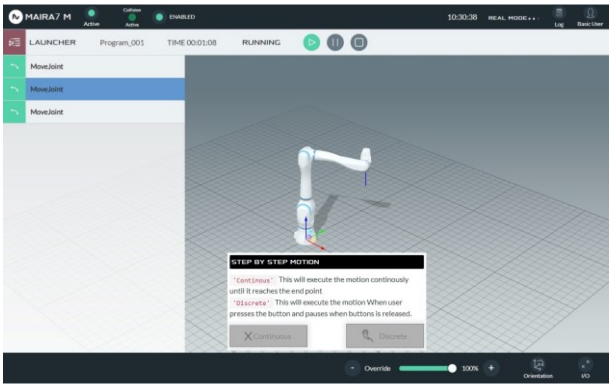
NOTICE	Once asked to switch to **SemiAutomatic** mode, pressing anywhere else than **Cancel** or **SemiAutomatic** will result in switching to **Automatic** mode.

11. In **SemiAutomatic** mode, briefly after switching mode, the "**STEP BY STEP MOTION**" dialog will appear.



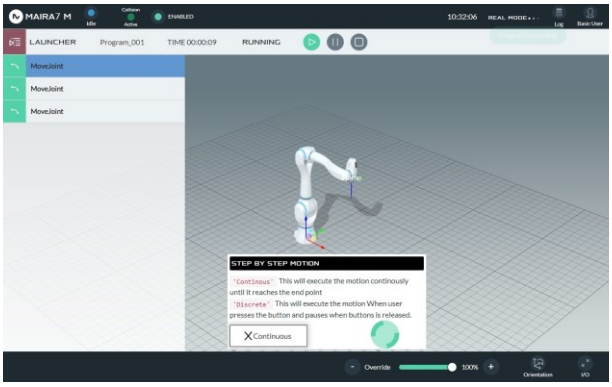
12. The **Continuous** button will execute program instructions (e.g. **MoveJoint** or **Wait**) step-by-step one after the other until the respective instruction is finished

- ▶ Tap the **Continuous** button
- ▶ Current instruction will be performed until finished
    - o Currently executed instruction is highlighted blue in the program overview
    - o The **Continuous** button is greyed out until the instruction is finished

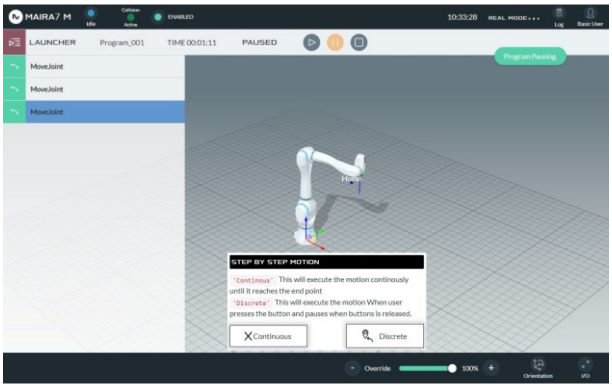


- ▶ Tap the **Continuous** button again to execute the next instruction
- ▶ Step through all the program instructions until the end of the program

13. The **Discrete** button will also execute program instructions (e.g. **MoveJoint** or **Wait**) step-by-step one after the other, but the respective instruction is only executed as long as the **Discrete** button is kept pressed:



- ▶ Keep the **Discrete** button pressed
- ▶ Current instruction will be performed as long as **Discrete** button is kept pressed
    - o Currently executed instruction is highlighted blue in the program overview
    - o If the **Discrete** button is released, execution of the respective instruction is stopped (and continued if kept pressed again)



- ▶ Keep the **Discrete** button pressed again to execute the next instruction
- ▶ Step through all the program instructions until the end of the program

NOTICE	If the program contains a loop command and the **Continuous** button is pressed at the end of the current loop, the next loop with its first instructions is executed. When using the **SemiAutomatic** mode with non-motion instructions like IF/ELSE IF/ELSE, pressing the **Continuous** button will advance to the respective "conditional" command block. To proceed further within an IF/ELSE IF/ELSE condition, the **Continuous** button must be pressed again. If the condition is met, it will execute the corresponding block; otherwise, it will move to the next program instruction. If a "**Wait Before**" or a "**Wait After**" timespan is inserted inside a Gripper App instruction, the whole Gripper App instruction will be executed in a single step if the **Continuous** button is pressed. It will not pause within the Gipper App instruction block. If a program does not contain any motion commands (e.g. **MoveJoint** or **MoveLinear**), and only contains logic instructions, please use the top bar dropdown menu to switch to **SemiAutomatic** mode (see 7.6).

11.3.1 Demo programs

NEURA provides some ready-to-use programs, which demonstrate the functionalities and usage of the AI Apps in specific scenarios. The programs can be found in **HOME >> PROGRAM** and a short overview is given here:

Program name	Description
NR_Voice_Pick	Program that is used to communicate with MAiRA using your voice to tell her which object to pick.
NR_Scan	Scanning of the environment and objects.
NR_Pick_General	Picking objects in the workspace with pick type general.
NR_Pick_PAgnostic	Picking objects in the workspace with pick type pose-agnostic.
NR_Pick_PAwDL	Picking objects in the workspace with pick type pose-aware DL.
NR_Pick_PAwnonDL	Picking objects in the workspace with pick type pose-aware non-DL.

Start a program by clicking on the program in the list and push the play button. Programs can also be duplicated (copy button) or edited (edit/save button).

See chapter to get more information on creating/editing programs.

11.3.1.1 Voice Pick

This program demonstrates the ability of MAiRA to listen to given commands, recognition of pretrained objects and generating a grasping pose for known and unknown objects.

Preparation

- Have a workspace defined (see 0)
- Place objects inside the workspace
- Turn on voice commanding (see 16.1)
- Turn on the camera (see 8.7)
- Load the correct object segmentation model (object detection settings)

- ► Activate the voice commanding by saying “Hey, MAiRA”
- ► Tell MAiRA to pick a specific object, “pick mouse” or nonspecific “pick any object”. Make sure the right AI model is chosen in the AI settings
- ► MAiRA will repeat the command before execution. Confirm with “yes” or decline with “”no”
- ► MAiRA generates a grasping pose (displayed on the GUI) and picks the object

11.3.1.2 Scan

The scan program gives an example of scanning the workspace using the Scan App. For more information and configurations, navigate to the Scan App.

Preparation

- Have a workspace defined (see 0)
- Place objects inside the workspace
- Turn on the camera (see 8.7)

11.3.1.3 Pick

Executing this program, MAiRA recognizes and grasps every object in the workspace one after the other using a specific pick type. See the explanation of the pick types here.

Preparation

- Have a workspace defined (see 0)
- Place objects inside the workspace
- Turn on voice commanding (see 16.1)
- Turn on the camera (see 8.7)
- Load the correct object segmentation model (object detection settings)

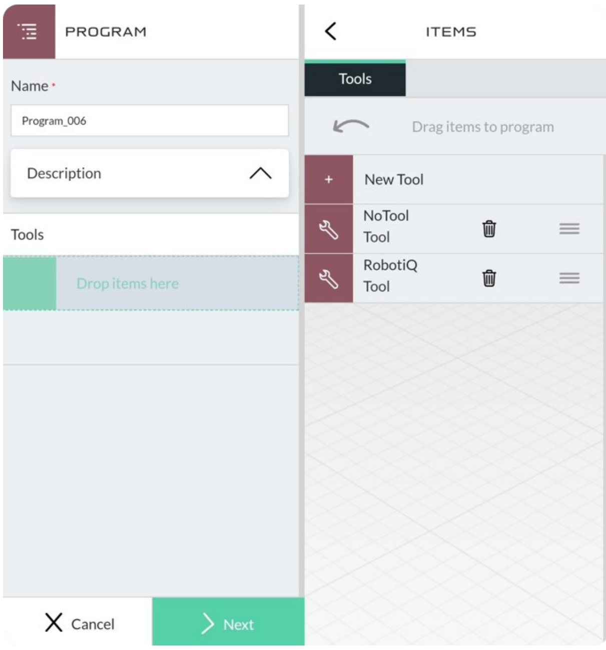
11.4 Creating a program

1. In the Program menu, tap on **New Program**.
2. **Tree Editor:** Create a program by dragging built-in function blocks into a tree structure.

1. Type in the desired name for the new program and add a description if needed.
2. Select a tool from the items list and drop it to the **Tools** area of your program or select the **New Tool** button to create a new item.

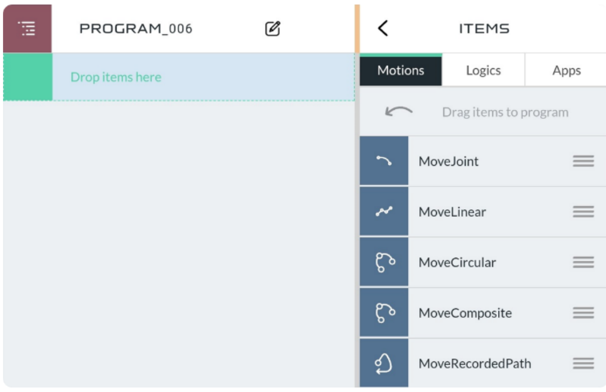
Item	Description
Tools	▶ Select the tool that is mounted to the TCP. Tools can also be created and edited in the **Settings** menu: **Settings > Tools**

3. Select **Next**.



Motion / Logic / Apps

4. Select an item from the list and drop it to the program tree.
5. To edit an item select it in the program tree and select the edit button.
   - ▶ For more information about the items see the following sections.
6. You can change the order just by dragging an item to the wanted position in the tree. Drop will only work, when position in tree makes sense in program syntax.
7. Select **Save** to save your settings.



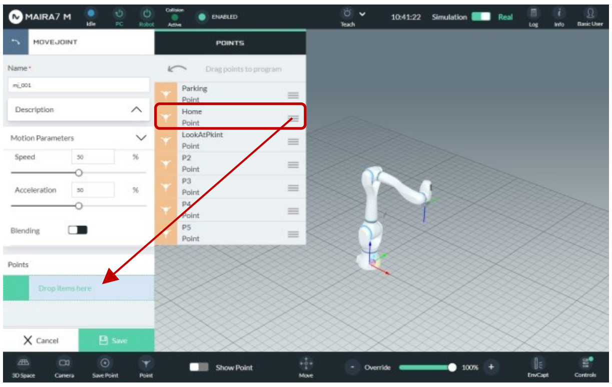
11.5 Motion

▶ To edit or delete an item select it in the program tree and click the **Edit** or **Delete** button.

11.5.1 MoveJoint

This command will move robot joints to selected points.

1. Edit name and add description if wanted.
2. Edit items and save settings.



Item	Description
Points	► Drop points to the list. Points that are already used will go to the bottom of the list.
	NOTICE If points are dragged from the **Points list** into a program (e.g. into a motion instruction) program specific copies of the points will be made. Any changes to points made within the program will not be reflected in the respective general points in the **Points list**. Program specific points that have been changed within a program should be renamed to avoid confusion. To update general points, changes must be applied outside of a program (e.g. from the **Jog Mode / Go to / Points** dialog, see 8.5.1.4, 8.5.2.4).
Speed	► Set speed (factor 0 to 100 %) * This is the percentage of the maximum allowed velocity set in the **Global Settings.**
Acceleration	► Set acceleration (factor 0 to 100 %) This is the percentage of the maximum allowed acceleration. *

* Enter value or drag ruler.

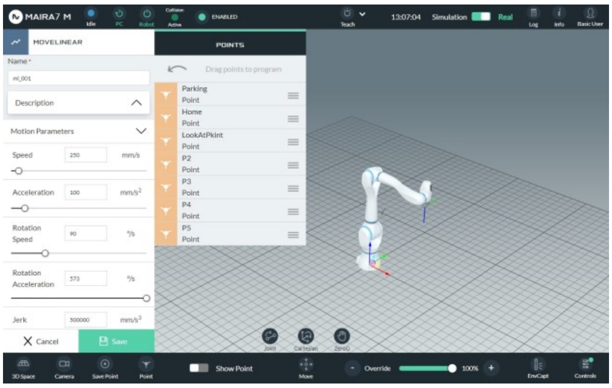
11.5.2 MoveLinear

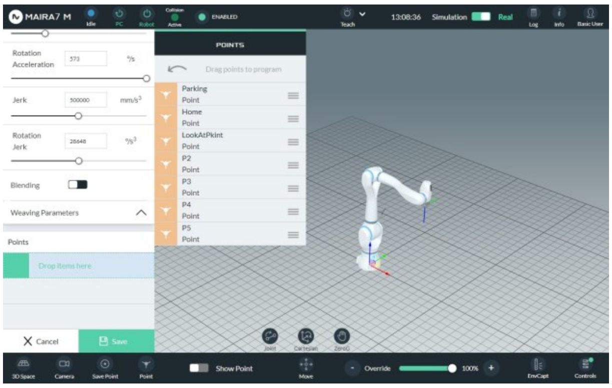
This command moves robot linear between points.

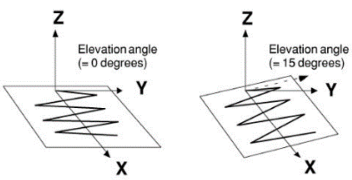
This command moves the robot linearly between points.

1. Edit name and add description if necessary.
2. Edit items and save settings.

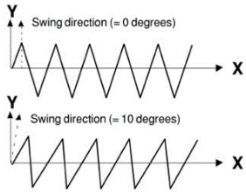
Item	Settings
Speed	Set speed. *
Acceleration	Set acceleration. *
Rotation Speed	Set rotation speed. *
Rotation Acceleration	Set rotation acceleration. *
Jerk	Set jerk. *
Rotation Jerk	Set rotation jerk. *
Blending	Check box to enable **blending**. Smoothly move through points without stopping at each point. Specify radius as moving is in cartesian space. TCP will blend in the direction of the next point.
Blending Radius	Set the blending radius in mm
Weaving Parameters	► Press the toggle to overlay the motion with a weaving motion ► Choose between the **Sine** & **Circle** patterns. ► Specify the **amplitudes** to the right and left of the motion (in the direction of the motion).







►Specify the **Azimuth** in degrees. This item specifies the inclination of weaving swing direction on the weaving plane. A positive value inclines the right edge at move direction, a negative value the left edge. 90 degrees means that the weaving motion is parallel to the motion. The range is -90° to +90°.



► Specify the **Elevation** in degrees. This item specifies the inclination of the weaving plane relative to the weaving frame. The range is -90° to +90°.

► Specify the **Frequency** of the weaving pattern. The maximum value is 3 Hz. The minim value is 0.01 Hz.

► The right and left delays are used to specify the start offset of the weaving motion, e.g., when there is an initial offset from the reference path at the beginning of the pattern.

Points

► Drag and drop points to the list. Or move robot into position and click “save point” on the bottom at the screen. This will automatically drop the new point into this list.

NOTICE	If points are dragged from the **Points list** into a program (e.g. into a motion instruction) program specific copies of the points will be made. Any changes to points made within the program will not be reflected in the respective general points in the **Points list**. Program specific points that have been changed within a program should be renamed to avoid confusion. To

		update general points, changes must be applied outside of a program (e.g. from the **Jog Mode / Go to / Points** dialog, see 8.5.1.4, 8.5.2.4.

* Enter value or drag ruler. If the specified speed exceeds the maximum allowed speed in the **Global Settings**, the motion will automatically be slowed down.

11.5.3 MoveCircular

This motion instruction lets you create circular robot motions by specifying three points. The circular motion will be performed from the specified Start Point over the given Waypoint to the defined End Point. Please see the note on specifying points below to avoid unexpected robot behavior.

► To use the instruction in your program, drag the instruction into your program tree.

To configure the instruction, follow these steps:

► Tap the instruction to select it.
► Tap the **Edit** icon.
► Set the parameters listed and explained below to configure the instruction according to your program needs.

Parameter	Explanation
Name	► Give the instruction a name or use the auto-generated one.
Description	► Give the instruction an optional description.
Motion Parameters	► Define your required parameter set. See “MoveLinear” for details.
Weaving Parameters	► Define your required parameter set. See “MoveLinear” for details.
Start Point	Define the starting point for your circular motion. ► Drag a point from the points list into the Start Point box or create a new point. See the note on specifying points below to avoid unexpected motion behavior.
Waypoint	Define the waypoint for your circular motion. ► Drag a point from the points list into the Waypoint box or create a new point. See the note on specifying points below to avoid unexpected motion behavior.
End Point	Define the end point for your circular motion. ► Drag a point from the points list into the End Point box or create a new point. See the note on specifying points below to avoid unexpected motion behavior.

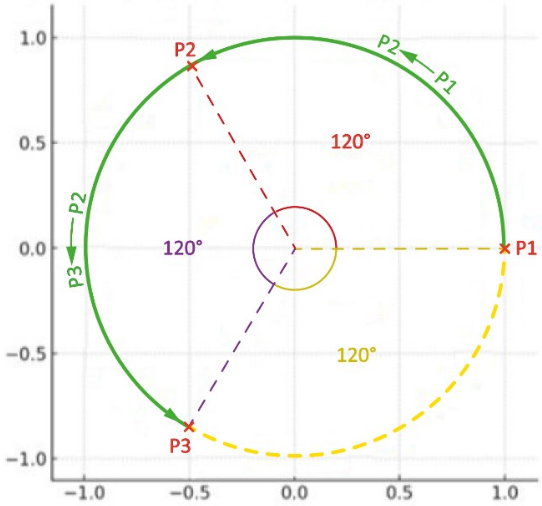
► To save your instruction settings, tap the **Save** button.

➔ The MoveCircular instruction is configured and can be used within your program.

Note on specifying points for the MoveCircular instruction:

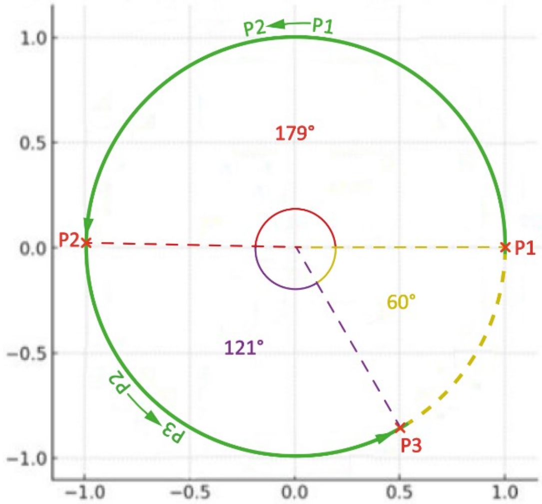
The angles between the specified points must be <180°. Otherwise, unexpected motion behavior can occur. Please see the following illustrations for a better understanding (P1 is the Start Point, P2 the Waypoint and P3 the End Point of the desired circular motion):

Fig. 1



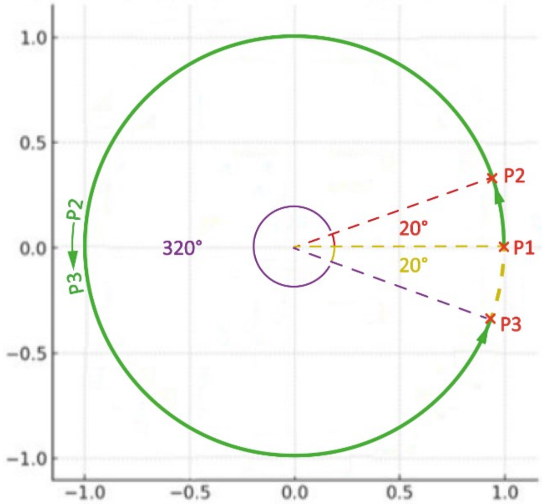
The angles between P1 and P2 as well as P2 and P3 are <180°. The circular motion will be performed on the assumed circle (dashed yellow) as expected (solid green arrows).

Fig. 2



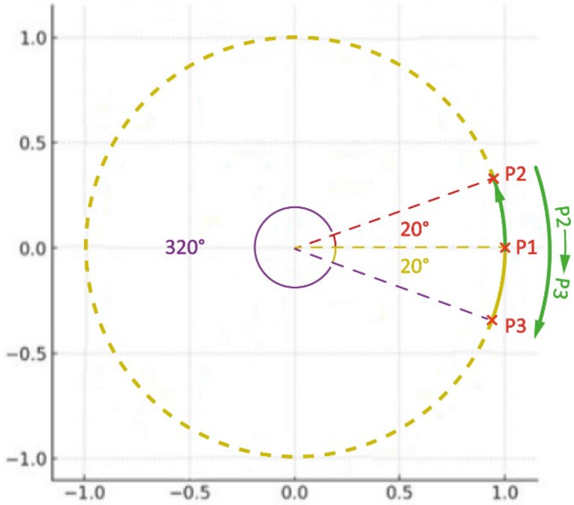
The angle between P1 and P2 is 179°, still <180°. The circular motion will be performed on the assumed circle (dashed yellow) as expected (solid green arrows). For angles = 180° unexpected behavior might or might not occur (see figures 3 and 4 for details).

Fig. 3

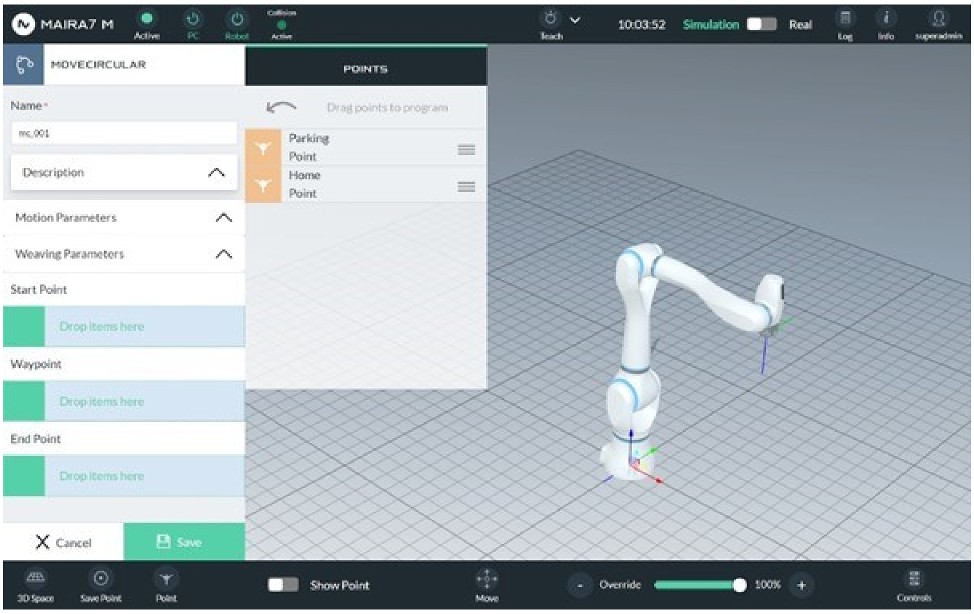


The angle between P2 and P3 is >180°. The figure depicts the expected motion behavior (solid green arrows) on the assumed circle (dashed yellow). But as the angle is >180°, the robot will behave differently. Fig. 4 shows the actual behavior.

Fig. 4



As the angle between P2 and P3 is >180° the robot will move along the shortest path from P1 to P2 (solid green arrow) and back from P2 to P3 (solid green arrow) not following the assumed circle (dashed yellow).

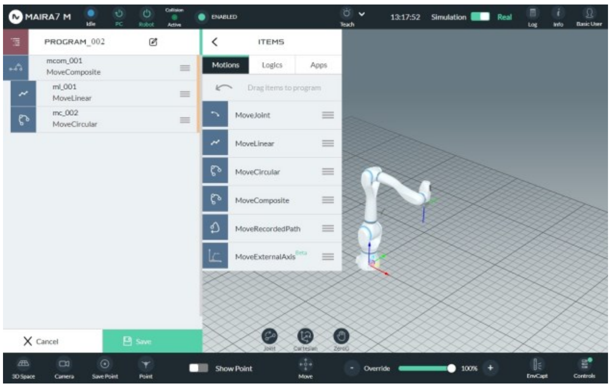


11.5.4 Move Composite

This command lets the robot follow a composite of cartesian movements (**MoveLinear** and **MoveCircular**).

1. Edit name and add description if wanted.
2. Edit speed and acceleration and other parameters if needed. The parameters inside the ML or MC movements are ignored.
3. The ML and MC movements must be subordinated.

Item	Settings
MoveLinear	Check the **MoveLinear** chapter.
MoveCircular	Check the **MoveCircular** chapter.

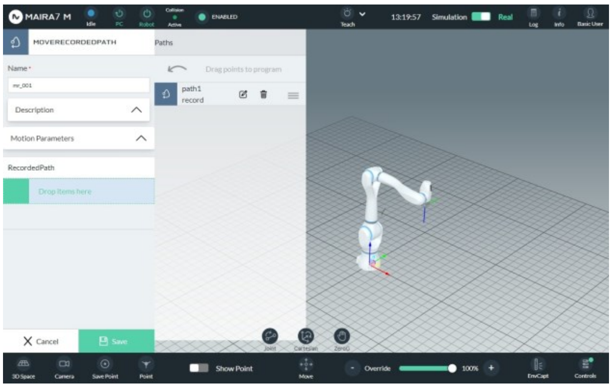


11.5.5 MoveRecordedPath

This command lets the robot follow a recorded path specified in ZeroG mode.

1. Edit name and add description if wanted.
2. Edit items and save settings.

Item	Settings
Do you want constant velocity?	Check box to enable constant velocity. The recorded trajectory will be altered to achieve a constant TCP velocity during the motion.
Speed	Set speed (enter value or drag ruler).
Recorded Path	Drop a recorded path to list.
Constant velocity	Values from 0 to 1000 mm/s.



11.5.6 MoveServo Beta

This motion instruction lets you move the robot using position and pose data of an object to be manipulated. Robots with integrated cameras and pose estimation capabilities can acquire the object data themselves. Other robots can use external camera-aided systems or programmable logic controllers (PLCs) to obtain the required object data.

► To use the instruction in your program, drag the instruction into your program tree.

To configure the instruction, follow these steps:

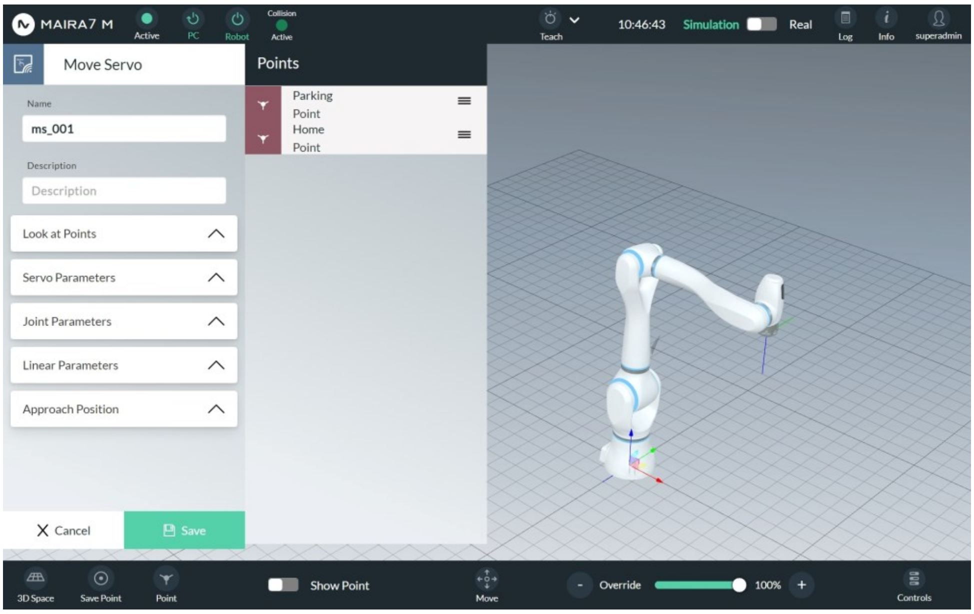
► Tap the instruction to select it.
► Tap the **Edit** icon.
► Set the parameters listed and explained below to configure the instruction according to your program needs.

Parameter	Explanation
Name	► Give the instruction a name or use the auto-generated one.
Description	► Give the instruction an optional description.
Look at Points	Look at points are points, the robot will move to in order to obtain camera images for object pose estimation. At least one look at point must be given. When using integrated cameras or external cameras mounted to the robot, the provision of multiple look at points is recommended (different viewing angles are beneficial for object pose acquisition). ► Drag one or multiple points from the points list into the look at points box. Without successful object detection/pose estimation, program execution will be terminated.
Servo Parameters	Select the use case: ► Camera: camera is the only available use case yet. Select the type of camera: ► Choose your camera from a list of supported devices (the devices EipServoXInterface and EipServoXOffsetInterface are meant to be used with PLCs, see note below). Select the type of servoing: ► Static: select static to detect an object pose once to move to a stationary object. ► Dynamic: select dynamic to continuously detect an object pose to follow a moving object.
Joint Parameters	► Set the speed and acceleration [%] for joint motions. The robot will move to the look at points in joint motion (cf. MoveJoint).
Linear Parameters	► Set the speed [mm/s] and acceleration [mm/s²] for linear motions. The robot will first move to the given object approach position and then to the object in linear motion (cf. MoveLinear)

Approach Position	▶ Define the object approach position by specifying an offset to the detected pose in the x-, y-, z-direction [mm]. ▶ The approach position offset is specified in the tool frame, the coordinate system referenced to the tool attached.

- ▶ To save your instruction settings, tap the **Save** button.

➔ The MoveServo instruction is configured and can be used within your program.



11.5.7 MoveRelative Beta

This motion instruction lets you move the robot relative to a predefined point.

- ▶ To use the instruction in your program, drag the instruction into your program tree.

To configure the instruction, follow these steps:

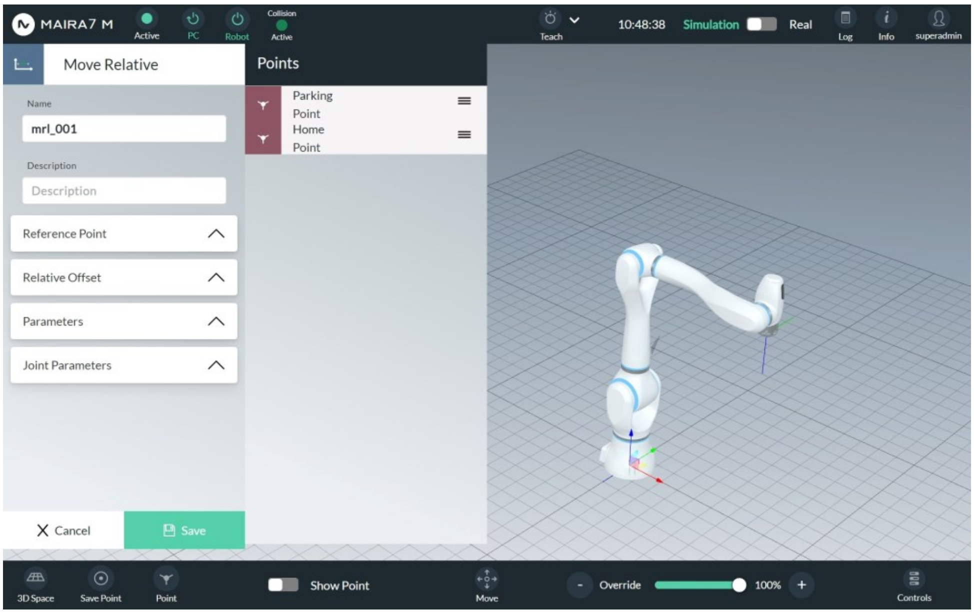
- ▶ Tap the instruction to select it.
- ▶ Tap the **Edit** icon.
- ▶ Set the parameters listed and explained below to configure the instruction according to your program needs.

Parameter	Explanation
Name	▶ Give the instruction a name or use the auto-generated one.
Description	▶ Give the instruction an optional description.
Reference Point	The reference point is the point in relation to which the relative motion will be performed. ▶ Drag a point from the points list into the reference point box.

Relative Offset	► Define the direction of the relative motion by specifying the corresponding offset in x-, y-, z-direction [mm] and/or rotation about those axes A, B, C [deg].
Parameters	Select the type of motion: ► Joint: set the type to joint to achieve a point-to-point motion. ► Linear: set the type to linear to achieve a linear motion. Select the reference frame: ► Base: set the reference frame to base to conduct the relative motion in the base coordinate system referenced to the base mounting surface. ► Tool: set the reference frame to tool to conduct the relative motion in the tool coordinate system referenced to the tool attached.
Joint Parameters / Linear Parameters	Depending on the selected type of motion, you can define the motion parameters: ► For joint motions, you can set the speed and acceleration [%]. ► For linear motions, you can set: o Speed [mm/s] o Acceleration [mm/s²] o Rotation Speed [°/s] o Rotation Acceleration [°/s²] o Jerk [mm/s³] o Rotation Jerk [°/s³] o Blending o Weaving

► To save your instruction settings, tap the **Save** button.

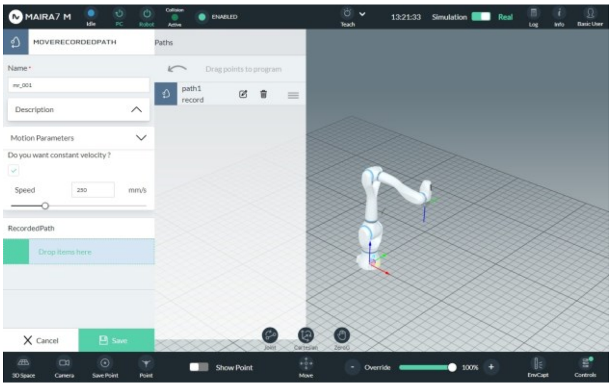
➔ The MoveRelative instruction is configured and can be used within your program.



11.5.8 MoveExternalAxisBeta

This command allows for control of an external axis (currently only Linak vertical lift axes).

1. Edit name and add description if wanted.
2. Edit items and save settings.



Item	Settings
Linear Axis	Select the linear axis you want to move.
IP	IP address of the linear axis (can only be changed via the **Settings >> ExternalAxis** menu).
Port	Communication port of the linear axis (can only be changed via the **Settings >> ExternalAxis** menu).
StrokeLength [mm]	Maximum allowed movement distance of the linear axis (can only be changed via the **Settings >> ExternalAxis** menu).
Current [A]	Power setting to accommodate for heavier lifts / faster movement speeds (can only be changed via the **Settings >> ExternalAxis** menu).
Position [mm]	Set the vertical position of the axis in mm.
Speed [mm/s]	The speed for axis movement in mm/s (can only be changed via the **Settings >> ExternalAxis** menu).

11.6 Logic

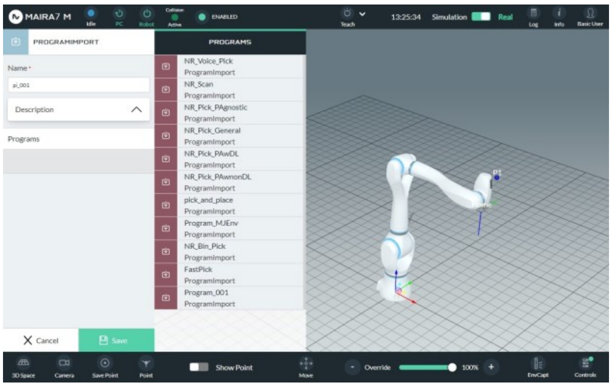
► To edit an item, select in the program tree and select the edit button.

11.6.1 Program Import

Import a program generated via the Tree Editor. This allows you to use existing programs as sub-routines in a program. Note, that changes applied to the subprogram after importing will not get updated. Remove and import the subprogram again for the changes to be reflected in the sub-routine.

1. Edit name and add description if wanted.
2. Edit item and save settings.

Item	Settings
Programs	► Select a program from the list to import it.

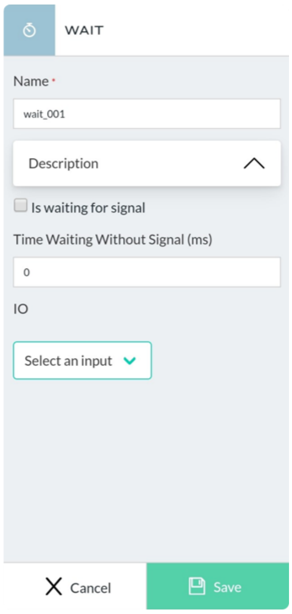


11.6.2 Wait

Add a wait step to the program.

1. Edit name and add description if wanted.
2. Edit items and save settings.

Item	Settings
Is waiting for signal	► Check box if the program must wait for a signal.
Time Waiting Without Signal (ms)	► Enter the wait time in milliseconds to specify the duration of the pause.
IO	► Select an input signal to wait for from the dropdown menu.

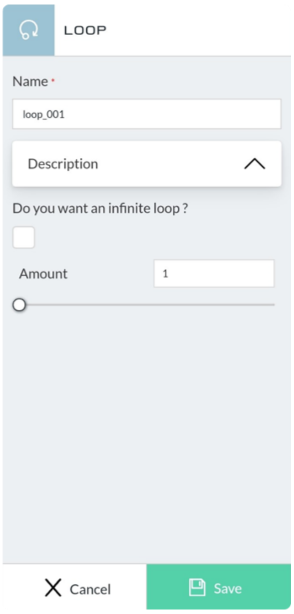


11.6.3 Loop

Add a loop to the program.

1. Edit name and add description if wanted.
2. Edit items and save settings.

Item	Description
Do you want an infinite loop?	► Check box to enable infinite loop.
Amount	► Set counter for number of repeated loops.

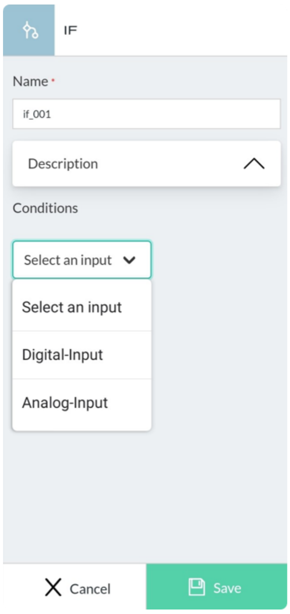


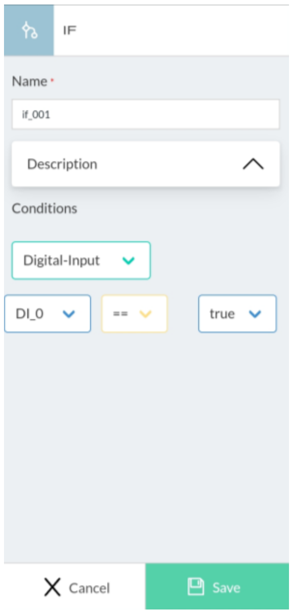
11.6.4 If

Add an if function to the program.

1. Edit name and add description if wanted.
2. Edit item and save settings.

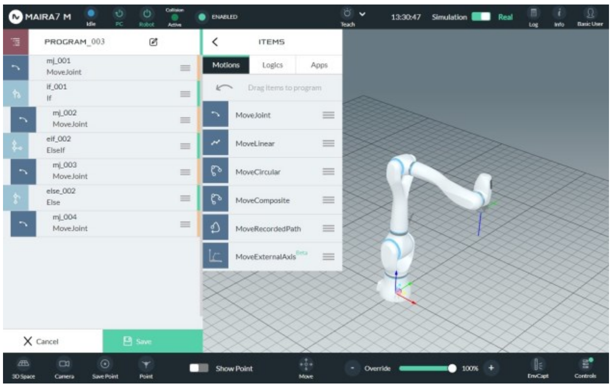
Item	Settings
Select an input	► Select an input type from the dropdown menu: **Digital-Input** **Analog-Input**
Input no.	► Select a numbered input from the dropdown menu: E.g. **DI_0**
Comparison operator	► Select the comparison operator from the dropdown menu:





	== means **equal** **!=** means **not equal**
Boolean condition	► Select the Boolean condition from the dropdown menu: **true** or **false**

Instructions within the **IF** statement must be nested underneath the **IF block** within the program tree editor.

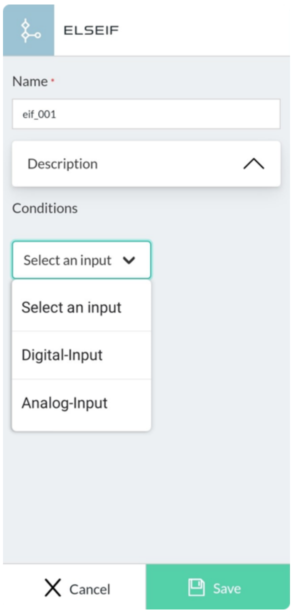


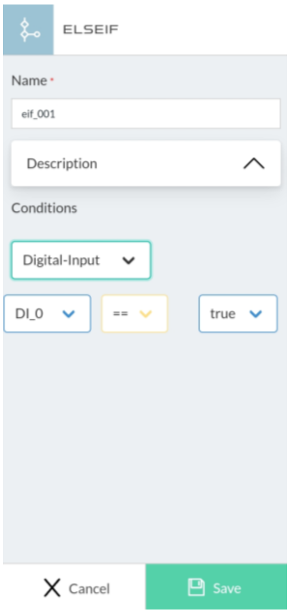
11.6.5 Else If

Add an else if function to the program. **ELSE IF** can only be added if there is already an **IF** statement within the program.

1. Edit name and add description if wanted.
2. Edit item and save settings.

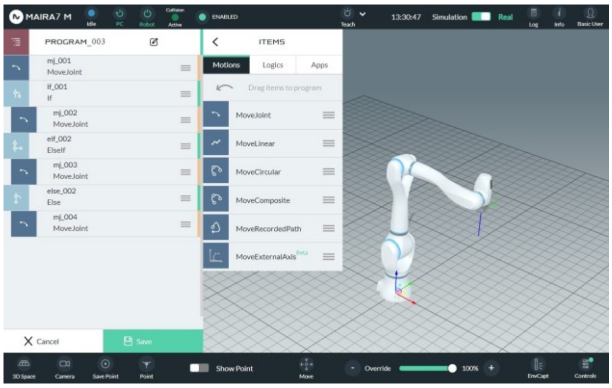
Item	Settings
Select an input	► Select an input type from the dropdown menu: **Digital-Input** **Analog-Input**
Input no.	► Select a numbered input from the dropdown menu: E.g. **DI_0**
Comparison operator	► Select the comparison operator from the dropdown menu:





	== means **equal** != means **not equal**
Boolean condition	► Select the Boolean condition from the dropdown menu: **true** or **false**

Instructions within the **ELSE IF** statement must be nested underneath the **ELSE IF block** within the program tree editor.

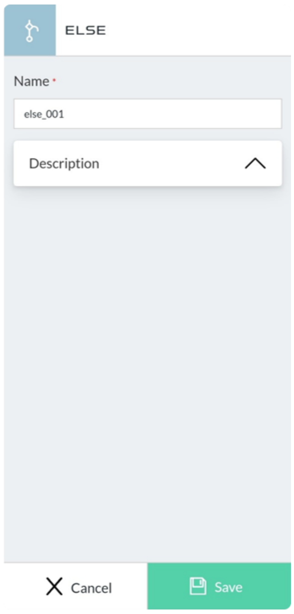


11.6.6 Else

Add an else function to the program. Note that for else function to work **if** must be created beforehand.

1. Edit name and add description if wanted.
2. Save settings.

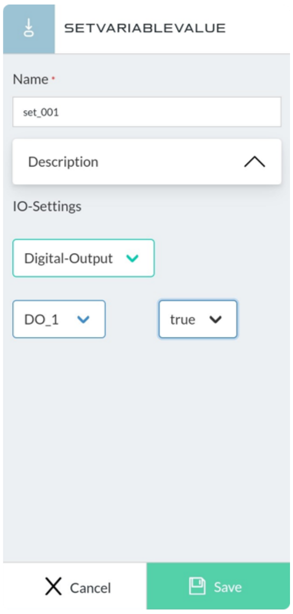
Instructions within the **ELSE** statement must be nested underneath the **ELSE block** within the program tree editor.



11.6.7 Set Variable Value

1. Edit name and add description if wanted.
2. Select a digital output from the dropdown menu and save settings.

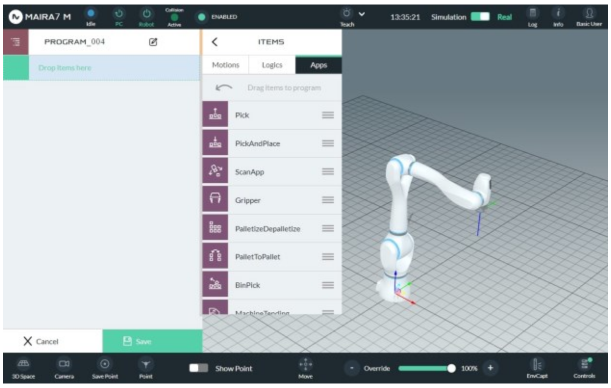
Item	Settings
IO-Settings	►Select a digital output from the dropdown menu. ►Set Output and respective status to true/false.



11.7 Apps

Apps provide guided automated processes (similar to wizards) that allow users to easily perform advanced program sequences. This app library includes a selection of application specific predefined apps. See the corresponding chapter for more information on the individual apps.

1. Drag and drop an app into the program tree.
2. To edit an item select it in the program tree and click the edit button.



11.7.1 Gripper

This App allows operators to use a gripper.

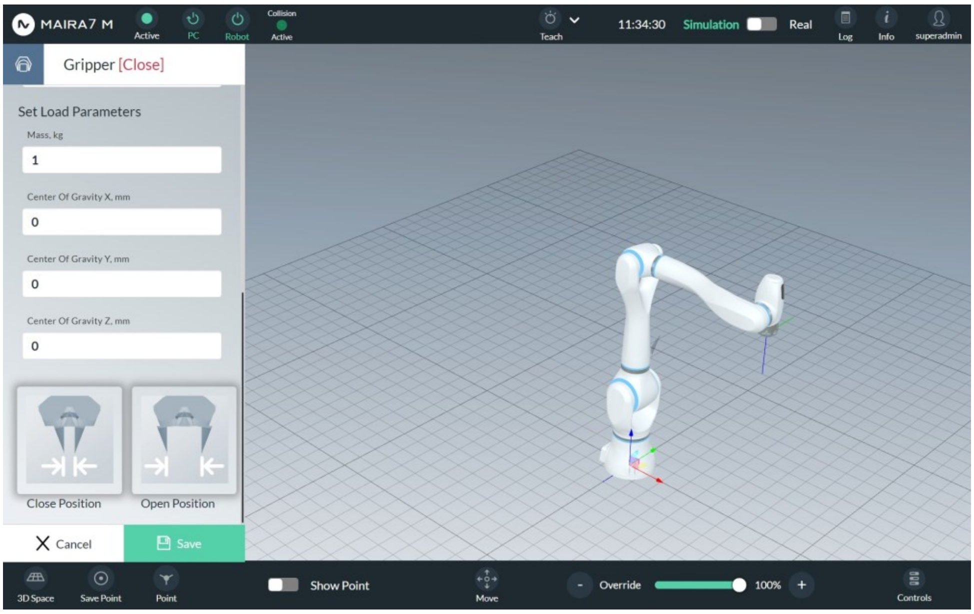
The Gripper app lets you use a gripper in your program.

- ► To use the app in your program, drag the app into your program tree.

To configure the app, follow these steps:

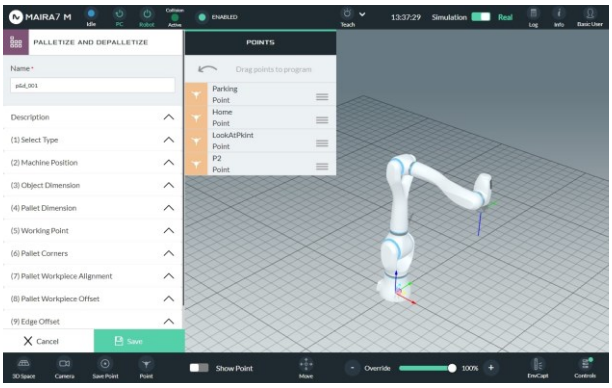
- ► Tap the app to select it.
- ► Tap the **Edit** icon.
- ► Set the parameters listed and explained below to configure the instruction according to your program needs.

Parameter	Explanation
Name	► Give the instruction a name or use the auto-generated one.
Description	► Give the instruction an optional description.
Select a Tool	🗐 Your gripper must be configured and set properly in the robot settings (see 15.2). ► Select your gripper from the dropdown list.
Gripper Parameters	
Wait Before, ms	► Specify a waiting time [ms] before the gripper action.
Close or Open	► Define the gripper action: ○ Close: select close, to close the gripper when the app is called in the program tree. ○ Open: select open, to open the gripper when the app is called in the program tree.
Wait After, ms	► Specify a waiting time [ms] after the gripper action.
Set Load Parameters	
Mass, kg	► Specify the mass [kg] of an object that will be grasped.
Center of Gravity X, mm	► Specify the center of mass [mm] of the object to be grasped in the X-direction.
Center of Gravity Y, mm	► Specify the center of mass [mm] of the object to be grasped in the Y-direction.
Center of Gravity Z, mm	► Specify the center of mass [mm] of the object to be grasped in the Z-direction.
Close Position	► Tap **Close Position** to directly manipulate the gripper and close it.
Open Position	► Tap **Open Position** to directly manipulate the gripper and open it.



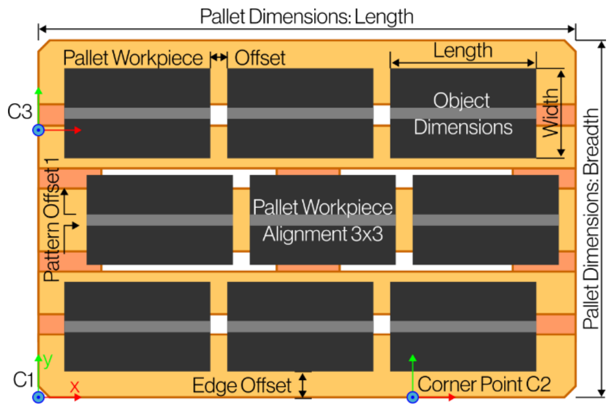
11.7.2 Palletize And Depalletize

The palletizing and depalletizing apps allow simple stacking and de-stacking of pallets. The user can specify the task, the relevant positions, dimensions and patterns by teaching or reusing existing points.



When adding a new instance of the palletizing and depalletizing app, a name and description can be entered. These parameters do not affect the actual execution of the app, but allow for identifying and reusing the parameterized app.

Palletizing and depalletizing require multiple parameters to provide enough flexibility for various types of pallets, object dimensions, stacking patterns and dimensions to ensure a stable stack of objects on the pallet. Refer to the dimension figure to the right for a visual description of the meaning of the different parameters.

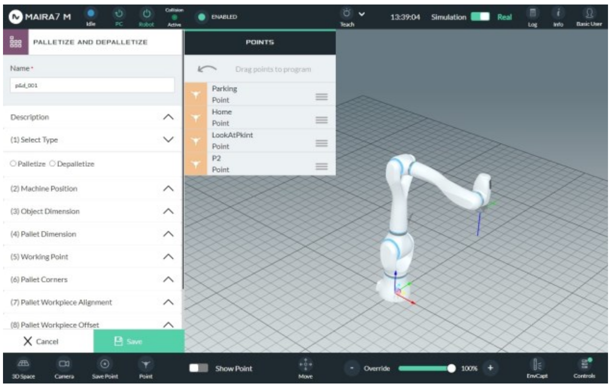
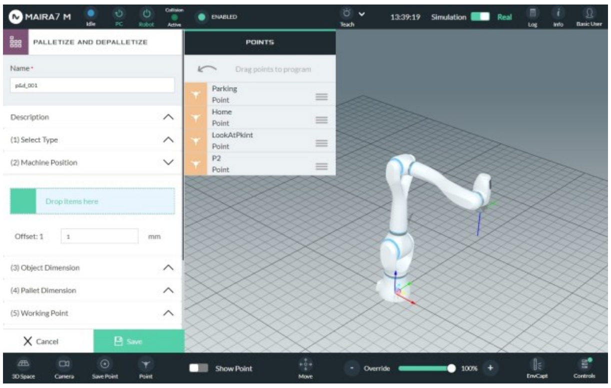


NOTICE	Please assure that the X-axis of the tool can be aligned with the chosen X-direction of the pallet by rotating the robot's joint 6. This ensures that required motions are possible and the tool is mounted to the robot's TCP in the correct orientation. If not, then change the X-direction of the pallet or reorient the tool.

When teaching points, please consider the following information:

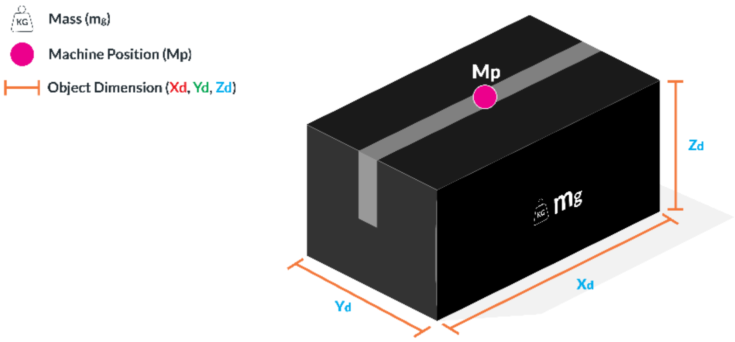
- The X-axis of the tool should be aligned parallel to the X-direction of the pallet.
- Points should be taught with the help of the actual palletizing/depalletizing objects.
- If using a vacuum gripper:
  - Grasp a palletizing object at its center and use ZeroG or jogging to move the robot to the point to be taught.
  - For pick and place operations, the height information of the taught points is used for grasping/releasing objects.
- If using a finger type gripper:
  - Close the gripper and use ZeroG or jogging to move the robot just above the center of the palletizing object.
  - For pick and place operations, the defined object height is used for grasping/releasing objects.
    - The gripper will go down half the object height from the Z-value of the taught point to grasp/release the objects.

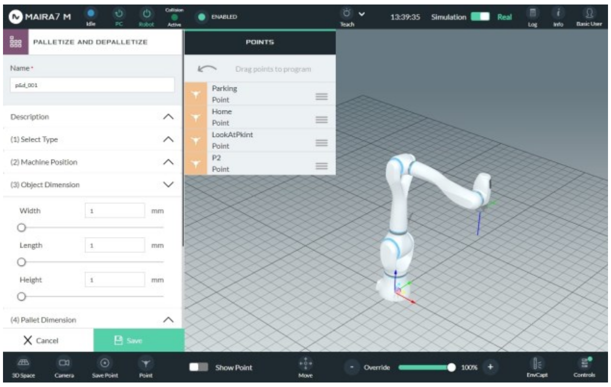
When teaching **Corner Points,** align the actual palletizing object to the respective corners, not the tool. The tool must be centered just above the objects as described before.

Item	Description
(1) Select Type	Select palletize or depalletize according to the intended task to perform.  **NOTICE** — Palletizing starts from the first object on the pallet. Depalletizing starts from the last object on the pallet.
(2) Machine Position	The objects to stack on the pallet arrive at a fixed point (objects to de-stack are discharged at a fixed point respectively). Teach the fixed provision point and the offset, which specifies the distance to lift the object from the pick/place position before the robot continues with the next motion.  **NOTICE** — The offset must be greater than the object's height.

(3) Object Dimensions

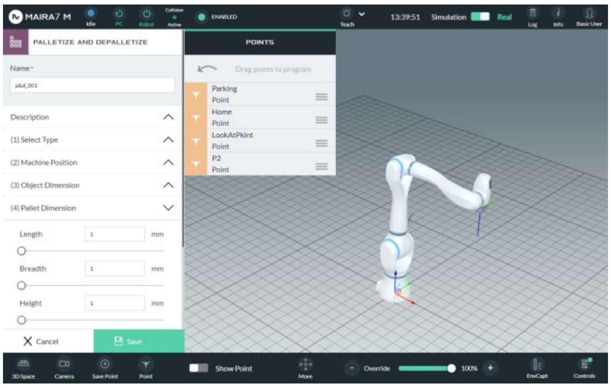
Specify the length (X-direction), width (Y-direction), and height (Z-direction) of the Objects.





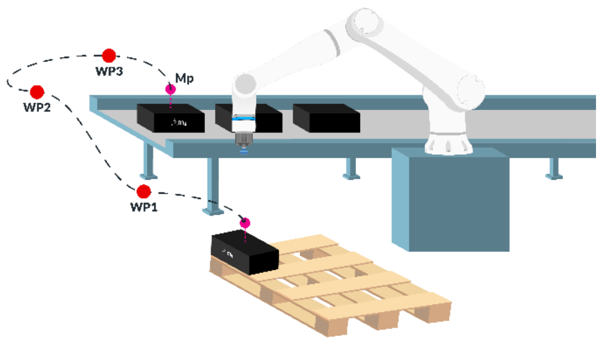
(4) Pallet Dimensions

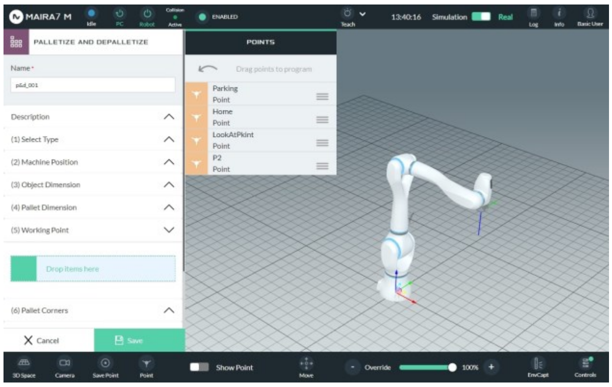
Define the length (X-direction), width (Y-direction), and height (Z-direction) of the pallet.



(5) Working Point

Define the transfer motion starting from the pallet to the Machine Position with one or multiple points. Ensure a collision-free path and enough height clearance to place the last layer.

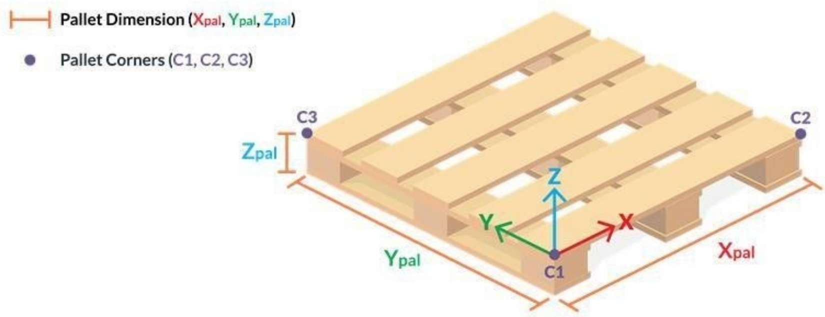


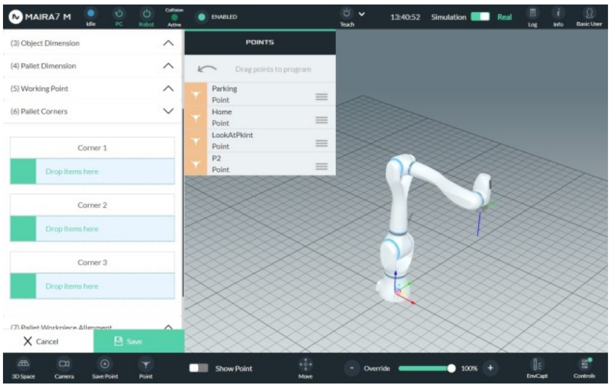


(6) Pallet Corners

To specify the position and orientation of the pallet, move the robot with the attached workpiece (see chapter introduction) to corner C1 and save the point. Move the robot along the length of the pallet and save point C2. Move the robot along the width of the pallet and save point C3.

- C1 (origin) and C2 in X-direction
- C3 in Y-direction

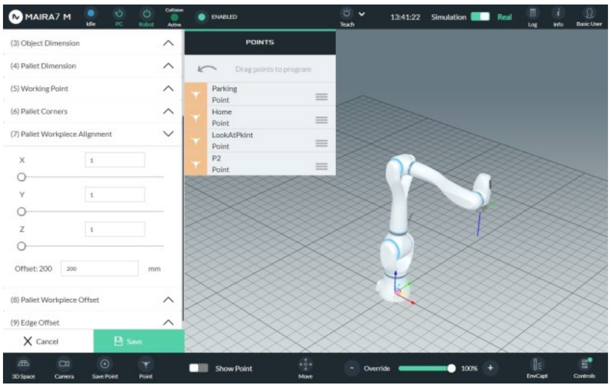




(7) Pallet Workpiece Alignment

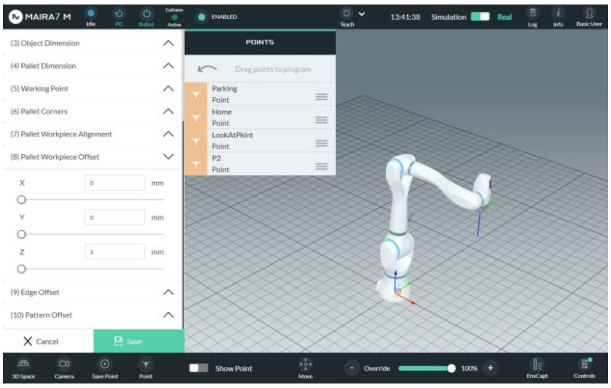
Specify the number of objects to arrange within one layer along the length (X-direction, columns), and the width (Y-direction, rows). Specify the number of layers (Z). Specify an offset for movements above objects.

- No. of rows * no. of columns * no. of layers = total no. of objects
- The height offset should be greater than the object height to avoid collisions. E.g. if object height is 100 mm, offset should be 120 or 150 mm.



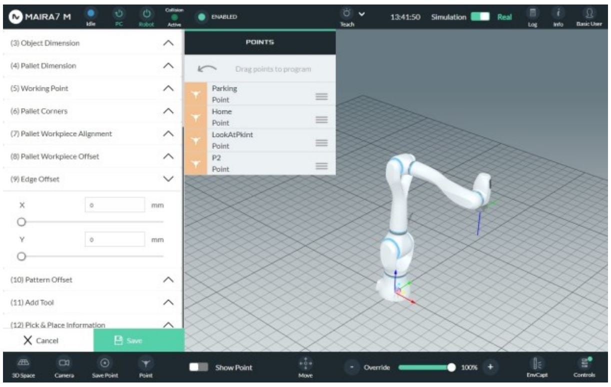
(8) Pallet Workpiece Offset

Specify the gaps in X-, Y-, and Z-direction between the objects. Ensure a precise measurement of the gaps between the objects for accurate picking and placing.



(9) Edge Offset

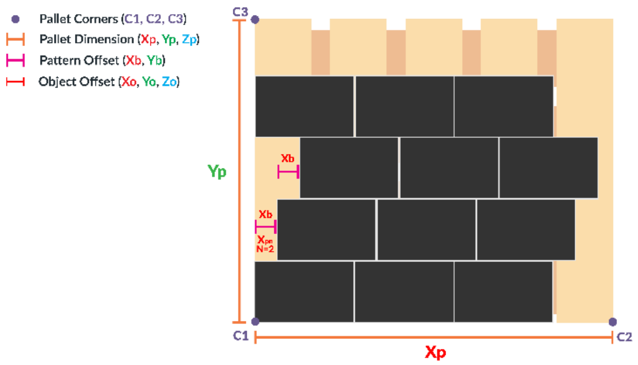
Specify the distance to the edge of the pallet. This value can be adjusted to center the objects on the pallet.

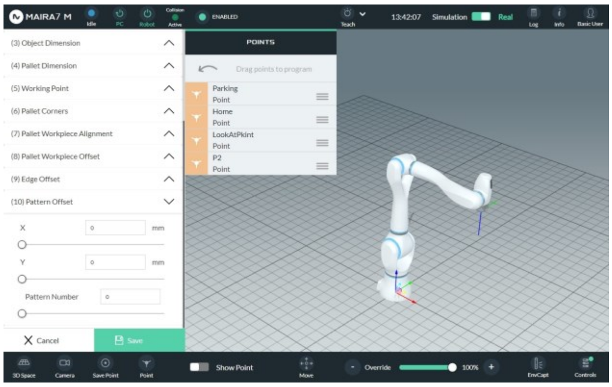


(10) Pattern Offset

For increased stability or complex patterns, the rows or columns of the pattern can be shifted. You can specify the shifting pattern by specifying the subsequent row shifts and the shifting amount per row and column. Depending on either specifying the X or Y shifting distance, the rows or columns will be shifted. In the example from the introduction, a shifting pattern with one subsequent row shift with a shifting distance in the X-direction is illustrated.

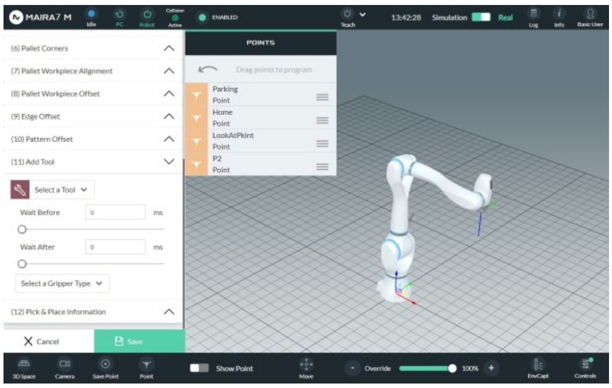
- Example for Pattern Numbers:
  - For the image below, the Pattern Number is 2.
  - Pattern in X direction is shifted by same offset 2 times (Xb) and returned to original pattern for the third row.
  - Pattern Numbers are possible only in one direction either X or Y.





(11) Add Tool

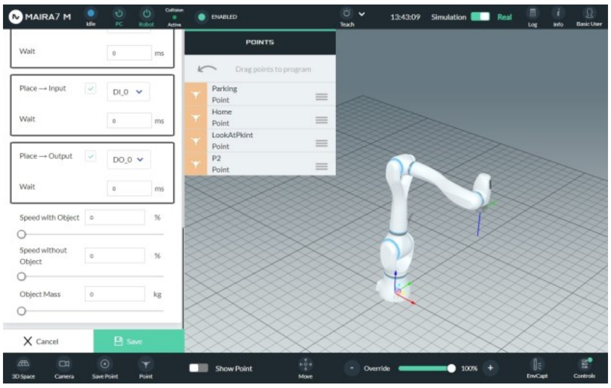
Add a tool to the app, specify wait times before picking and after placing. Specify the tool type, either Finger or Vacuum Gripper.



(12) Pick & Place Information

This section allows us to set input and output settings for orchestrating the pick and place operations. You can specify the inputs for being ready to pick and being ready to place the next item. You can specify the outputs indicating a picked and indicating a placed item. Additional wait times delay the motions of the robot based on the input signals and signal a finished operation by setting the outputs.

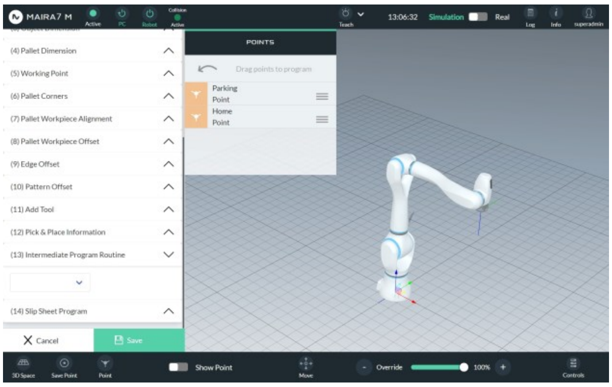
- Pick → Input Waits for a signal before picking
- Pick → Output Sends a signal after picking
- Place → Input Waits for a signal before placing
- Place → Output Sends a signal after placing
- Speed with Object Set the speed for when the robot is moving with the object
- Speed without Object Set the speed for when the robot is moving without the object
- Object mass Specify object mass



NOTICE	Please modify the Default settings. Speeds must be higher than 0%. I/Os must be deactivated if not used. Otherwise, the robot might wait for signals that never come.

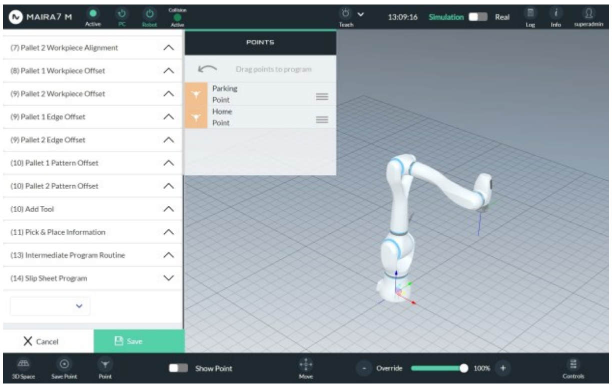
(13) Intermediate program routine

The user can define an intermediate program routine, which will be executed between the actual palletizing or depalletizing (picking or placing) step, e.g. to move around obstacles, reorient a grasped object or any other action to be performed in the transition between the palletizing/depalletizing sequence. The program needs to be created separately (**HOME** >> **Programs** >> **New Program**) and then selected from the dropdown menu within the intermediate program routine definition.



(14) Slip Sheet Program

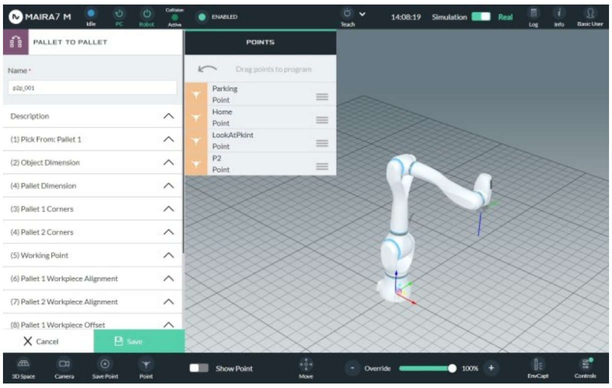
The user can define a slip sheet program, which will be executed after finishing one layer of a palletizing or depalletizing sequence, e.g. to place a separation layer in between two layers of palletized or depalletized goods. The program needs to be created separately (**HOME** >> **Programs** >> **New Program**) and then selected from the dropdown menu within the slip sheet program definition.



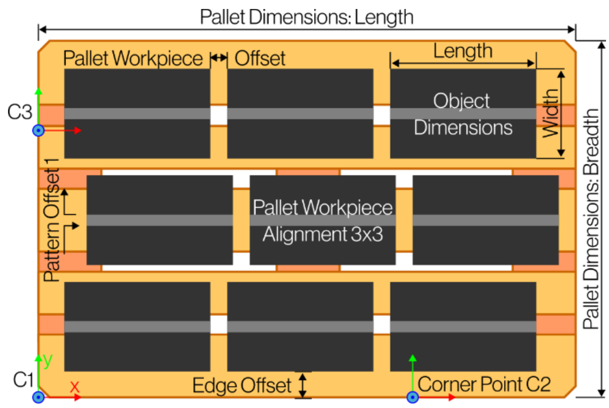
11.7.3 Pallet To Pallet

The pallet to pallet app allows to transfer objects from one pallet to another pallet. The user can specify the task, the
relevant positions, dimensions and patterns by teaching or
reusing existing points.

When adding a new Instance of the pallet-to-pallet app, a name and description can be entered. These parameters do not affect the actual execution of the app, but allow for identifying and reusing the parameterized app.



Transferring objects from one pallet to another requires defining multiple parameters to provide enough flexibility for various types of pallets, object dimensions, stacking patterns and dimensions to ensure operation without errors. Refer to the dimension figure to the right for a visual description of the meaning of the different parameters.

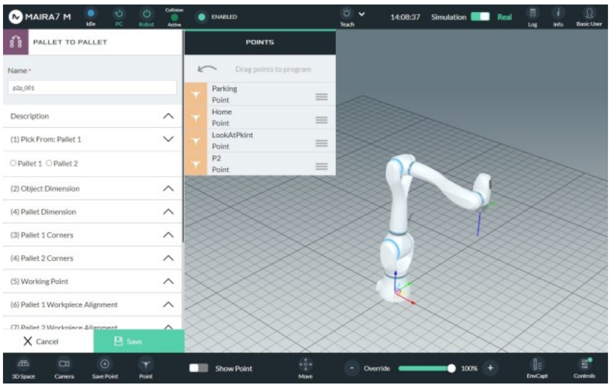


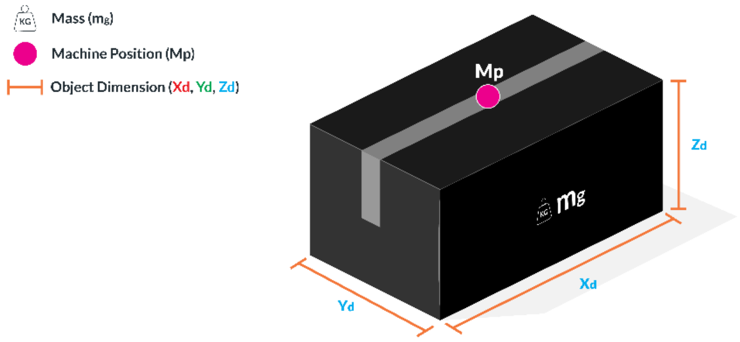
NOTICE	Please assure that the X-axis of the tool can be aligned with the chosen X-direction of the pallet by rotating the robot's joint 6. This ensures that required motions are possible and the tool is mounted to the robot's TCP in the correct orientation. If not, then change the X-direction of the pallet or reorient the tool.

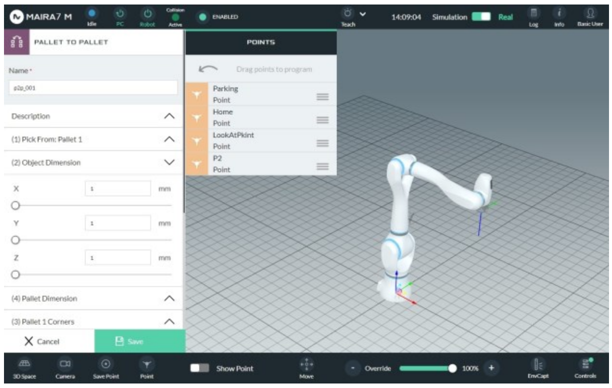
When teaching points, please consider the following information:**

- The X-axis of the tool should be aligned parallel to the X-direction of the pallet.
- Points should be taught with the help of the actual palletizing/depalletizing objects.
- If using a vacuum gripper:
  - Grasp a palletizing object at its center and use ZeroG or jogging to move the robot to the point to be taught.
  - For pick and place operations, the height information of the taught points is used for grasping/releasing objects.
- If using a finger type gripper:
  - Close the gripper and use ZeroG or jogging to move the robot just above the center of the palletizing object.
  - For pick and place operations, the defined object height is used for grasping/releasing objects.
    - The gripper will go down half the object height from the Z-value of the taught point to grasp/release the objects.

When teaching **Corner Points,** align the actual palletizing object to the respective corners, not the tool. The tool must be centered just above the objects as described before.

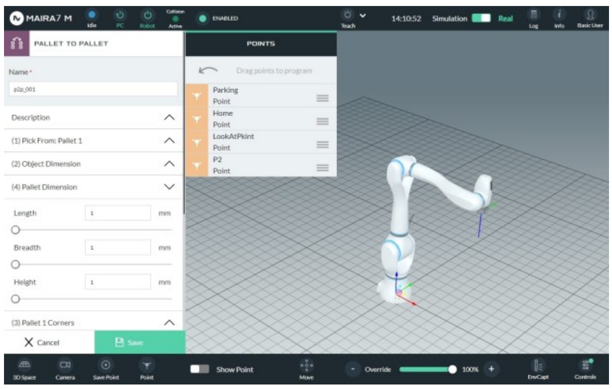
Item	Description
(1) Pick From	Select, if the objects are picked from pallet 1 or pallet 2. 
(2) Object Dimension	Specify the length (X-direction), width (Y-direction), and height (Z-direction) of the Objects.





(3) Pallet Dimension

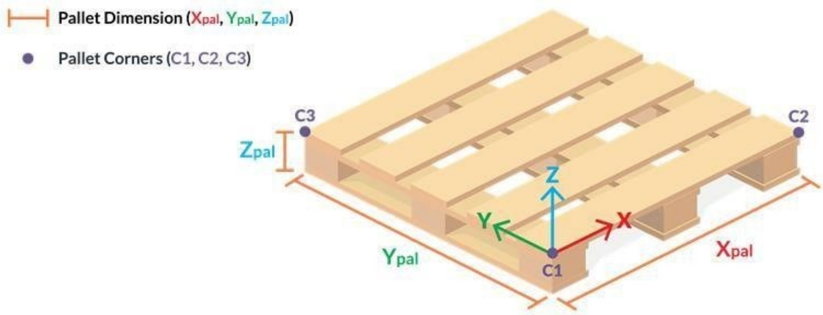
Define the length (X-direction), width (Y-direction), and height (Z-direction) of the pallet.

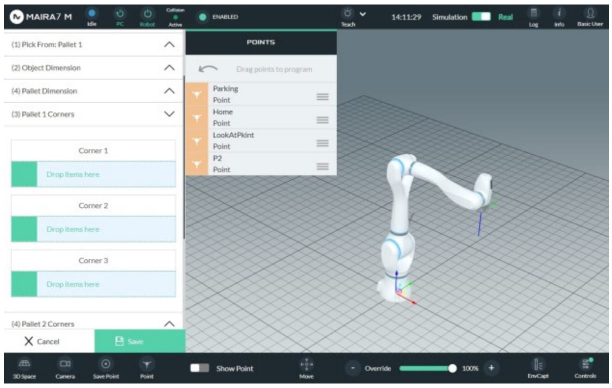


(4) Pallet 1 Corners

To specify the position and orientation of pallet 1, move the robot with the attached workpiece (see chapter introduction) to corner C1 and save the point. Move the robot along the length of the pallet and save point C2. Move the robot along the width of the pallet and save point C3.

- C1 (origin) and C2 in X-direction.
- C3 in Y-direction



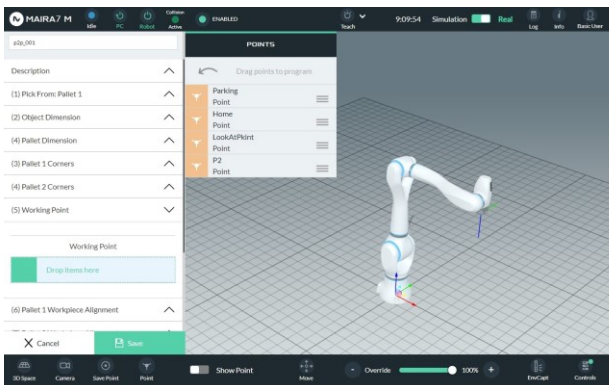


(5) Pallet 2 Corners

See (3)

(6) Working Point

Define the transfer motion between the pallets with one or multiple points. Ensure a collision-free path and enough height clearance to place the last layer.



(7) Pallet 1 Workpiece Alignment

Specify the number of objects to arrange on pallet 1 within one layer along the length (X-direction, columns), and the width (Y-direction, rows). Specify the number of layers (Z). Specify an offset for movements above objects.

- No. of rows * no. of columns * no. of layers = total no. of objects
- The height offset should be greater than the object height to avoid collisions. E.g. if object height is 100 mm, offset should be 120 or 150 mm.

(8) Pallet 2 Workpiece Alignment

See (6)

(9) Pallet 1 Workpiece Offset

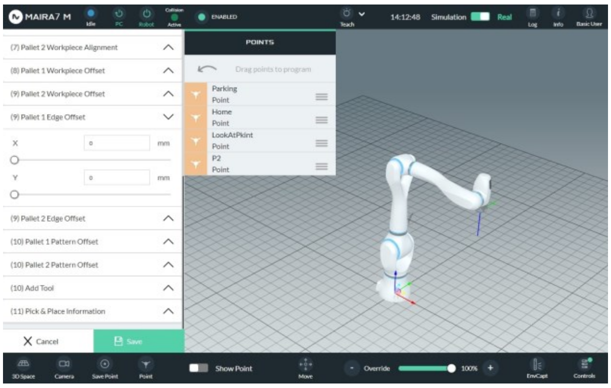
Specify the gaps in X-, Y-, and Z-direction between the objects for pallet 1. Ensure a precise measurement of the gaps between the objects for accurate picking and placing.

(10) Pallet 2 Workpiece Offset

See (8)

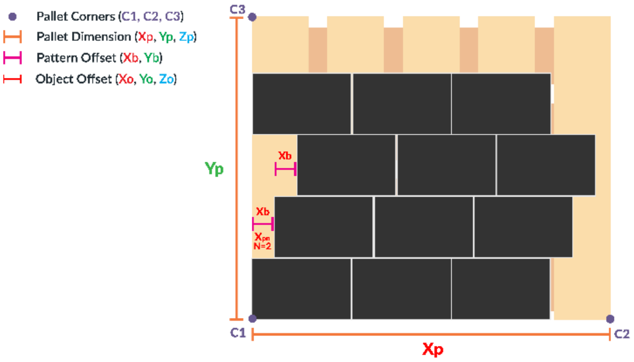
(11) Pallet 1 Edge Offset

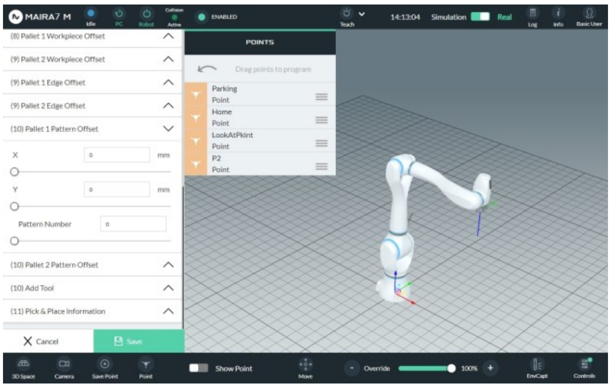
Specify the distance to the edge of pallet 1. This value can be adjusted to center the objects on the pallet.



(12) Pallet 2 Edge Offset	See (10)
(13) Pallet 1 Pattern Offset	For increased stability or complex patterns, the rows or columns of the pattern can be shifted. You can specify the shifting pattern by specifying the subsequent row shifts and the shifting amount per row and column. Depending on either specifying the X or Y shifting distance, the rows or columns will be shifted. In the example from the introduction, a shifting pattern with one subsequent row shift with a shifting distance in the X-direction is illustrated.

- Example for Pattern Numbers:
  - For the image below, the Pattern Number is 2.
  - Pattern in X direction is shifted by same offset 2 times and returned to original pattern for the third row.
  - Pattern Numbers are possible only in one direction either X or Y.





(14) Pallet 2 Pattern Offset

See (12)

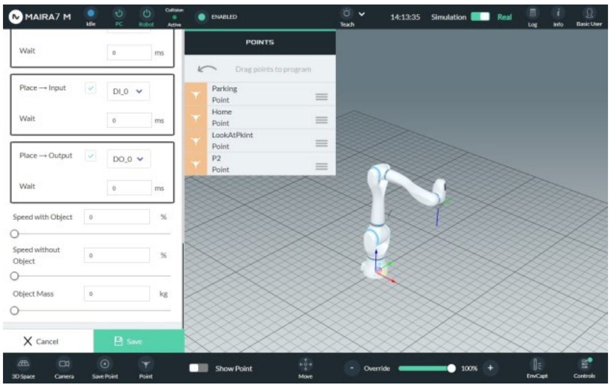
(15) Add Tool

Add a tool to the app, specify wait times before picking and after placing. Specify the tool type, either Finger or Vacuum Gripper.

(16) Pick & Place Information

This section allows us to set input and output settings for orchestrating the pick and place operations. You can specify the inputs for being ready to pick and being ready to place the next item. You can specify the outputs indicating a picked and indicating a placed item. Additional wait times delay the motions of the robot based on the input signals and signal a finished operation by setting the outputs.

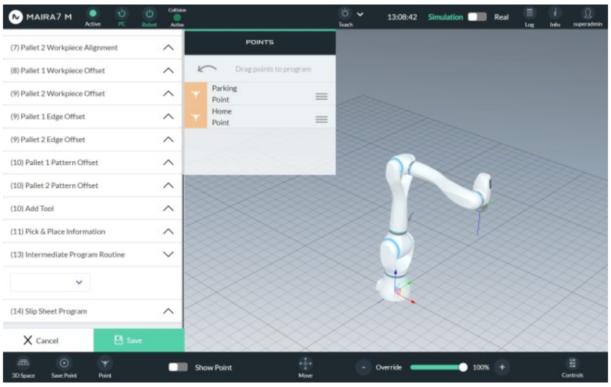
- Pick → Input Waits for a signal before picking
- Pick → Output Sends a signal after picking
- Place → Input Waits for a signal before placing
- Place → Output Sends a signal after placing
- Speed with Object moving Set the speed for when the robot is with the object
- Speed without Object moving Set the speed for when the robot is without the object
- Object mass Specify object mass



NOTICE	Please modify the Default settings. Speeds must be higher than 0%. I/Os must be deactivated if not used. Otherwise, the robot might wait for signals that never come.

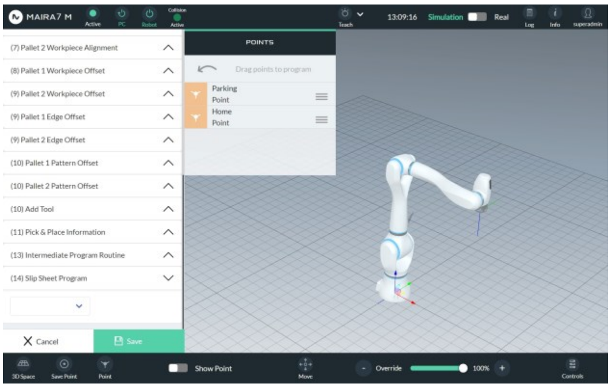
(17) Intermediate program routine

The user can define an intermediate program routine, which will be executed between the actual palletizing or depalletizing (picking or placing) step, e.g. to move around obstacles, reorient a grasped object or any other action to be performed in the transition between the palletizing/depalletizing sequence. The program needs to be created separately (**HOME** >> **Programs** >> **New Program**) and then selected from the dropdown menu within the intermediate program routine definition.



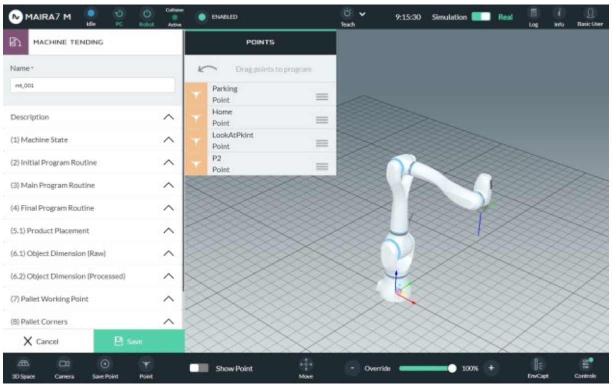
(18) Slip Sheet Program

The user can define a slip sheet program, which will be executed after finishing one layer of a palletizing or depalletizing sequence, e.g. to place a separation layer in between two layers of palletized or depalletized goods. The program needs to be created separately (**HOME** >> **Programs** >> **New Program**) and then selected from the dropdown menu within the slip sheet program definition.



11.7.4 Machine Tending

The machine tending app provides the functionality to load stock material into and unload finished parts from a machine.
To provide the necessary level of flexibility, the machine tending app allows us to create initial and final routines for the execution before and after a batch of material and to pick and place the stock material from and to a pallet.



The actions of the robot are synchronized with the machining cycle and allow for fast and efficient production runs. The settings and parameters necessary to create a machine tending application are described in this section.

When adding a new Instance of the machine tending app, a name and description can be entered. These parameters do not affect the actual execution of the app, but allow for identifying and reusing the parameterized app.

When teaching points, please consider the following information:

NOTICE	Please assure that the X-axis of the tool can be aligned with the chosen X-direction of the pallet by rotating the robot's joint 6. This ensures that required motions are possible and the tool is mounted to the robot's TCP in the correct orientation. If not, then change the X-direction of the pallet or reorient the tool.

- The X-axis of the tool should be aligned parallel to the X-direction of the pallet.
- Points should be taught with the help of the actual objects.
- Close the gripper and use ZeroG or jogging to move the robot just above the center of an object.
- For pick and place operations, the defined object height is used for grasping/releasing objects.
  - The gripper will go down half the object height from the Z-value of the taught point to grasp/release the objects.

When teaching **Corner Points,** align the actual object to the respective corners, not the tool. The tool must be centered just above the objects as described before.

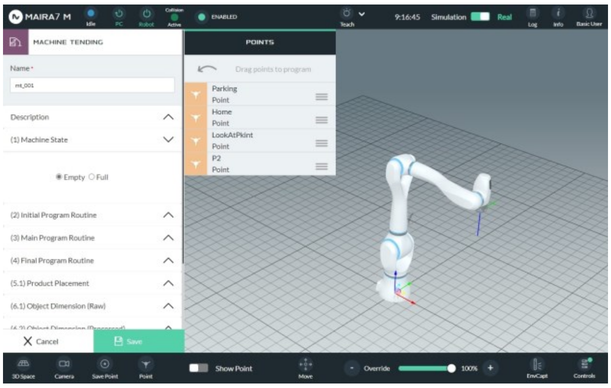
Item	Description

(1) Machine State

Select if the machine is empty or full when initializing the machine tending application.

If the initial state of the machine is set to **Empty**, the robot checks if there are any raw products present on the pallet. If raw products are present, the robot will pick them up and transport them to the designated working point.

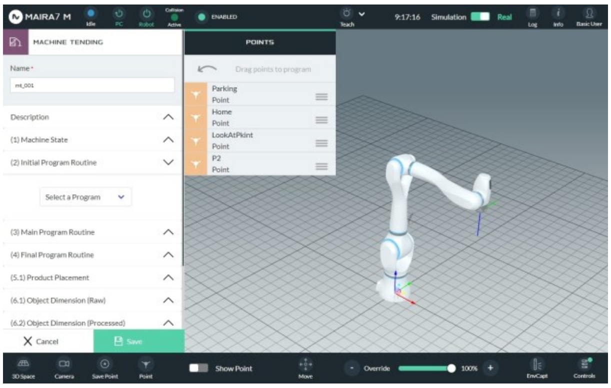
When the machine starts in a **Full** state. The robot will first travel from its designated working point to the machine, where it will interact with the processed part. Afterwards, it will return to its original working point while carrying the processed part.



(2) Initial Program Routine

The user can define an initial starting program routine, which accounts for the state of the machining process when initiating the machine tending application (see (1)). Create a program which extracts the last part from the machine if the machine is **Full** on initiation. Create a starting sequence which puts an unprocessed part into the machine if the machine is **Empty** on initiation.

The program needs to be created separately (**HOME** / **Programs** / **New Program**) and then selected from the dropdown menu within the **Initial Program Routine** definition.



NOTICE

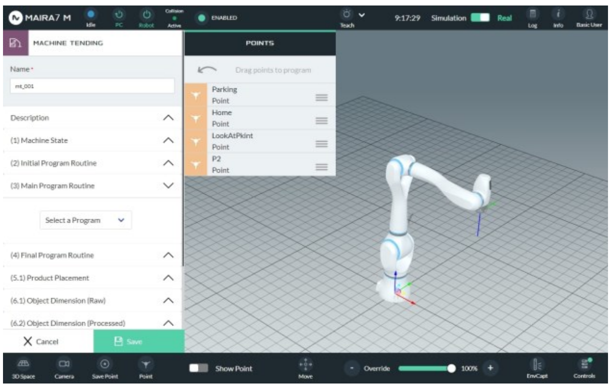
If the Initial Program Routine is used to place the first stock part into an **Empty** machine, (11) Number of Starting Pieces must be set to 2 then. Otherwise, the robot will move to the first (already empty) position on the object pallet within the **Main Program Routine**.

Best practice:
Ideally the Initial Program Routine is used to bring the setup into a state from where it's able to perform its tasks in a loop (Main Program Routine). The Final Program Routine is then used to revert to the initial state. This way, Initial and Final Program Routine could be as simple as joint motions to the Working Point and IO status validation.

(3) Main Program Routine

This program routine is executed for each part during the batch machining process (except for the first and last parts if they are handled by Initial or Final Program Routine).

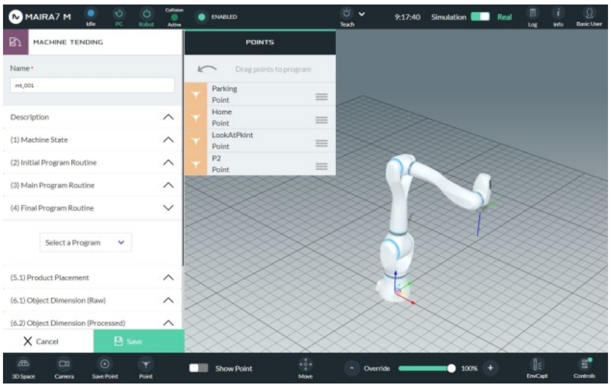
The program needs to be created separately (**HOME / Programs / New Program**) and then selected from the dropdown menu within the **Main Program Routine** definition.



(4) Final Program Routine

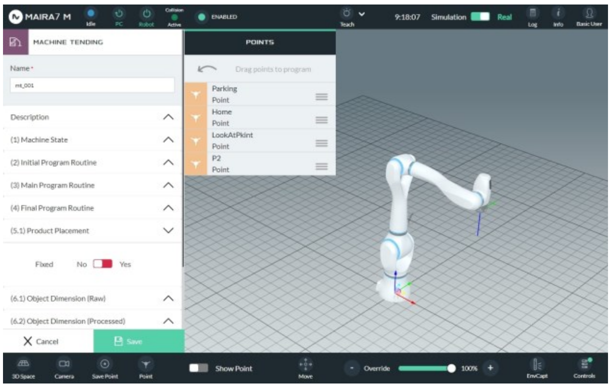
After the batch machining process has been executed, the final program routine is called. This final routine can include moving the robot to the working point or signaling the finished batch via an IO signal.

The program needs to be created separately (**HOME** / **Programs** / **New Program**) and then selected from the dropdown menu within the **Final Program Routine** definition.



(5) Product Placement

After the processing of the part, the part is placed onto a fixed location or back onto the pallet. Select **Yes** if you have included the fixed location in the previously defined program routines. Select **No**, if the part is placed back onto the original pallet location.



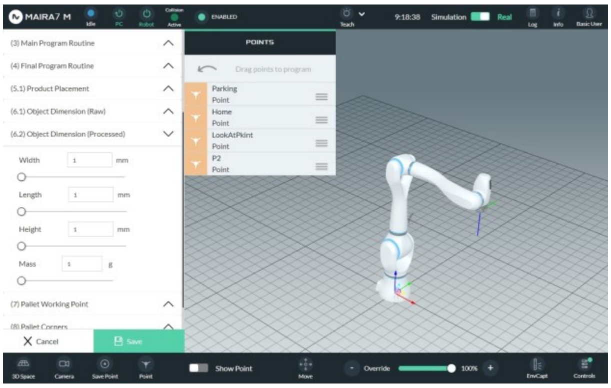
(6.1) Object Dimensions (Raw)

Define the length, width, height and mass of the unprocessed raw object.


MAIRA7 M
Idle
PC
Robot
Collision
Active
ENABLED
Teach
9:18:25
Simulation
Real
Log
Info
Basic User
(1) Machine State
(2) Initial Program Routine
(3) Main Program Routine
(4) Final Program Routine
(5.1) Product Placement
(6.1) Object Dimension (Raw)
Width
1
mm
Length
1
mm
Height
1
mm
Mass
1
g
Cancel
Save
POINTS
Drag points to program
Parking
Point
Home
Point
LookAtPkint
Point
P2
Point
3D Space
Camera
Save Point
Point
Show Point
Move
Override
100%
EnvCapt
Controls

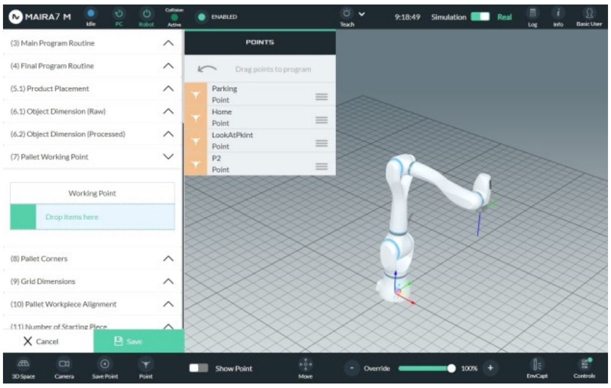
(6.2) Object Dimensions (Processed)

Define the length, width, height and mass of the processed object.



(7) Pallet Working Point

Define a safe point above the pallet, which is approached before and after each tending operation. Ensure that it is identical to the starting point of the Initial Program Routine and to the end point of the Final Program Routine.

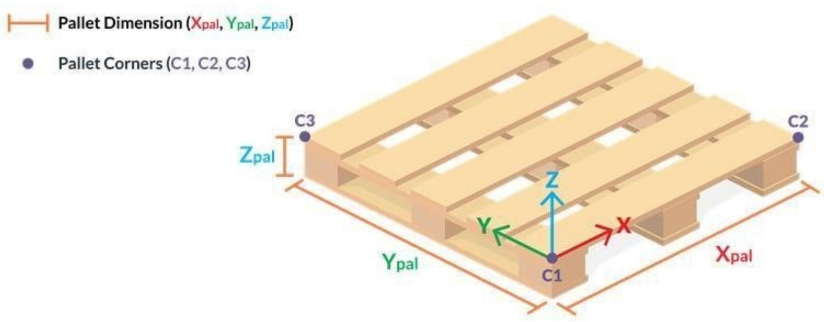


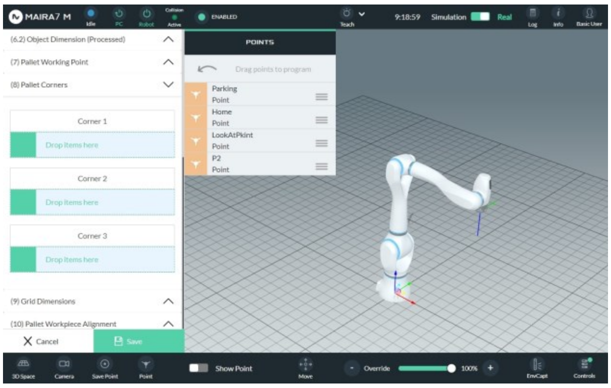
NOTICE	The motion between the working point and picking/placing of objects is automatically generated. The motion between the working point and picking/placing from/into the machine is to be created by the user through the corresponding program routines.

(8) Pallet Corners

To specify the position and orientation of the pallet, move the robot with the attached workpiece (see chapter introduction) to corner C1 and save the point. Move the robot along the length of the pallet and save point C2. Move the robot along the width of the pallet and save point C3.

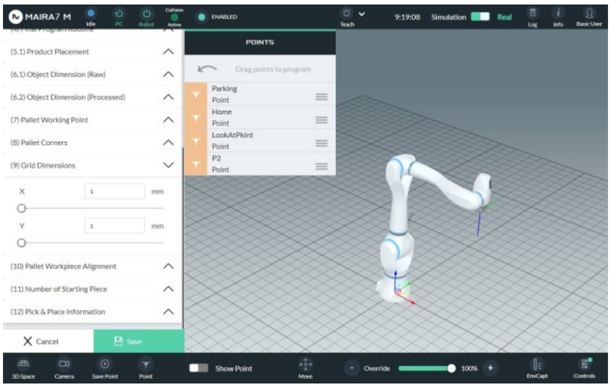
- C1 (origin) and C2 in X-direction.
- C3 in Y-direction





(9) Grid Dimensions

Specify the dimensions of the spacing between items on the pallet. Grid dimensions can be similar to or bigger than the (Raw) object dimensions.

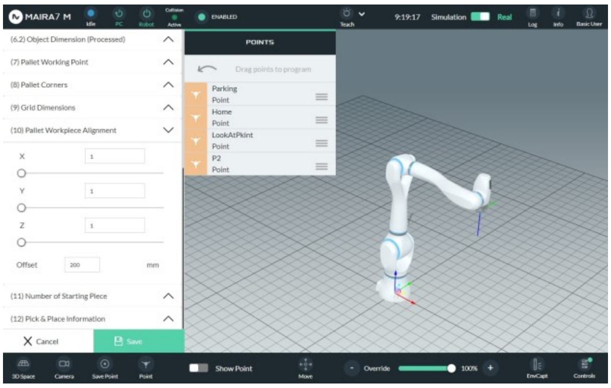


NOTICE	For objects smaller than the grid elements, the robot picks the parts from each grid element's corner closest to C1, hence they must be positioned accordingly. After machining, the robot places the parts back at the center of each grid element.

(10) Pallet Workpiece Alignment

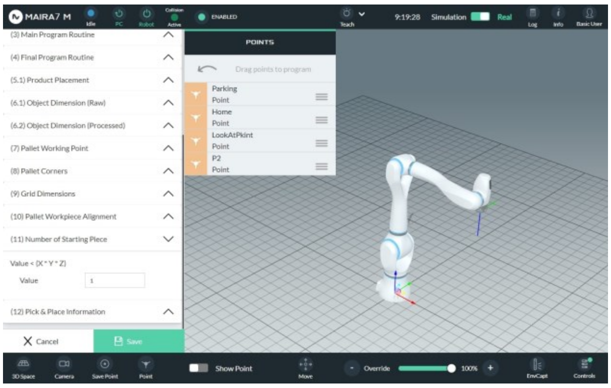
Specify the number of objects to arrange within one layer along the length (X-direction, columns), and the width (Y-direction, rows). Specify the number of layers (Z). Specify an offset for movements above objects.

- No. of rows * no. of columns * no. of layers = total no. of objects
- The height offset should be greater than the object height to avoid collisions. E.g. if object height is 100 mm, offset should be 120 or 150 mm.



(11) Number of Starting Pieces

Specify the number of the object to start the machine tending sequence. It should be greater or equal than 1 (first item) and lower or equal to the last item to process.



NOTICE	If the Initial Program Routine is used to place the first stock part into an **Empty** machine, the Number of Starting Pieces must be set to 2 then (see (2)). Otherwise, the robot will move to the first (already empty) position on the object pallet within the **Main Program Routine**.

(12) Pick and Place Information

This section allows us to set input and output settings for orchestrating the pick and place operations. You can specify the inputs for being ready to pick and being ready to place the next item. You can specify the outputs indicating a picked and indicating a placed item. Additional wait times delay the motions of the robot based on the input signals and signal a finished operation by setting the outputs.

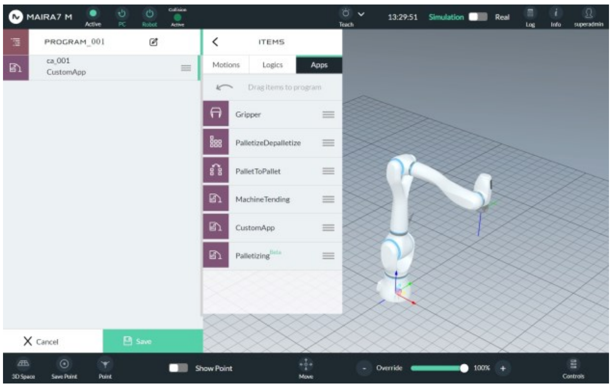
- Pick → Input Waits for a signal before picking
- Pick → Output Sends a signal after picking
- Place → Input Waits for a signal before placing
- Place → Output Sends a signal after placing

NOTICE	Please modify the Default settings. I/Os must be deactivated if not used. Otherwise, the robot might wait for signals that never come.

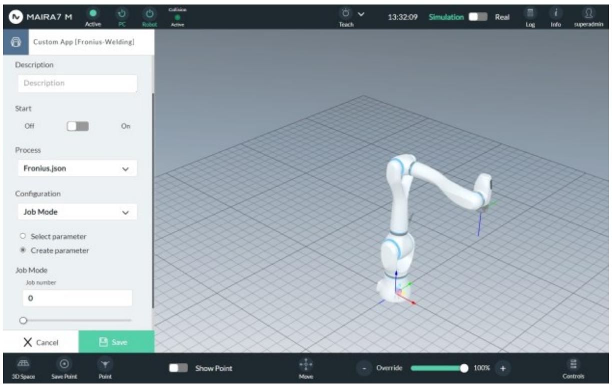
11.7.5 CustomApp

The CustomApp provides quick access to external tools within your programs.

1. Edit **Name** and add **Description** if wanted.
2. Set the action to **Start** or **Stop** with the slider by toggling between Off (Stop) <> On (Start).
3. Choose a **Configuration file** from the dropdown menu.
   - Configuration files can be imported via the **External Source** tab within the **Settings** menu (see 15.14).



- Configuration files are used to establish and handle communication with external devices, e.g. welding power sources.



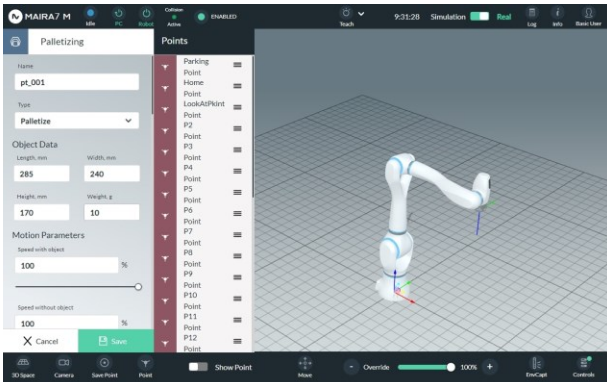
4. Choose a **Configuration** from the dropdown menu.
   - Configurations are defined within the configuration files.
5. Enter the parameters required for the selected configuration.
   - Parameters are also defined within the configuration files.

   a. They can be entered manually (**Create parameter**) or parameter sets can be defined and automatically loaded (**Select parameter**) (also see 15.14 External Source).
6. Proceed with **Next.**
7. **Save** your **CustomApp** and continue with your outer program logic.

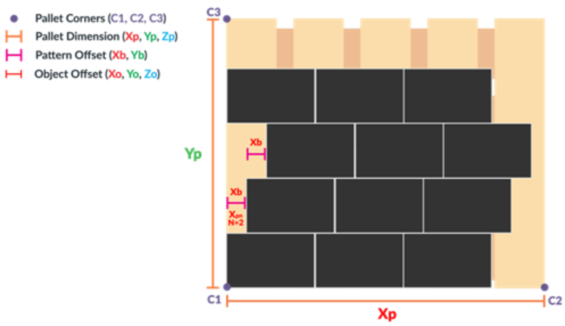
11.7.6 Palletizing[Beta]

The **Palletizing**[Beta] app allows simple stacking and de-stacking of pallets.

The user can specify the task, the relevant positions, dimensions and patterns by teaching or reusing existing points.



Refer to the dimension figure to the right for a visual description regarding the meaning of the different parameters.



NOTICE	Please assure that the X-axis of the tool can be aligned with the chosen X-direction of the pallet by rotating the robot's joint 6. This ensures that required motions are possible and the tool is mounted to the robot's TCP in the correct orientation. If not, then change the X-direction of the pallet or reorient the tool.

When teaching points for your palletizing application, please consider the following information:

- The X-axis of the tool should be aligned parallel to the X-direction of the pallet.

- Points should be taught with the help of the actual palletizing/depalletizing objects.
- If using a vacuum gripper:
  - Grasp a palletizing object at its center and use ZeroG or jogging to move the robot to the point to be taught.
  - For pick and place operations, the height information of the taught points is used for grasping/releasing objects.
- If using a finger type gripper:
  - Close the gripper and use ZeroG or jogging to move the robot just above the center of the palletizing object.
  - For pick and place operations, the defined object height is used for grasping/releasing objects.
    - The gripper will go down half the object height from the Z-value of the taught point to grasp/release the objects.

When teaching **Corner Points,** align the actual palletizing object to the respective corners, not the tool. The tool must be centered just above the objects as described before.

► Follow the steps below to provide your palletizing application with the necessary information

  ○ Scroll down within the left menu pane to see all the available configuration options

    ▪ The table below explains all options from top to bottom

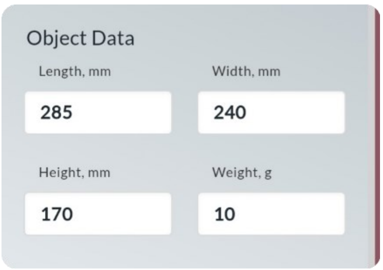
Action	Description
► Enter a **Name** for you palletizing application or leave the generic one ► Select your application **Type** from the dropdown menu ○ **Palletizing** ○ **Depalletizing**	

NOTICE	The name does not affect the actual app execution, it only allows for identifying and reusing the parameterized app.

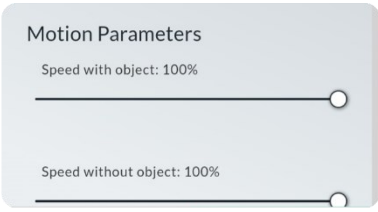
NOTICE	For Palletizing, the first item placed on the pallet will be near the chosen origin of the pallet.

NOTICE	For Depalletizing, the first item picked from the pallet will be near the corner diagonally opposite to the chosen origin.

- Specify the **Object Data** (objects to be palletized or depalletized) by entering the corresponding values
  - **Length** [mm]
  - **Width** [mm]
  - **Height** [mm]
  - **Weight** [g]

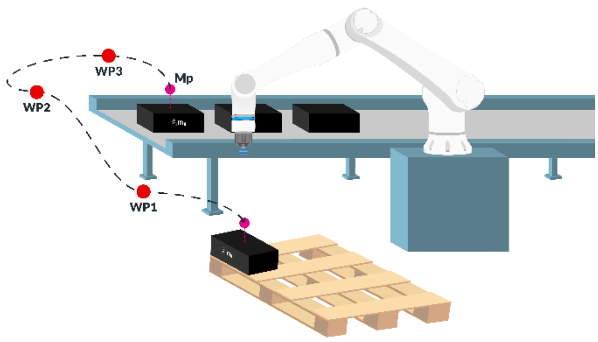


- Define the **Motion Parameters**
- Use the sliders to adjust the speeds
  - **Speed of the robot with an object** [%]
  - **Speed of the robot without an object** [%]



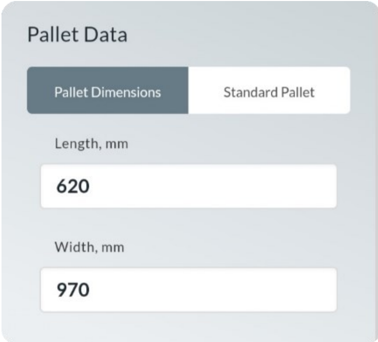
- Specify the **Working Points**

The working points define the transfer motion starting from the pallet to the Machine Position with one or multiple points (see the image to the right). Ensure a collision-free path and enough height clearance to place the last layer of the palletizing objects.

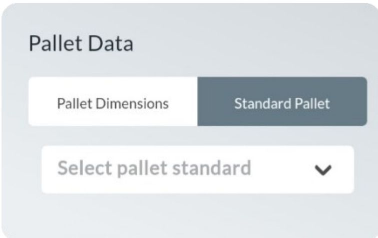


- ▶ Drag and drop points from your points list or teach new points

- ▶ Specify the **Pallet Data**
- ▶ Select **Pallet Dimensions** to enter custom values for
  - **Length** [mm]
  - **Width** [mm]



- ▶ Select **Standard Pallet** to choose from a list of standard dimensions provided in the dropdown menu



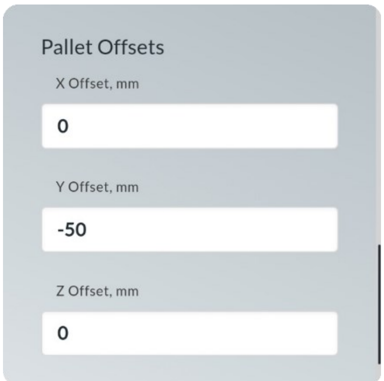
NOTICE	**Length** corresponds to the **X** direction of the chosen pallet frame.

NOTICE	**Width** corresponds to the **Y** direction of the chosen pallet frame.

- ▶ Define your **Pallet Offsets**

Define how the robot should approach the pallet items by specifying directional offsets.

- ▶ Enter the corresponding values
  - **X Offset** [mm]
    - Positive or negative to the pallet frame



- Y Offset [mm]
  - Positive or negative to the pallet frame
- **Z offset** [mm]
  - Positive to the pallet frame

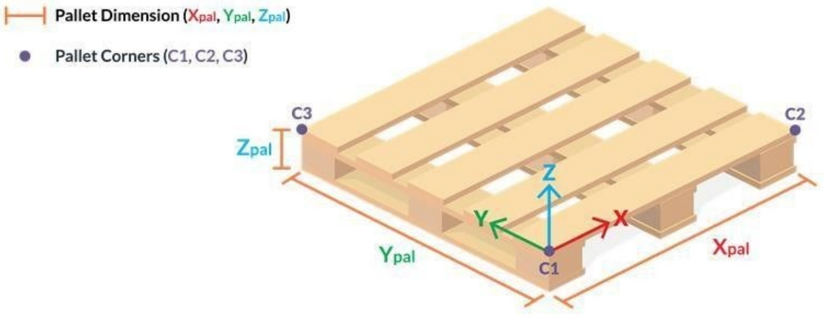
▶ Specify the pallet orientation with **Pallet Corners**

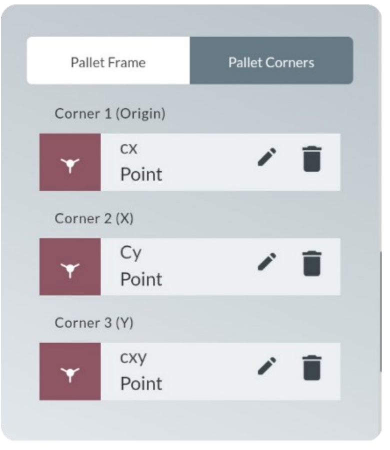
To specify the position and orientation of a pallet, move the robot (**ZeroG** or **Cartesian Jogging**) to corner **C1** and save the point. Move the robot along the length of the pallet and save point **C2**. Move the robot along the width of the pallet and save point **C3**.

NOTICE	Please refer to the section **"When teaching points, please consider the following information:"** from the beginning of this chapter

▶ Drag and drop points from your points list or teach new points

- **C1** (origin) and **C2** in X-direction
- **C3** in Y-direction

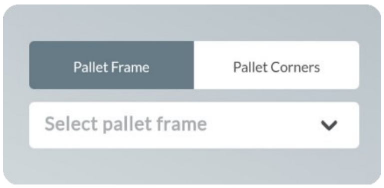




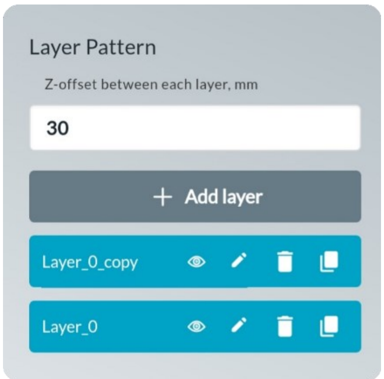
Instead of using **Pallet Corners** to specify pallet orientation, you can also use **Pallet Frames**. However, the frame teaching methodology is

- Alternatively, specify the pallet orientation a **Pallet Frame**

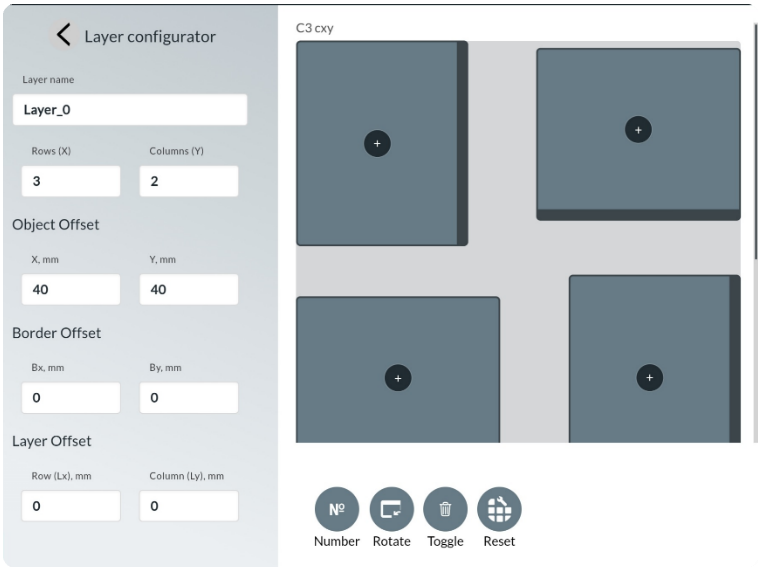
like the corner teaching method. It's just a way of reusing frames instead of three saved points.



- Define the **Layer Pattern**
- Specify a vertical **Z-offset** [mm] between layers
  - Use 0 (zero) if items are directly placed on top of each other
- Click the **Add Layer** button to add layers
- Use the icons on a layer to perform one of the following operations
  - View
  - Edit
  - Delete
  - Copy

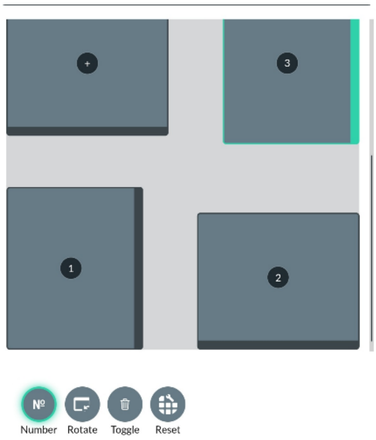


- Click the **Edit** button to configure a layer with the **Layer configurator**

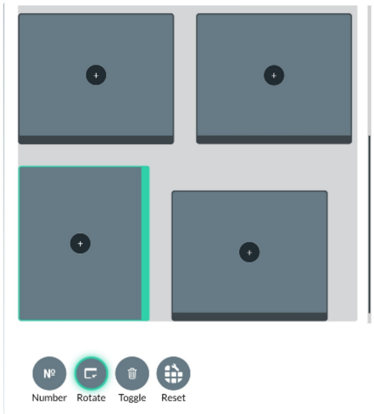


- The left menu pane allows you to specify the layer in general
  - **Layer Name** (optional)
  - **Rows** and **Columns** specify the number of rows and columns
  - **Object Offset** specifies the offset between each pallet item
  - **Border Offset** specifies the offset to the edge of the pallet
  - **Layer Offset** can be used to offset an entire row or column by the given value
- The right menu pane shows the actual pallet layout and allows to modify the layout or individual items as described below

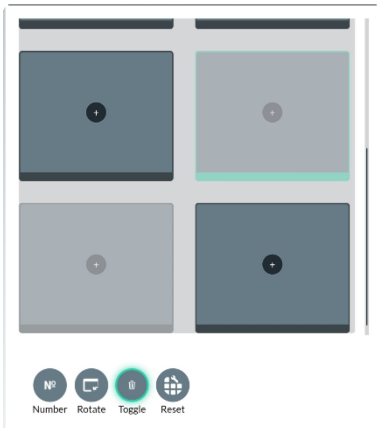
- The **Number** button enables you to create custom sequences



- The **Rotate** button rotates individual items on the grid by 90 degrees

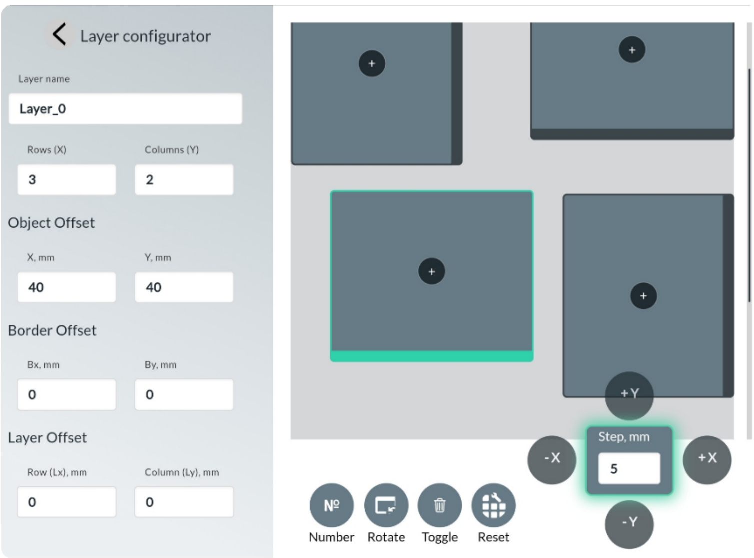


- The **Toggle** button hides or shows individual items on the grid
  - Hidden items will not be picked or placed

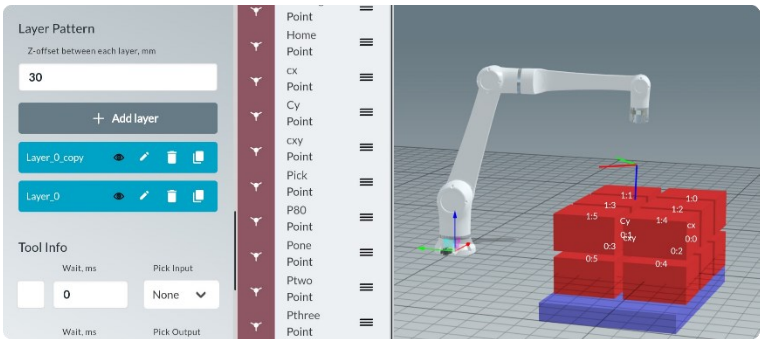


- The **Reset** button resets all changes

- You can also select an individual item to move it in x or y direction
  - This overrides the before specified offsets for the particular item



- After configuring/editing a layer, the 3D visualization in the GUI will show it accordingly[5]



- Provide the **Tool Info** to configure application specific I/O signals and or waiting times
- Use the checkboxes to select whether the signal should be True or False
  - Checked means True
  - Unchecked means False
- Enter a **Wait** time [ms] if desired
- Specify an I/O port via the corresponding dropdown menu

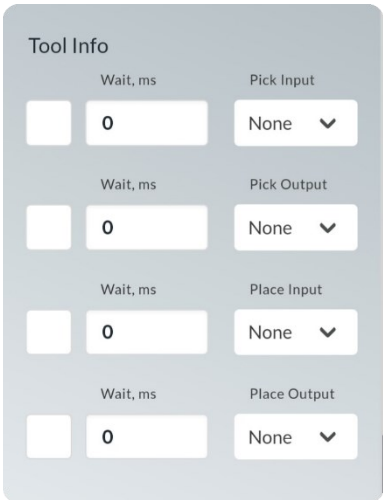
NOTICE	If no external peripherals are being used, then select None

Pick Input: waits for a signal before picking and waits after receiving the signal if time is specified.

Pick Output: Sends a signal after picking and waits before sending the signal if time is specified.

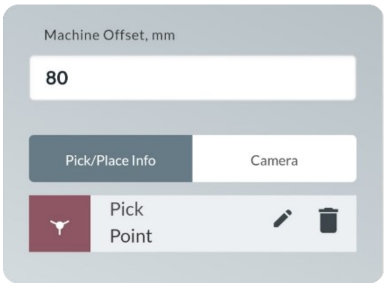
Place Input: Waits for a signal before placing and waits after receiving the signal if time is specified.

Place Output: Sends a signal after placing and waits before sending the signal if time is specified.



[5] Screenshot shows LARA, same for MAiRA.

- Specify the **Pick/Place Info**
- Enter a value for **Machine Offset** [mm] to specify the approach offset in the Z direction for the Pick/Place location
- Drag and drop a point from your points list or teach a new point from where items will be picked (in case of palletizing) or placed (in case of depalletizing)
- Alternatively, use a **Camera** to identify a Pick/Place point **(FUTURE RELEASE)**

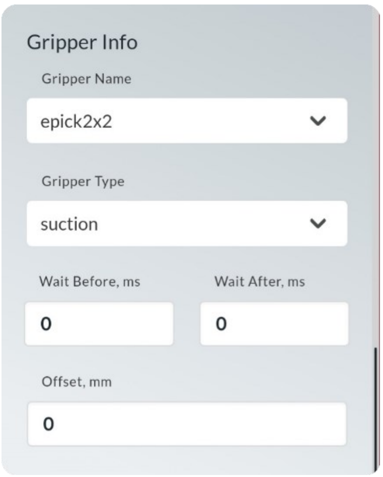


- Provide the **Gripper Info**
- Select a **Gripper Name** from the dropdown menu
- Select a **Gripper Type** (finger or suction) from the dropdown menu.
- Specify the **Wait before** [ms] and **Wait after** [ms] times
- Specify an **Offset** [mm]

Wait before: if time is given, the robot will wait before picking an object and then start moving.

Wait after: if time is given, the robot will wait after placing the object and then start moving.

Offset[mm]: can be used to fine-tune the grasping of an item with the gripper by adding offsets to the previously taught position.



11.7.7Pick

The robot can pick objects from a designated region (workspace) using a gripper mounted to its flange. Various pick types are supported (see 11.7.7.1).

See 11.7.11 for troubleshooting of **Pick**, **PickAndPlace** and **BinPick** Apps.

1. Edit name and add description if wanted.
2. Edit items and save settings.

Item	Description
(1) Select Workspace	Drag and Drop from the list of pre-recorded workspaces. Workspaces can be recorded using the **Workspaces** wizard.
(2) Set Look at Point	Drop point or save point from where the robot can overlook the workspace and all items to be picked.
(3) Move to Look at Point	Move robot to Look at Point.
(4) Type of Pick	Select type of pick (see for 11.7.7.1 details). 1. Autonomous pick: for Random pick 2. Data based pick: for Pose aware pick 3. Pointing pick: for Interactive pick Activate Path planning if desired. This feature will smooth out the waiting time between each part of motion (and avoid collision during moving). Note that this will cause computational overhead.
(5) Object to Pick	Choose “selective” pick to pick only selected objects. Add objects to be selectively picked. The camera will capture an image from the current view and generate recommendations for objects to be picked. You may select from the image by clicking the desired object. Else any objects in view will be picked.
(6) Bin Workspace	For bin picking applications, select a pre-defined bin workspace. Workspaces can be recorded using the **Workspaces** wizard.
(7.1) Application Parameters	Select and define application parameters to fine tune the pick quality (see 11.7.7.2).
(7.2) Motion Parameters	Select and define motion parameters to fine tune the pick quality (see 11.7.7.3).

11.7.7.1 Pick types

Type	Autonomous pick	Data-based pick	Pointing pick
Precision	low	high	low
Object detection required?	no	yes	no
Pose estimation required?	no	yes	no
Interactive?	no	no	yes

Autonomous Pick

The autonomous pick allows picking an object without knowing its pose. Accordingly, neither pre-training nor pre-recording is required, instead, picking is performed on the fly based on the sensor input solely. Still, a user should specify gripper parameters, e.g. minimum/maximum opening of the gripper, to make the pick motion more reliable. Autonomous picks can generally be used for sorting, objects handover, cleaning, or other tasks that do not require precise motion.

Data-Based Pick

Data-based picks are based on the pose of an object. Respectively, pose estimation should be trained in advance. The sensor input is used to estimate the pose of the object. Pre-generated picks are transferred to the detected objects w.r.t the estimated poses, ranked, and the best pick is executed.

Data-based picks should be used in applications where precise picks are required, e.g., for fragile objects, objects being difficult to pick, or in cases where the picked objects should be placed precisely after the pick.

Pointing Pick

Pointing pick allows selecting an object according to a user's request. The object is specified by pointing with a right hand. The robot provides instructions by voice, making the interactive pick an easy-to-follow process. For the selected objects, picks are generated automatically.

NOTICE	Only relatively large objects could be picked with pointing pick, since small objects might not be recognized as selectable.

11.7.7.2 Application parameters

Picking is guided by a set of parameters. Default values are sufficient for most scenarios.

Parameter	Description
collision_pcd	Whether to check for collisions between end effector and point cloud captured from the sensor.
collision_workspace	Whether to check for collisions between end effector and workspace.
is suction gripper	Turn on if a suction gripper is used.
pre_grasp_distance	The distance from pre-grasp pose to grasp pose along the z axis (approach axis) of the end effector.
post_grasp_distance	The distance from post-grasp pose to grasp pose along the z axis of the robot (intuitively speaking toward up).
max grasp candidates per instance	The maximum number of grasps candidates that will be returned.
max pick attempts per capture	The maximum number of pick attempts per capture.
threshold_radius	The angle threshold for random pick. Only picks that have an angle to approach direction within the range of this threshold will be returned.

general_grasp_offset_x	The additional x offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
general_grasp_offset_y	The additional y offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
general_grasp_offset_z	The additional z offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
collision_space_padding	The padding of the collision space.
general place offset	The offset along the z-axis of the place pose in the robot base coordinate frame. The place pose is generated relative to the pick pose.
approach_direction	The desired approach direction w.r.t the workspace.
compare_ranking_method	Method used to rank the picks. Available options are:
joint_state	Rank candidates according to the joint-wise distance from pick candidate to the reference joint state. Default joint state is the current joint state (the moment when you select this option for the first time).
orientation	Rank the candidates according to the orientation of the approach direction. The closer the pick is to the desired approach direction, the better.
height	Rank the candidates according to the height of the pick along the approach direction. The higher the better.
quality	Rank the candidates according to the quality of the pick. The higher the better.

11.7.7.3 Motion parameters

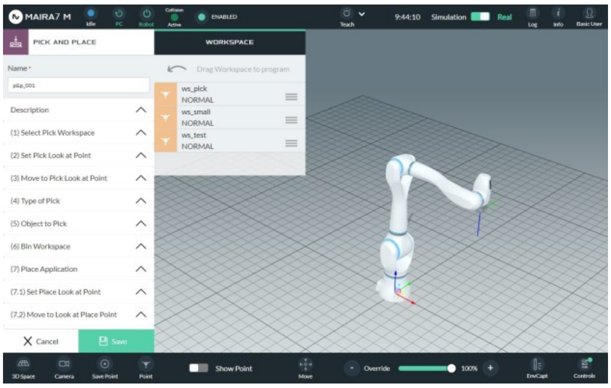
Parameter	Description
joint speed	The speed of Move Joint.
joint acceleration	The acceleration of Move Joint. Should match the joint speed, otherwise the robot may shake.
linear speed	The speed of Move Linear.
linear acceleration	The acceleration of Move Linear. Should match the joint speed, otherwise the robot may shake.

11.7.8 PickAndPlace

This function allows the robot to execute a pick and place task. For picking, refer to the documentation of the Pick App (see 11.7.7). For placing, three types are provided (see 11.7.8.1).

See 11.7.11 for troubleshooting of **Pick**, **PickAndPlace** and **BinPick** Apps.

1. Edit name and add description if wanted.
2. Edit items and save settings.



Item	Description
(1) Select Pick Workspace	Drag and Drop from the list of pre-recorded workspaces. Workspaces can be recorded using the **Workspaces** wizard.
(2) Set Pick Look at Point	Drop point or save point from where the robot can overlook the workspace and all items to be picked.
(3) Move to Pick Look at Point	Move robot to Pick Look at Point.
(4) Type of Pick	Select type of pick (also see 11.7.7.1). 1. Autonomous pick: for Random pick 2. Data based pick: for Pose aware pick 3. Pointing pick: for Interactive pick Activate Path planning if desired. This feature will smooth out the waiting time between each part of motion (and avoid collision during moving). Note that this will cause computational overhead.
(5) Object to Pick	Choose “selective” pick to pick only selected objects. Add objects to be selectively picked. The camera will capture an image from the current view and generate recommendations for objects to be picked. You may select from the image by clicking the desired object. Else any objects in view will be picked.
(6) Bin Workspace	For bin picking applications, select a pre-defined bin workspace. Workspaces can be recorded using the **Workspaces** wizard.

Item	Description
(7) Place Application	Drag and Drop from the list of pre-recorded workspaces. Workspaces can be recorded using the **Workspaces** wizard.
(7.1) Set Place Look at Point	Drop point or save point from where the robot can overlook the workspace and all items to be placed.
(7.2) Move to Look at Place Point	Move robot to Place Look at Point.
(7.3) Select Place Type	Select type of place (see 11.7.8.1 for details). 1. Pointing 2. Marker 3. Workspace
(8.1) Application Parameters	Select and define application parameters to fine tune the PickAndPlace app (see 11.7.8.2).
(8.2) Motion Parameters	Select and define motion parameters to fine tune the PickAndPlace app (see 11.7.8.3).

11.7.8.1 Place types

Pointing

Use your hand to point to a place within your workspace. The position the index finger is pointing to will be stored as place pose.

Marker

Use ArUco marker as placing position. Currently only a dictionary of specific markers is supported in the GUI interface. More can be added manually from the AI Hub.

Workspace

Use the center of the rasterized workspace as place position. Each grid will be visited iteratively. The column and row number can be assigned through the GUI.

11.7.8.2 Application parameters

Picking is guided by a set of parameters. Default values are sufficient for most scenarios.

Parameter	Description
collision_pcd	Whether to check for collisions between end effector and point cloud captured from the sensor.
collision_workspace	Whether to check for collisions between end effector and workspace.
is suction gripper	Turn on if a suction gripper is used.

general_grasp_detection_method	Not implemented yet.
pre_grasp_distance	The distance from pre-grasp pose to grasp pose along the z axis (approach axis) of the end effector.
post_grasp_distance	The distance from post-grasp pose to grasp pose along the z axis of the robot (intuitively speaking toward up).
max_grasp_candidates_per_instance	The maximum number of generated grasp candidates for each object instance.
max_pick_attempts_per_capture	The maximum number of pick attempts after one single capture of camera measurement.
threshold_radius	The angle threshold for random pick. Only picks that have an angle to approach direction within the range of this threshold will be returned.
general_grasp_offset_x	The additional x offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
general_grasp_offset_y	The additional y offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
general_grasp_offset_z	The additional z offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
collision_space_padding	The padding of the collision space.
general place offset	A vertical offset distance between position of placing object and surface.
approach_direction	The desired approach direction w.r.t the workspace.
compare_ranking_method	Method used to rank the picks. Available options are:
joint_state	Rank candidates according to the joint-wise distance from pick candidate to the reference joint state. Default joint state is the current joint state (the moment when you select this option for the first time).
orientation	Rank the candidates according to the orientation of the approach direction. The closer the pick is to the desired approach direction, the better.
height	Rank the candidates according to the height of the pick along the approach direction. The higher the better.
quality	Rank the candidates according to the quality of the pick. The higher the better.

11.7.8.3 Motion parameters

Parameter	Description
joint speed	The speed of Move Joint.
joint acceleration	The acceleration of Move Joint. Should match the joint speed, otherwise the robot may shake.
linear speed	The speed of Move Linear.
linear acceleration	The acceleration of Move Linear. Should match the joint speed, otherwise the robot may shake.

11.7.9 Scan App

The GPU powered rapid scanning capability allows the robot to digitize the scene around itself in a few seconds as it moves the integrated 3D camera around. The reconstructed 3D mesh can be used for purposes such as tool-path generation, obstacle avoidance, for 3D data collection for AR/VR use-cases etc.. If the texture mapping option is selected, the textured mesh can be visualized using the capture environment feature.

1. Edit name and add description if wanted.
2. Edit items and save settings.

Item	Description
(1) Select workspace	Drag and Drop from the list of pre-recorded workspaces. Workspaces can be recorded using the **Workspaces** wizard.
(2) Motion points	Motion points are points that the robot needs to move to scan the environment. The robot moves through the points with or without pause.
(3) Scanning type	Select scanning types to crop the output point cloud and mesh: **Environment**: the output of scan is point cloud and mesh without cropping. **Workspace**: the output point cloud and mesh are cropped within the given workspace. **Object**: the output point cloud and mesh are cropped within the given workspace with additional filters that are specific for single object scanning.

(4) Camera Pose Type	Select camera pose type with respect to the robot: Marker, Robot, SLAM.
(5) File Name	The name of the file, where the point cloud and mesh are stored.
(6) Parameters	

Parameter	Description
collision_pcd	Whether to check for collisions between end effector and point cloud captured from the sensor.
env mesh simplification factor	The percentage of triangles is simplified from original mesh that is applied for environment scanning. If you scan a large environment, please increase this value.
Obj Mesh Simplification Factor	The percentage of triangles is simplified from original mesh that is applied for object and workspace scanning.
Env scanning voxel size	The size of truncated signed distance function (tsdf) volume for environment scanning. This value depends on the GPU memory of the robot and how large the environment that is scanned. Increase the value if you scan large environment.
Obj scanning voxel size	The size of truncated signed distance function (tsdf) volume for object and workspace scanning.
No of Texture Mapping Views	The number of images used for mapping texture on the triangle mesh. Increase the value if you scan large scenes.

11.7.10 BinPick

In addition to the Pick application, bin pick provides better tools to handle tricky situations, for instance when all the objects are either clustered or stick at the corner, the robot will perform extra action trying to separate the objects.

If a list of object names is given, then only the selected object will be picked. The order of picking is decided by the algorithms. If a fixed order should be performed, multiple Bin Pick apps could be added in a series with different selected objects.

For each selected object, a drop point must be assigned. This point will be used to drop the object in the robot's hand.

Both a normal workspace and a bin workspace must be recorded before performing a bin pick.

See 11.7.11 for troubleshooting of **Pick**, **PickAndPlace** and **BinPick** Apps.

1. Edit name and add description if wanted.
2. Edit items and save settings.

Item	Description
(1) Select Workspace	Drag and Drop from the list of pre-recorded workspaces. Workspaces can be recorded using the **Workspaces** wizard.
(2) Bin Workspace	Drag and Drop from the list of pre-recorded bin workspaces. Workspaces can be recorded using the **Workspaces** wizard.
(3) Set Look at Point	Drop point or save point from where the robot can overlook the workspace and all items to be picked.
(4) Move to Look at Point	Move robot to Look at Point.
(6) Object to Pick	The object to be picked. Candidates for picking will be generated given the current camera view. The particular object can be selected by clicking.
(7.1) Application Parameters	Select and define application parameters to fine tune the pick quality (see 11.7.10.1).
(7.2) Motion Parameters	Select and define motion parameters to fine tune the pick quality (see 0).

11.7.10.1 Application parameters

Parameter	Description
collision_pcd	Whether to check for collisions between end effector and point cloud captured from the sensor.
collision_workspace	Whether to check for collisions between end effector and workspace.

pre_grasp_distance	The distance from pre-grasp pose to grasp pose along the z axis(approach axis) of the end effector.
post_grasp_distance	The distance from post-grasp pose to grasp pose along the z axis of the robot (intuitively speaking toward up).
min_aperture	The minimum distance between two fingers that the generated picks could ever have.
max_aperture	The maximum distance between two fingers that the generated picks could ever have.
default_opening	The distance between two fingers when the end effector is approaching the object.
max_number_of_grasps	The maximum number of grasps that will be returned.
approach_direction	The desired approach direction w.r.t the workspace.
threshold_radius	The angle threshold for random pick. Only picks that have an angle to approach direction within the range of this threshold will be returned.
general_grasp_offset_x	The additional x offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
general_grasp_offset_y	The additional y offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
general_grasp_offset_z	The additional z offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
collision_space_padding	The padding of the collision space.
compare_ranking_method	Method used to rank the picks. Available options are:
joint_state	Rank candidates according to the joint-wise distance from pick candidate to the reference joint state. Default joint state is the current joint state (the moment when you select this option for the first time).
orientation	Rank the candidates according to the orientation of the approach direction. The closer the pick is to the desired approach direction, the better.
height	Rank the candidates according to the height of the pick along the approach direction. The higher the better.
quality	Rank the candidates according to the quality of the pick. The higher the better.

11.7.10.2 Motion parameters

Parameter	Description
joint speed	The speed of Move Joint.
joint acceleration	The acceleration of Move Joint. Should match the joint speed, otherwise the robot may shake.
linear speed	The speed of Move Linear.
linear acceleration	The acceleration of Move Linear. Should match the joint speed, otherwise the robot may shake.

11.7.11 PickV2 Beta

PickV2 is an upgraded version of the Pick App, offering all the features of the original app along with the following new functionalities:

1. **Multiple Picks with Single Capture**: Execute multiple picks using just one camera measurement capture.
2. **Model Selection**: Choose between object detection and pose estimation models within the app settings.
3. **Bin Box Configuration**: Configure bin box detection and pose estimation either statically or dynamically via parameterization.
4. **Visualization**: View bin pose estimation results directly in the app settings.
5. **Foolproof Object Selection**: Select objects to be picked from a list tailored to the chosen object detection model, minimizing errors.
6. **Interactive Object Selection**: Use an interactive (finger pointing) method to select detected objects.
7. **Reference Points for Robot Motion**: Introduce start and end reference points to append robot motion before and after the pick.
8. **Pick Success Check**: Use gripper feedback to determine successful or failed picks.
9. **Result Evaluation**: Evaluate pick results based on criteria such as the number of successful picks and completion of picks, supporting IF/ELSE conditions.
10. **User Interface Improvements**: Enhanced user interface for a better overall experience.

Pick Types

There are three main pick types in the PickV2 App:

1. General Pick
2. Pose-Agnostic Pick
3. Pose-Aware Pick

General Pick

The general pick finds automatically by itself objects to be picked in the workspace and generates pick candidates (which to pick and how to pick) based on the sensor input solely. No prior knowledge or data of objects is required, neither pre-training nor pre-recording. This pick type can be used for objects handover, cleaning or other tasks that don't require a specific placement.

Pose-Agnostic Pick

The pose-agnostic pick employs pre-trained object detection module to distinguish different kinds of objects in the workspace and only pick the certain kinds of objects specified or selected by the user. No pose information of the objects is required by this pick type (therefore, pose-agnostic). This pick type is useful for sorting objects of different kinds in the workspace into different placements or bin boxes.

Pose-Aware Pick

The pose-aware pick executes pick motion with detected pose of the object to be picked (therefore, pose-aware). To do this, the robot firstly detects certain kinds of objects specified by the user using the object detection module. A pose estimation module is then called to detect the poses of the selected objects, based on which final picks are generated. Pick candidates of the object shall be pre-defined firstly in the object coordinate system by teaching or automatic generation. The pose estimation module can be either deep learning-based (pose-aware DL) or non-deep learning-based (pose-aware nonDL). This pick type can be used for precise pick, pick and place and tasks which require the robot to pick objects from a specific side or to place the object precisely after the pick.

Pick within Bin Box

The three pick types described above can be executed in either a standard workspace (e.g., a flat table) or within a bin box. When picking from a bin box, the pose of the bin box must be known to avoid collisions. In the PickV2 App, the bin box pose can be configured statically through the GUI or detected dynamically using the bin detection module.

Section	Description
Information	The name and description of the project.
Workspace	A pre-recorded workspace as region of interest.
Look at Points	Select multiple points as look-at points. These points are used to scan the workspace and objects.
Reference Points	Set reference points for start and end of the picking process.
Bin Workspace	Select a pre-recorded bin workspace.
Bin Pose	Enable or disable bin detection during picking.
Method	With bin detection method uses the bin model to localize the bin, without bin detection method it needs to have a bin workspace set up.
Bin Model	Load a bin model.
Bin Pose Estimation Period	Modify the frequency of the bin detection.
Cropping Factor	Set the percentage of the bin height, which is used for bin detection. Value can affect the speed of detection.
Pick Type	General, pose-agnostic, pose-aware (DL) or pose-aware (Non-DL).

<table>
<tr><td>Objects</td><td>The object to be picked.</td></tr>
<tr><td>Touch in Image</td><td>With bin detection method uses the bin model to localize the bin, without bin detection method it needs to have a bin workspace set up.</td></tr>
<tr><td>Select from a List</td><td>Load a bin model.</td></tr>
<tr><td>Point to Object</td><td>Modify the frequency of the bin detection.</td></tr>
<tr><td>Motion Planning</td><td>Set motion planning parameters.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>joint speed</td><td>The speed of Move Joint.</td></tr>
<tr><td>joint acceleration</td><td>The acceleration of Move Joint. Should match the joint speed, otherwise the robot may shake.</td></tr>
<tr><td>linear speed</td><td>The speed of Move Linear.</td></tr>
<tr><td>linear acceleration</td><td>The acceleration of Move Linear. Should match the joint speed, otherwise the robot may shake.</td></tr>
</table>
</td></tr>
<tr><td>Advanced Parameters</td><td>
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>collision_workspace</td><td>Whether to check for collisions between end effector and workspace.</td></tr>
<tr><td>is_suction_gripper</td><td>Turn on, if a suction gripper is used.</td></tr>
<tr><td>general_grasp_detection_method</td><td>Not implemented yet.</td></tr>
<tr><td>pre_grasp_distance</td><td>The distance from pre-grasp pose to grasp pose along the z axis(approach axis) of the end effector.</td></tr>
<tr><td>post_grasp_distance</td><td>The distance from post-grasp pose to grasp pose along the z axis of the robot (intuitively speaking toward up).</td></tr>
<tr><td>max_grasp_candidates_per_instance</td><td>The maximum number of generated grasp candidates for each object instance.</td></tr>
<tr><td>max_pick_attempts_per_capture</td><td>The maximum number of pick attempts after one single capture of camera measurement.</td></tr>
<tr><td>threshold_radius</td><td>The angle threshold for random pick. Only picks that have an angle to approach direction within the</td></tr>
</table>
</td></tr>
</table>

	range of this threshold will be returned.
general_grasp_offset_x	The additional x offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
general_grasp_offset_y	The additional y offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
general_grasp_offset_z	The additional z offset for pick in end effector coordinate. This will ignore the collision checking. Please use carefully.
collision_space_padding	The padding of the collision space.
general_place_offset	A vertical offset distance between position of placing object and surface.
default_opening	The distance between two fingers when the end effector is approaching the object.
approach_direction	The desired approach direction w.r.t the workspace.
compare_ranking_method	Method used to rank the picks. Available options are
joint_state	Rank candidates according to the joint-wise distance from pick candidate to the reference joint state. Default joint state is the current joint state (the moment when you select this option for the first time).
orientation	Rank the candidates according to the orientation of the approach direction. The closer the pick is to the desired approach direction, the better.
height	Rank the candidates according to the height of the pick along the approach direction. The higher the better.

	quality	Rank the candidates according to the quality of the pick. The higher the better.

11.7.12 PlaceV2 Beta

The place application offers functionalities to place grasped objects to specific positions within the workspace. Three place types are supported:

Point selection

Drag and drop pre-defined points from the list. Choose one point in "Single Point" mode or multiple points in "Multiple Points" mode.

Gesture

Show the robot a place point by pointing to it using the right hand. This place type is only available in "Single Points" modus.

Grid Based Points

Multiple placing points are defined by using a grid. They can be configured by several parameters (see below). This place type is only available in "Multiple Points" modus.

Section	Description
Information	The name and description of the project.
Workspace	A pre-recorded workspace as region of interest.
Look at Points	Select multiple points as look-at points. These points are used to scan the workspace and objects.
Bin Workspace	Select a pre-recorded bin workspace.
Bin Pose	Enable or disable bin detection during picking.
Method	With bin detection method uses object detection to identify the region of interest to localize the bin pose, whereas without bin detection uses the given workspace as region of interest to localize the bin pose.
Bin Model	Load a bin model.
Bin Pose Estimation Period	Modify the frequency of the bin detection.
Cropping Factor	Set the percentage of the bin height, which is used for bin detection. Value can affect the speed of detection.
Place Type	Point selection, gesture or grid-based points. Grid parameters: **Parameter** — **Description** **origin** — Drag& drop a saved point as origin of the grid.

	offset[x,y,z]	If bin localization is enabled, the offset in mm between bin origin and grid origin can be used as grid origin.
	cell count	The number of grid cells in x,y,z direction of the grid origin.
	cell size	The distance in x,y,z between two grid points.
	sequence	Available choices: [xy,xz,yx,yz,zx,zy], defines the axis order of looping through the grid place points.

Motion Planning

Set motion planning parameters.

Parameter	Description
joint speed	The speed of Move Joint.
joint acceleration	The acceleration of Move Joint. Should match the joint speed, otherwise the robot may shake.
linear speed	The speed of Move Linear.
linear acceleration	The acceleration of Move Linear. Should match the joint speed, otherwise the robot may shake.

Advanced Parameters

Parameter	Description
pre_place_distance	The distance in mm for linear motion to move to the place pose.
post_place_distance	The distance in mm for linear motion to move away from the place pose.
object_orientation	The orientation of the object at the desired place position.
pick_orientation	Use the same orientation for placing as the object was grasped for placing.
workspace_orientation	Align the place orientation with the place workspace.
place_offset_z	The offset in TCP direction in mm, positive values will put the place position higher, negative values lower.

11.7.13 Troubleshooting for Pick, PickAndPlace and BinPick Apps

	Questions and Answers
Q:	The pick is always too high (above the object) or just gently touched the surface of the object.
A:	Check that: 1. **max_aperture** and **default_opening** are not too small. Sometimes due to imprecise measurements, detected objects appear to be larger than real objects. 2. the workspace recorded is not too high for the object to be picked. Take a closer look at the point cloud visualized in GUI to ensure the object is fully shown: if not, reduce the height of the workspace. A general fix is adding a small value to **general_grasp_offset_z**.
Q:	The program quits right after I click **Play** button.
A:	Some components may be started incorrectly, e.g., the spawn time of control was longer than expected, or the camera was not started at first. Go to Supervisor page, which is the default page of the browser. Refresh the page and check if all components are running (especially motion planning and marker detection). If not, please restart them.
Q:	There is a warning/error that the instance segmentation is not working.
A:	The object detection model might be selected incorrectly. Quit the program, go to **AI** settings, and select a suitable model.
Q:	There is a warning/error that no object is detected.
A:	Check if the correct model for instance segmentation is selected. If so, please make sure the object is visible. If the problem persists, please retrain your object detection model with other lighting conditions and background.
Q:	The pick is not aligned with the object geometry.
A:	Check if the object is close to the workspace boundary. If so, move the object to the middle of the workspace, or enlarge the workspace. **NOTICE** If you change the workspace size, please make sure the updated workspace is fully visible from the current look-at point. Please also check if the point cloud is of sufficient quality. E.g., if the object to be grasped is dark or shiny, under some lighting conditions the camera may not work as expected. In this case, please adjust the lightning to ensure the point cloud quality is reasonably good.
Q:	There is always a warning about no collision-free pick.
A:	Sometimes, possible picks can be mistaken for collisions due to point cloud quality. In this case, you could adjust **default_opening** (+-10mm). Alternatively, if using General picks, you could disable **collision_pcd**.
Q:	Sometimes objects are picked successfully and sometimes not.

A:	For different objects, the best results can be achieved with different grasp parameters. If your fingertip is flexible, you may also try disabling **collision_workspace** and **collision_pcd**.
Q:	No valid grasp is generated even after long-lasting object detection.
A:	Please first check if too many irrelevant points are visible. If so, adjust your workspace so that fewer such points are inside the workspace. For small objects, please make sure that the objects are still visible.

12 WIZARD MENU

Wizards assist users in setting up and enhancing the AI functionalities on the robot.

► To access the available wizards, navigate to **HOME** >> **WIZARDS**

Entry	Meaning
Dynamic Parameter Identification	Re-identify the dynamical parameters (mass, center of mass, friction) of the entire robot. Can also be used to identify tool parameters.
Force Torque Sensor	The Force Torque Sensor (FTS) wizard lets you add an external FTS to your robot and helps with setup and calibration.
Robot Calibration	Calibrate the kinematic parameters of the robot after repairing an axis or to improve the absolute accuracy of the robot.
Workspaces	Define the operational area for the robot by recording a workspace. Crucial for AI applications.
Grasp Pose Recorder	Record and train preferred grasps for known objects. Objects must be trained for object detection and pose estimation in advance.

12.1 Dynamic Parameter Identification (DPI)

The Dynamic Parameter Identification is a real-world experiment which allows you to identify the dynamical parameters of the robot. This wizard can be used to improve the **ZeroG mode,** increase the robustness of the **Collision Detection** or to identify the **dynamic tool parameters**.

► The robot will perform two motions to collect data from its internal sensors and then recalibrate its dynamical parameters.

► Make sure to read the instructions in every step of the DPI wizard. Follow the detailed instructions.

► During the DPI wizard, **collision detection is deactivated**, and the operator is required to ensure that the workspace is free of obstacles and humans. **DPI is not a collaborative mode**, and a password is required to execute the wizard. The operator is required to ensure the safety in this scenario.

NOTICE	The DPI wizard considers the user settings regarding the gravity acceleration vector and can also be performed when the robot is mounted on a wall, the ceiling or any other inclined configuration.

WARNING

Make sure that you can trigger the Emergency-Stop-Button at any time! Always have a look at the robot that you want to calibrate. Avoid collisions of the robot with itself, the tool, humans and the environment.

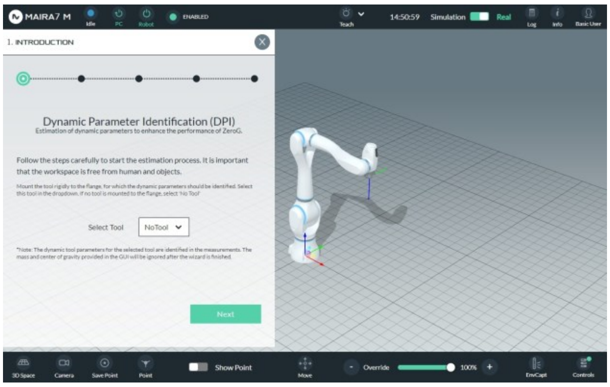
Start with low velocities and when you feel save and the robot is fine you can increase the value.

Procedure:

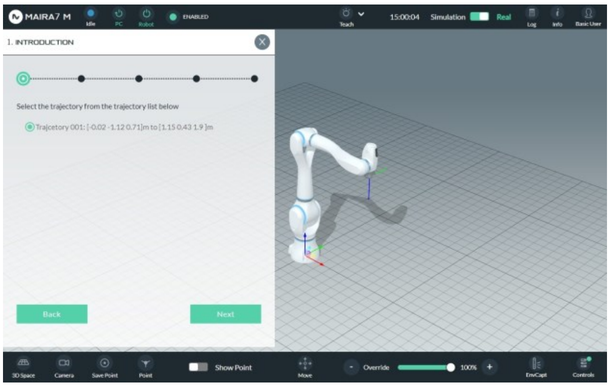
► When performing DPI with a tool mounted and selected, the tool parameters will be included into the flange of the robot. The dynamical tool parameters from the tool settings page will not be considered once a DPI with this tool is executed.

NOTICE	The trajectory also considers a **tool of 30cm x 30cm x 25cm to avoid self-collisions** with a possibly mounted tool.

► The 3D visualization in the GUI should now show the robot in approximately the same pose as the real robot.



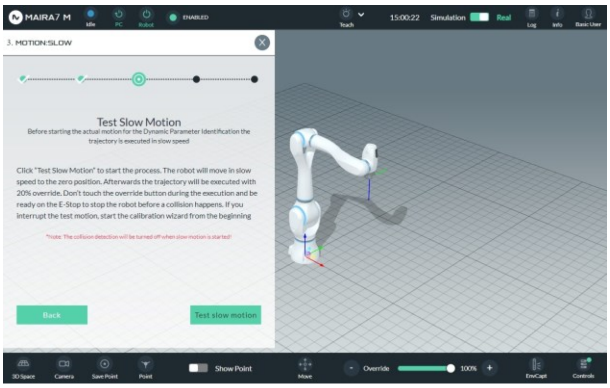
NOTICE	DPI should only be performed after warming up the robot through motion for at least 20 minutes (1 hour recommended). If DPI is performed without warming up the robot, the results might be impaired (ZeroG/collision detection might be negatively affected).



► Start the DPI Wizard by entering the password

► Select the mounted tool from the dropdown and tap **next.**

► A list of trajectories will be shown. The DPI also considers Global Settings regarding the joint range limits. Running a DPI trajectory is not possible when the limits are too restrictive. (Warning will be shown.) Choose a valid trajectory and tab **next.**

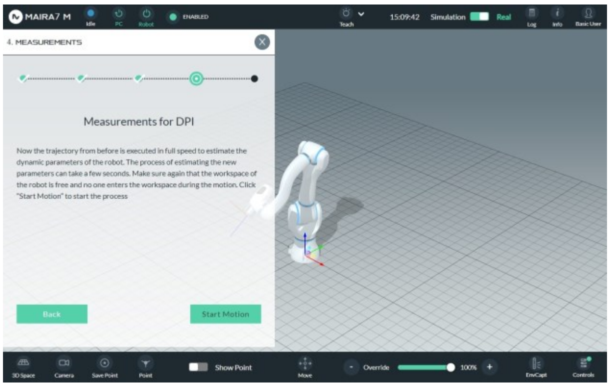


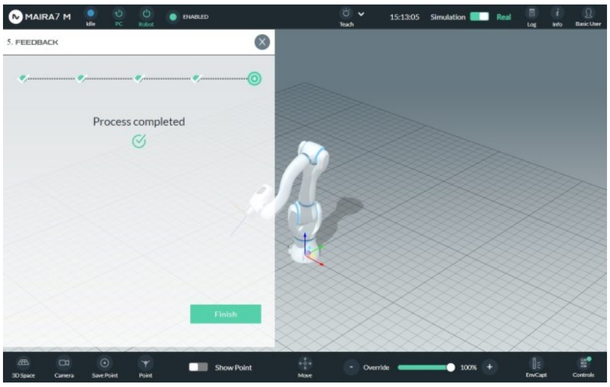
► To start the motion of the robot, Automatic Mode is needed (switched automatically).

► The wizard consists of **three main steps**:

- One slow test run of the trajectory (10% speed) to make sure the robot does not collide with its surroundings. **During the test run, be prepared to stop the robot at any time.**

- One full speed run to collect the sensor data. The same trajectory will now be run at full speed. The override cannot be changed manually.
- Evaluation of the data & reloading of the parameters.





► The robot will perform a **very fast motion** in the **positive x- and z-planes of its base frame (in front of the robot)**. Make sure that the robot has sufficient space to perform the identification trajectory.

► DPI also considers Global Settings regarding the joint range limits. Running DPI is not possible when the limits are too restrictive. (Warning will be shown.)

► After the execution of the wizard, the new parameters are loaded automatically, and collision detection is turned on again.

► Dynamic Parameter Identification can also be used to identify unknown tool dynamic parameters:

- After running DPI, the previous parameters are overwritten and lost. Currently, we recommend backing up the parameters via **NeuraUSB**, and reinserting them if desired.
- When unmounting the tool, the operator needs to select NoTool to adapt the dynamic parameters.

► Performing DPI is recommended in the following situations:

- Unavailability or difficulty to find the correct tool parameters.
- Unsatisfying robustness of the collision detection.
- Improving hand-guiding smoothness for ZeroG Mode.
- Change of environmental temperature.

NOTICE	After performing DPI, perform a simple validation check of the ZeroG mode, and eventually adjust the friction and holding torque sliders.

12.2 Robot Calibration

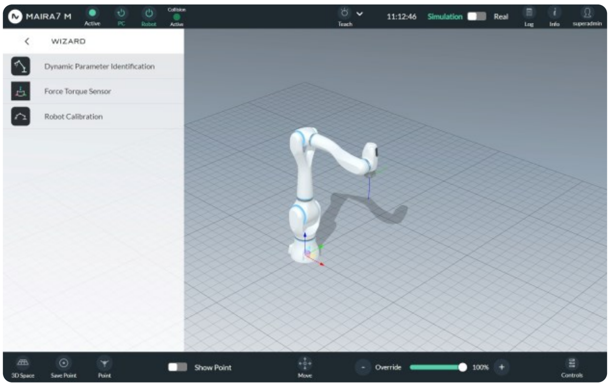
In the process of **closed loop calibration**, the kinematic parameters (length, angles, ...) of the robot are estimated to achieve a **high absolute accuracy.** This might be necessary after changing an axis or to improve the absolute accuracy of the Cobot. Note that the tool parameters are not identified in this procedure. To perform the kinematic calibration, a special tool is required. Please contact your integrator for such a tool.

When to perform the Kinematic Calibration?

- After replacing an axis
- When the absolute accuracy of the robot is not as desired

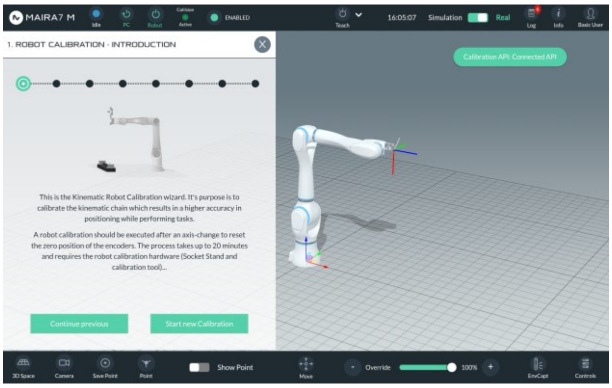
► **General Process of Closed Loop Calibration**

- Hardware Setup: the robot and the required tool & sockets are mounted.
- Using the Wizard in the GUI, the experimental data collection is performed.
- The user will move the ball of the mounted tool into 2 different sockets in **ZeroG mode** several times.
- During the experiment, the required data is collected.
- After the completion of the data, the new kinematic parameters are computed, and the results are visualized.
- In the end the user can choose to apply the results of the Closed Loop Calibration.



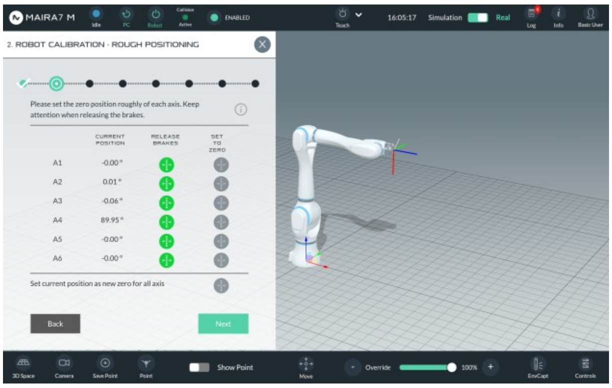
1. **What is required to perform the Kinematic Calibration?**

   - You will need a NEURA Closed Loop Calibration tool & the corresponding sockets.
   - A setup in which robot base and sockets do not move relative to each other. Ideally, robot and socket are mounted on the same table or on bases that are fixed to the ground.
   - The calibration process will take approximately 15 minutes.
   - Either continue the previous calibration or start a new calibration. A calibration can only be continued if it has not been applied yet. Another requirement to continue a previous data recording is that the reference position of the socket device did not change.



2. **Rough Positioning**

   - If performing the Closed Loop Calibration Wizard after repairing an axis, make sure to adjust the zero position of the replaced axes to an approximately correct position. Page 2 of the Closed Loop Wizard will provide you with all the functionality for this. For information on how to do this, please refer to the **Break Release** section below.
   - This slide is only available if a new calibration is started.



Break Release:

This feature enables the operator to open the brakes of individual axes. This is required after changing an axis or to recover from a collision. The robot axis must be turned off before opening the brakes.

- ► Pressing the red button for an axis will open the brake of that axis and allows the operator to move the axis.
- ► Opening the brake is only possible when the robot is turned off.
- ► Using the **Set to Zero** button will set the current joint configuration as the new Zero position of the

axis. This is only recommended after changing an axis.

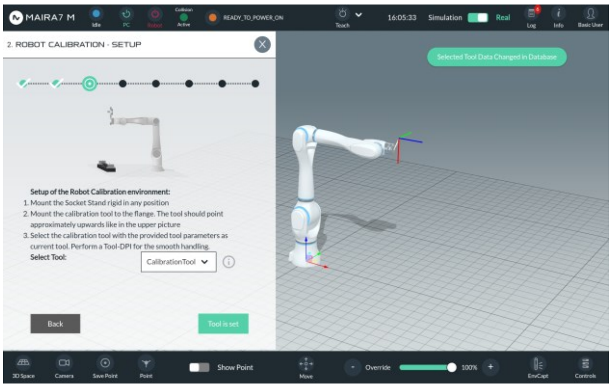
The brake features should only be used by trained personnel. The consequences of changing the Zero position of an axis are the responsibility of the operator.

Note: changing the Zero position will affect the positioning of the robot. Afterwards, the robot needs to be recalibrated to achieve high absolute accuracy.

⚠ WARNING
Manually setting the zero position of the axis overwrites the calibrated zero positions. Setting wrong zero positions can damage the system and lead to unexpected behavior of the robot system.

3. Setting up the Closed Loop Calibration

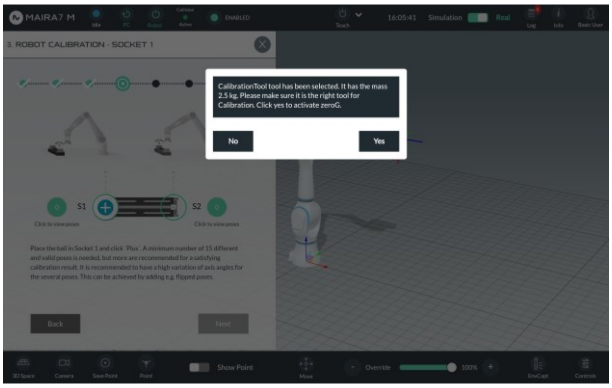
- The robot and the socket need to be fixed in a setup such that robot base and sockets do not move relative to each other. Ideally, robot and socket are mounted on the same table or on bases that are fixed to the ground.
- The screws of the robot base and the socket need to be fixated properly so that during the process, the **socket does not move relative to the base of the robot**.
- If the socket moves relative to the robot base during the experiment, the results will be poor, and the closed loop calibration needs to be repeated.
- Open the loose socket with an Allen key to release tension of the socket stand. Then fix this socket again before starting the data recording.
- Additionally, the Closed Loop Calibration tool needs to be properly mounted to the robot's flange, with properly fixated screws.
- It is recommended to mount the tool when the robot is in a joint configuration of [0°, 0°, 90°, 0°, 0°, 0°]. Then mount the tool that the ball points approximately upwards. In this configuration the default installed "CalibrationTool" with following parameters is valid:
  - Mass: 2.534kg



- o Center of Mass X: -42.13mm
- o Center of Mass Y: -17.45mm
- o Center of Mass Z: 13.5mm

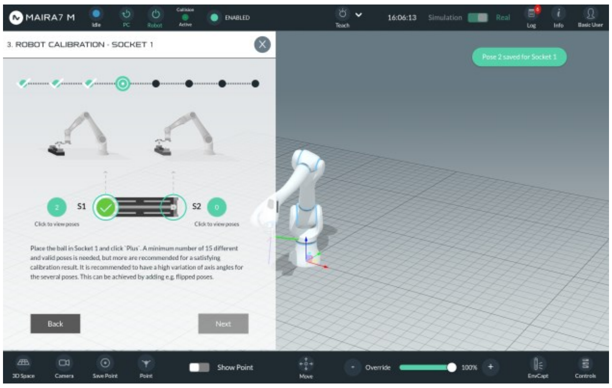
o Verify that the tool parameters are matching to the mounting position of the tool by turning on **ZeroG mode.** It is the responsibility of the operator to insert the correct tool parameters. Incorrectly inserted parameters can lead to undesired behavior.

4. Calibration Socket 1: Data Collection



o To perform the data collection, the **ZeroG mode** will be turned on.

o Then move the ball of the Closed Loop Calibration Tool into **Socket 1.** The ball needs to be placed precisely into the socket and there should not be any tension in the system. It does not matter which socket is chosen as **Socket 1,** but the sockets should not be switched during the whole process.

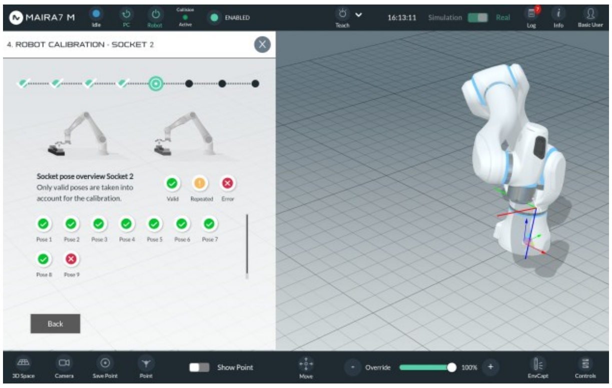


o Press on the **Plus-Icon** to record the measurement. During the measurement, make sure the robot is not moved. A measurement takes approx. 2 seconds. The quality of the measured pose is visualized through either a green tick (valid), a yellow exclamation mark (repeated pose) or a red cross (outlier / movement during measurement).

o Change the joint configuration of the robot and guide the ball back into **Socket 1**. Press the **Plus-Icon** again to record the pose.

o It is important that the joint angles are different from the previously saved pose. It is recommended to take the robot out of the socket to guide it into a different configuration. Make sure to vary all axis as much as possible. It is recommended to record **normal poses** and **flipped poses.** In flipped poses, axis 1 is moved close to the limits and axis 3 is rotated in vice versa direction compared to normal poses. The more variance in the distribution of the poses, the better the results of the Closed Loop Calibration.

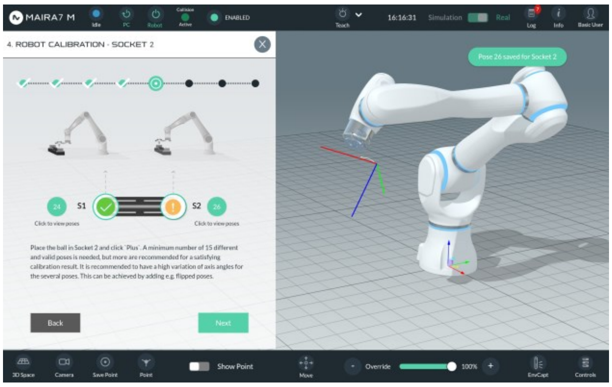
- **Repeat the process** of taking the robot out of Socket 1, changing its joint configuration, placing it back into Socket 1 and saving the data. **Only record the measurements inside Socket 1.**
- In this matter record a **minimum of 15 valid poses for each Socket**. 30 poses per Socket are recommended.
- **A pose overview** can be seen when clicking on the button with the number of the taken poses. A valid pose is visualized with a green tick. A repeated pose is visualized with a yellow exclamation mark and an outlier is visualized with a red cross. Only valid poses are considered into the calculation. Each pose can be even manually deleted by clicking on it and typing on the trash-bin button.



- Once sufficiently valid poses are recorded, you can click **Next** to start the recording for Socket 2.

5. **Calibration Socket 2: Data Collection**

- Record poses for **Socket 2.**
- It is important that until now, only measurements in Socket 1 were recorded. If poses were recorded wrongly, they can be deleted manually in the **pose overview for each socket**.

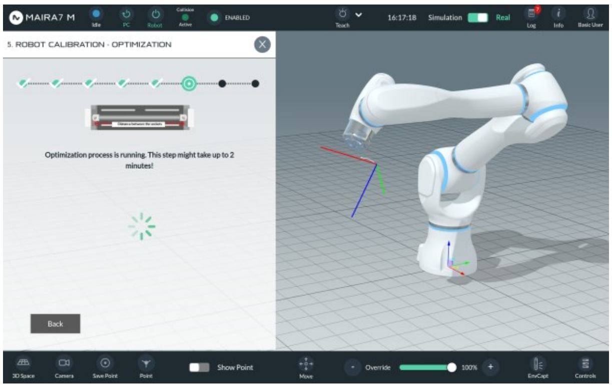


- To record the data: Place the ball of the Closed Loop Calibration Tool into Socket 2 and press on **Plus** to record the data for this pose. Then change the joint configuration, insert the tool into Socket 2 again, and save the next pose.
- Continue this process until a minimum of 15 valid poses are recorded.
- Once enough poses are recorded the green check mark will pop up. Press **Next** to complete the data collection process.

6. **Optimization**

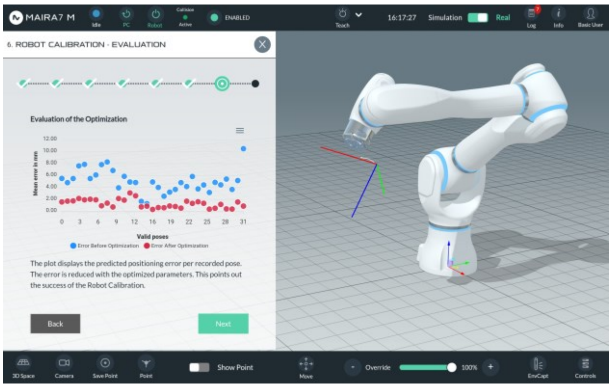
- In the optimization step, the operator needs to insert the distance between the two sockets. The distance is determined by NEURA's in-house high-precision measurement devices and is printed onto the Closed Loop Calibration Sockets.

- Insert the value with the highest precision given in millimeters, and double check the value. A wrongly inserted value will negatively influence the results.
- Press on **Next** to start the optimization process. This will take up to 2 minutes.



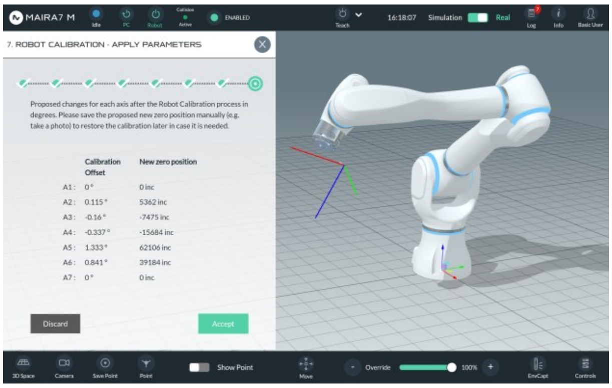
7. Evaluation

- In this step the evaluated data is presented.
- The plot will show the absolute accuracy of all the recorded points before and after the calibration (so with the old and the new kinematic parameters)
- In **blue**, one can see the **absolute accuracy before the calibration**: in **red** the absolute accuracy **after** the calibration.
- Note that overall, the red dots should be below the blue dots, otherwise something went wrong in the process.
- Additionally, if there are some blue dots that are far away from all the other points, that means that the user recorded points that were not placed properly in the socket and the Closed Loop Calibration process should be repeated.

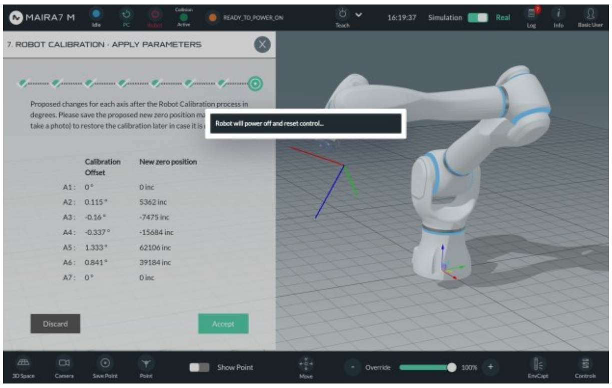


8. Apply Parameters

- In this last step, the operator can decide whether to apply the newly identified parameters or not to.
- The decision should be made based on the plot from the previous step (Is the absolute accuracy after the calibration better than before?) and based on the suggested changes of the zero position of the joints.
- Visualized are also the suggested changes to the zero position of each joint. These changes should be relatively small, except on the axis that was repaired or visibly far off.



- Click on **Accept** to set the new parameters. Consecutively, the parameters will be loaded and the robot restarts.
- It is suggested to note down the new absolute encoder offsets, which are shown on this page as well.



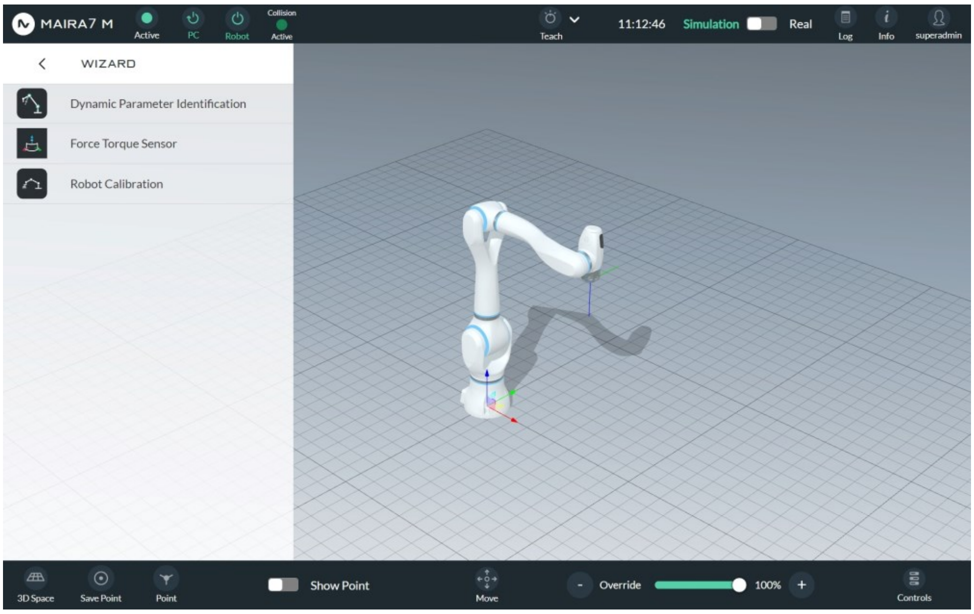
12.3 Force Torque Sensor

The Force Torque Sensor (FTS) wizard lets you add an external FTS to your robot and helps with setup and calibration.

🗐 Your FTS is mounted to the robot's mechanical tool interface (tool center point (TCP) flange).

► To start the force torque sensor wizard, navigate to **Home > Wizard**.

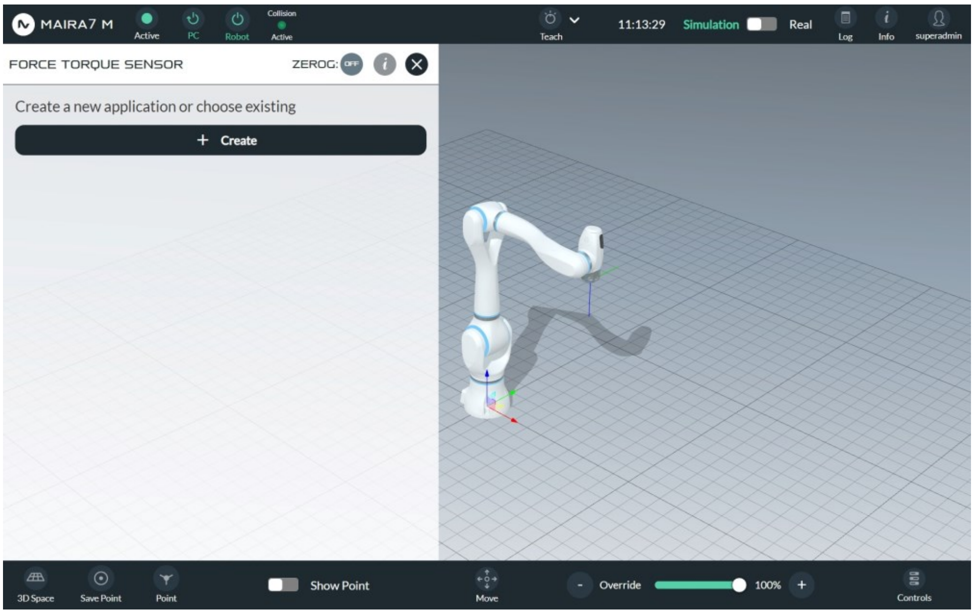
► Tap Force Torque Sensor.



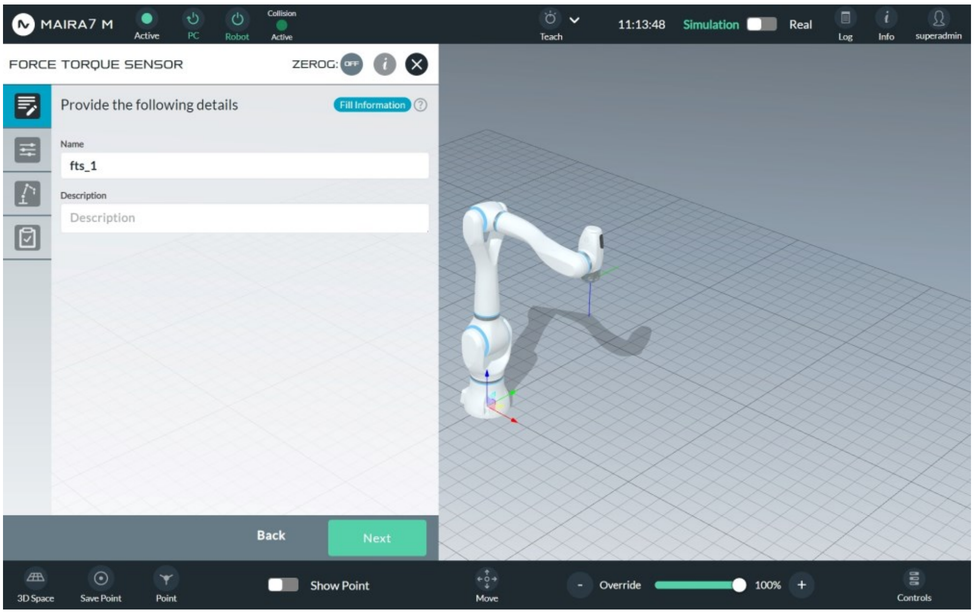
► Tap **+ Create** to create a new FTS.

► Or tap an existing FTS to select it, then tap the **Edit** icon to edit its configuration.

► Set the parameters listed and explained below to configure your FTS according to its specification.



Parameter	Explanation
Name	▶ Give the FTS a name or use the auto-generated one.
Description	▶ Give the FTS an optional description.



▶ Tap **Next**

Parameter	Explanation
Sensor Model	▶ Select a FTS from the list of supported devices. Currently only the Robotous_RFT80-6A02-D FTS is supported.

▶ Open the robot's control box and connect the FTS's USB cable to any of the available control box USB ports. Follow the instructions in the robot's user manual.

Parameter	Explanation
TCP Offset	Define the FTS offset relative to the robot's TCP flange. ▶ Tap the **Question mark** icon in the upper right corner of the FTS wizard to see the tool tip regarding offset definition. ▶ Specify the T2 – Sensor TCP offset in x-, y-, z-direction [mm] and/or rotation about those axes A, B, C [deg].
COM	Define the FTS's weight and center of mass (COM). ▶ Tap the **Question mark** icon in the upper right corner of the FTS wizard to see the tool tip regarding COM definition. ▶ Specify the weight of the FTS [kg]. ▶ Specify the M – Center of mass (COM) in x-, y-, z-direction [mm].

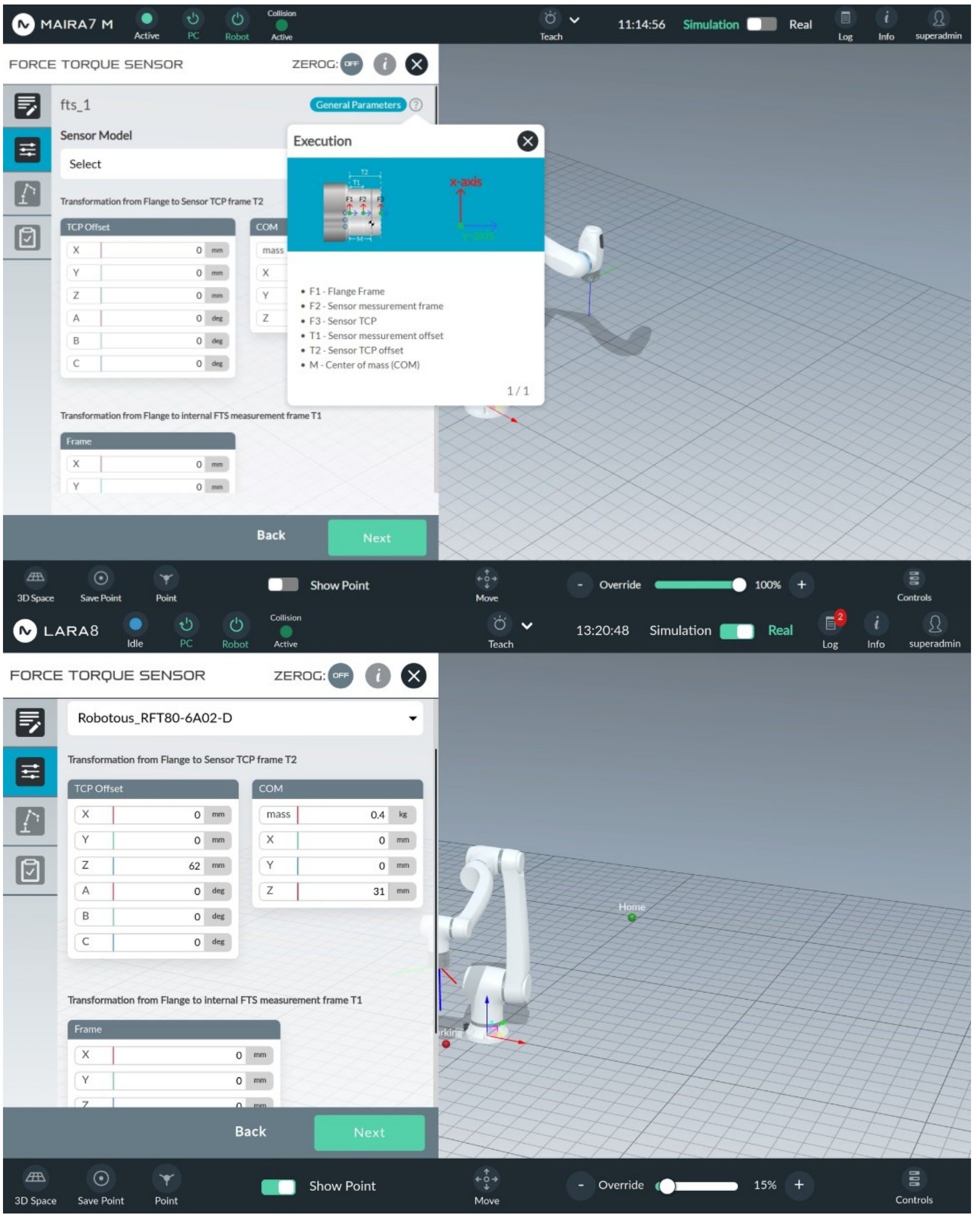
Frame

Define the FTS measurement offset relative to the robot's TCP flange.

► Tap the **Question mark** icon in the upper right corner of the FTS wizard to see the tool tip regarding offset definition.

► Specify the T1 – Sensor measurement offset in x-, y-, z-direction [mm] and/or rotation about those axes A, B, C [deg].

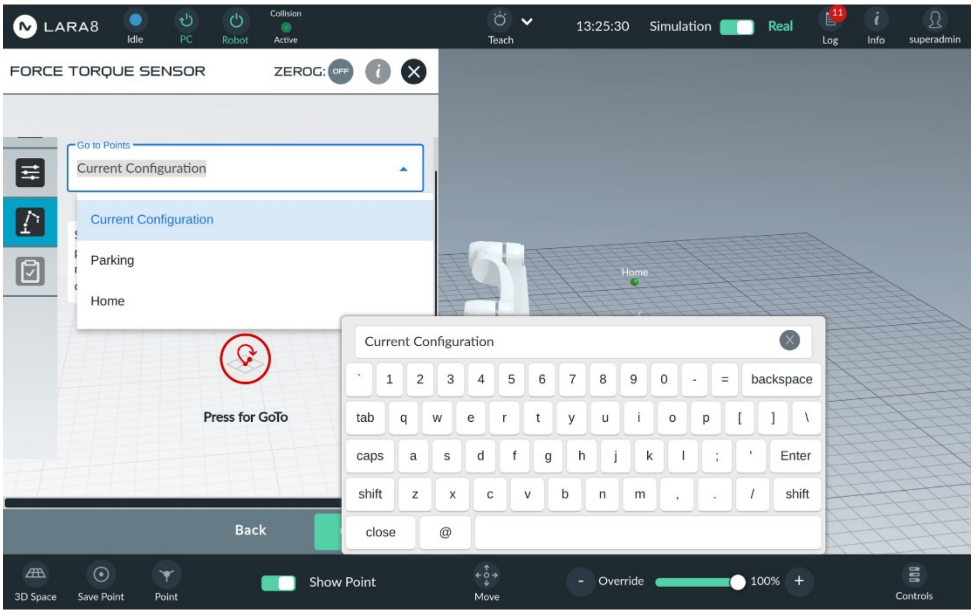
Information about the sensor's measurement reference frame can usually be found in the manufacturer's user manual.



Screenshot shows the LARA robot, but illustration is also valid for the MAiRA robot.

► Tap **Next**

Parameter	Explanation
Go to Points	Select a starting point for the FTS calibration. For calibration, the robot's axis 5 and axis 6 or axis 6 and axis 7 (the last 2 axes depending on your type of robot) will move in 90 deg steps and collect FTS data for taring. To move the robot into a start position for calibration: ► Select a point from the **Go to Points** dropdown menu. ► Tap **Press for GoTo** to move the robot to the selected point. ➔ The robot is in a start position for calibration. Alternatively: ► Use ZeroG to move the robot into a start position by hand. ► Select **Current Configuration** from the **Go to Points** dropdown menu. ➔ The robot is in a start position for calibration.



Screenshot shows the LARA robot, but illustration also valid for the MAiRA robot.

► Tap **Calibrate** to start the FTS calibration

The robot's axis 5 and axis 6 or axis 6 and axis 7 (the last 2 axes depending on your type of robot) will move in 90 deg steps and collect FTS data for taring.

► Observe the calibration movement and ensure that the sensor assembly does not collide with the robot body.

► After successful calibration, FTS biases and live readings are displayed in the FTS wizard.

► Tap **Save** to save your FTS configuration.

► Navigate to **Home > Settings > General**.

► Select your FTS from the **Select FTS** drop-down menu

➔ The FTS is now configured and ready to be used with your robot.

12.4 Workspace Wizard

The Workspace wizard allows users to define the operational area for the robot by recording a workspace. Since most AI functionalities and applications depend on a predefined workspace, it is crucial to complete the following steps before using these applications.

To create a new workspace:

- 🗐 The robot is powered on / ENABLED
- 🗐 The wrist camera is activated
- 🗐 The workspace is clearly visible by the wrist camera
- 🗐 The workspace is clean and collision-free

► Start the **Workspaces** wizard via the main menu **HOME >> Wizard >> Workspaces**

► Press the **+ Create** button to add a new workspace

► Enter a unique workspace **Name**

► Record a Look-at Point

- A look-at point is the point, where the robot is moving to before scanning the workspace. Select a look-at point from a list of pre-recorded points. Choose a point so that the camera can capture the complete workspace.

- Select a workspace type
  - **NormalWorkspace**
  - **BinWorkspace**
    - For **BinWorkspace** select a **BinMeshModel** if available or one will be created as part of the workspace

- Select a Teach Mode via the **Method** dropdown menu
  - Teach Using Robot
  - Teach Using Gesture

- Record Limit Points
  - A Workspace is located inside a 3D volume grounded on the supporting surface. Respectively, it can be defined with 1) a parallelogram area within the supporting surface and 2) a height upon the supporting surface.
  - First, one needs to specify the area of interest on the supporting surface. To this end, three limit points should be provided: Left, Right, and Rear.
  - For each limit point, set its position manually if using robot, or by pointing if using gestures. For gestures, select **Left/Right/Rear** limit, and click the **Teach** icon.

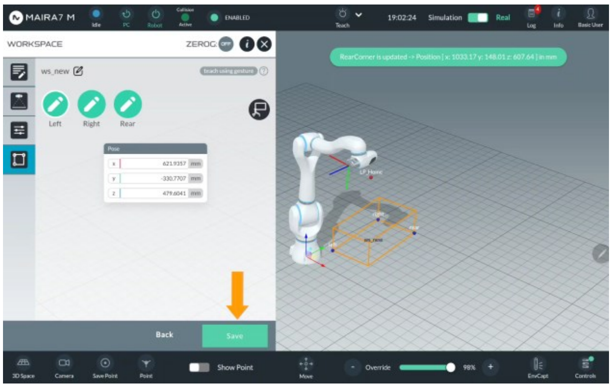
- ► Show your right hand to the robot and wait until it detects it
  - o Make sure to show your hand (all fingers) first and then point to a specific point. When pointing, place your hand on the table. Do not point from above otherwise the recorded point position is not accurate.

- ► Point to the left limit using the second finger (forefinger)

- ► Wait until the limit is recorded and a popup notification appears

- ► Repeat the same actions for the Right and Rear limits

- ► Press the **Save** button
- ➔ Your new workspace has been configured and saved
- ► Existing workspaces can be selected for further actions
    - o Edit
    - o View/hide
    - o Delete
    - o Duplicate



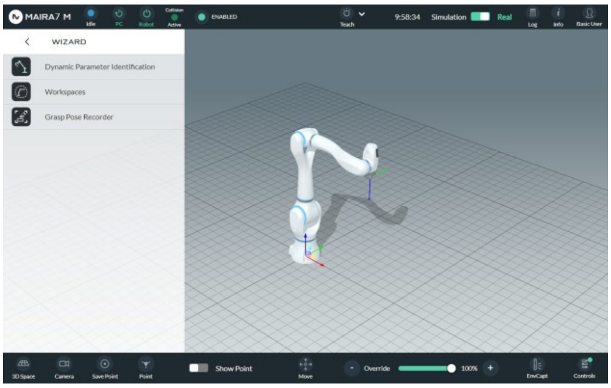
12.5 Grasp Pose Recorder Wizard

The Grasp Pose Recorder enables users to record and train preferred grasps for known objects. Please note that objects must be trained for object detection and pose estimation.

This function closely integrates with any pose-aware pick and place application on the robot. The grasps generated in the manipulation app and data-based pick app are based on the recorded grasps using this wizard.

To record a new grasp:

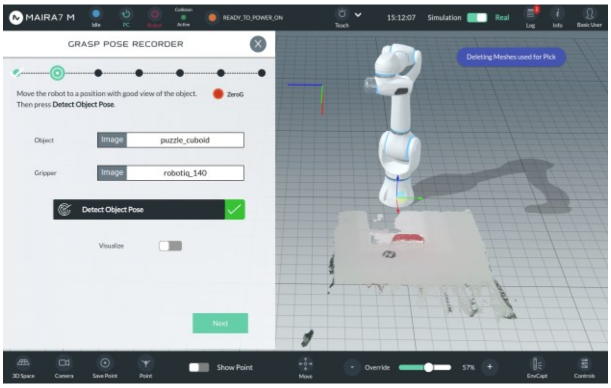
- The robot is powered on / ENABLED
- ZeroG is available
- Upload your object folder to **/data/object_perception/objects/user_objects** (only needed if object not already loaded)
- Set up proper object detection and pose estimation models
- The workspace is clearly visible by the wrist camera
- The workspace is clean and collision-free



NOTICE	The object should not be moved during the whole recording process, since the recorded data gets converted into the object's frame.

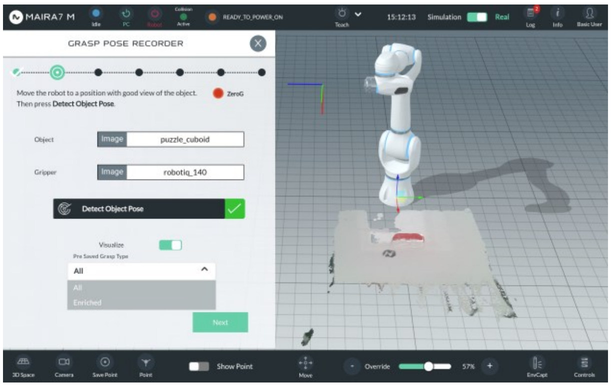
- ► Navigate to the **Grasp Pose Recorder** wizard via the main menu **HOME >> Wizard >> Grasp Pose Recorder**

- ► Press the **Start** button
- ► Choose an object and a gripper from the list menu
- ► Press **Detect object pose**



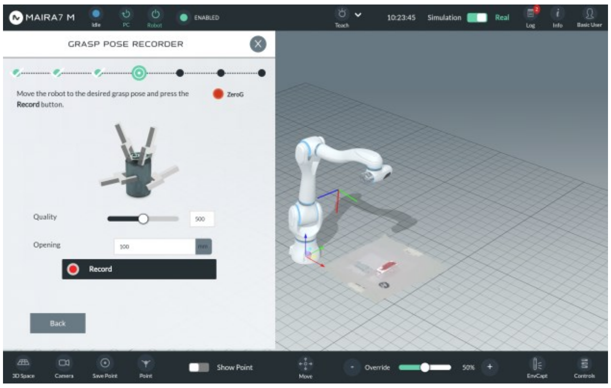
NOTICE	When object detection and pose estimation are successfully completed, you will hear a beep and see the object in the GUI. The first time scanning an object might take some time, as soon as the object is scanned once, the process will go faster.

- ► To visualize recorded graphs, activate **Visualize** via the slider
- ► Two types of poses can be selected
  - o All: all pre-saved grasp poses are shown (parent and child grasp poses)
  - o Enriched: only child grasp poses are shown

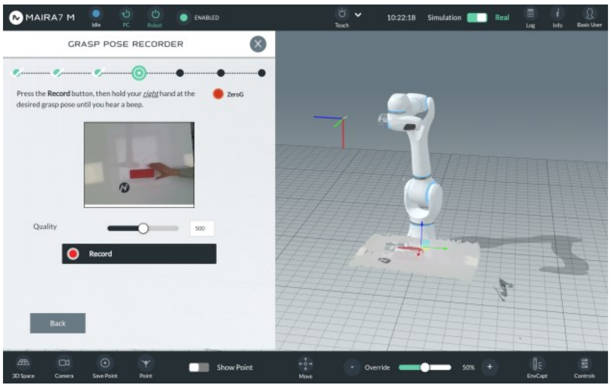


NOTICE	An original recorded grasp pose is saved as a “parent” grasp pose. Based on parent grasp poses, there are automatically created multiple “child” grasp poses, that enrich the set of possible grasp poses.

- ► To record grasp candidates, select **Robot** or **Hand** via the radio buttons
  - o Robot: the robot can be moved to a specific position, and this position is then saved as the grasp pose for the object. Further you can define the quality of the grasp and the opening of the gripper.
  - o In Hand mode, you are asked to show the grasp manually with your hand. Further you can define the quality of the grasp.



Successful grasp recording will be visually and acoustically indicated.



CAUTION

Be careful, as the robot might move to get a better viewpoint for capturing the hand.

NOTICE	Set the quality of the grasp, defined by a number between 0 (least preferred grasp) and 1000 (most preferred grasp).

- ▶ **Discard** or **Confirm** the taught grasp via the corresponding buttons
- ▶ Press **Record Next** to teach additional grasps or press finish to continue with taught grasps

- ▶ To correct recorded grasps by post-processing, press the **Configure** button (parameters are explained in 12.5.1 below)
  - o Not necessary for teaching in robot mode, because of higher inherent precision
  - o The correction is performed automatically. For each recorded grasp, multiple candidates are generated within the range specified by the parameters.
  - o The candidates are filtered to guarantee that they are:
    - ▪ Not too close to object boundaries
    - ▪ Perpendicular to the normal in the contact point
    - ▪ Cover large surfaces
  - o Two best antipodal grasps meeting these criteria will be saved
  - o For symmetrical objects, several grasps will be added considering the symmetry.
  - o For a large number of grasps, correction may be time consuming
- ▶ If there are previously recorded grasps for this specific object, a user can add the newly recorded ones to the existing set (**Merge**) or overwrite all previous generated grasps (**Overwrite**).

- To visualize recorded graphs, activate **Visualize** via the slider and select pose categories to be visualized via the dropdown menu
- Three types of poses can be selected
    - All: all pre-saved grasp poses are shown (parent and child grasp poses)
    - Enriched: only child grasp poses are shown

- To save the generated grasp poses and finish the wizard press **Done**

➔ Your object specific grasps have been recorded

12.5.1 Correction Parameters

Parameters	Description
x_angle_min	The rotation interval of enriched candidates [-3.14 - 3.14]
x_angle_max	The rotation interval of enriched candidates [-3.14 - 3.14]
y_angle_min	The rotation interval of enriched candidates [-3.14 - 3.14]
y_angle_max	The rotation interval of enriched candidates [-3.14 - 3.14]
z_angle_min	The rotation interval of enriched candidates [-3.14 - 3.14]
z_angle_max	The rotation interval of enriched candidates [-3.14 - 3.14]
x_displacement_min	The translation interval of enriched candidates [-1.0 - 1.0]
x_displacement_max	The translation interval of enriched candidates [-1.0 - 1.0]
y_displacement_min	The translation interval of enriched candidates [-1.0 - 1.0]
y_displacement_max	The translation interval of enriched candidates [-1.0 - 1.0]
z_displacement_min	The translation interval of enriched candidates [-1.0 - 1.0]
z_displacement_max	The translation interval of enriched candidates [-1.0 - 1.0]
x_rotate_times	The times of rotation [0 - 1000]
y_rotate_times	The times of rotation [0 - 1000]
z_rotate_times	The times of rotation [0 - 1000]
x_translate_times	The times of translation [0 - 1000]

y_translate_times	The times of translation [0 - 1000]
z_translate_times	The times of translation [0 - 1000]
sample_points	The number of samples from mesh to pointcloud [0 - 500000]
leaf_size	The downsampling leaf size [0.001 - 0.1]
min_percentage	The min percentage of points that should be close enough to the finger tips [0.0001 - 1.0]
middle_difference	The min allowed difference in percentage of distance between object to left and right finger [0.0001 - 1.0]
touch_threshold	The points of object that have less distance than touch_threshold will be treated as touched points [0.0001 - 1.0]
friction_angle	The friction angle for stabiliy check [0.0 - 1.57]
mesh_scale	The scale factor of the current object mesh, e.g. 0.001 for mm [0.001 - 1000]

12.5.2 Troubleshooting

Questions and Answers	
Q:	After pressing **Detect object pose**, application waits for long time
A:	It may happen if you are trying to use object detection or pose estimation model for the first time during the current session, as it takes time to set the model up. Please wait.
Q:	**Detect object pose** failed
A:	Please check if you loaded the correct object detection and pose estimation model. Please also check if the object is fully visible.
Q:	There is no object visualized in the GUI
A:	Please check if **<your object name>_downsampled.obj** file exists.
Q:	My hand pose was not recorded correctly.
A:	Please make sure you are using your pointing finger and thumb on your right hand. Please make sure that these fingers are fully visible during the recording.
Q:	My recorded pose is wrong in Robot mode.
A:	Please make sure you are using the correct gripper and the offset of the TCP of the gripper is set correctly. Please make sure that ZeroG is turned off.
Q:	I want to record grasps on the opposite side of the object.

A:	You can save the current grasp. Then, you can select the Merge option to add new grasps, turn the object, and restart the recording process.
Q:	My enrichment failed.
A:	It may happen if the original recorded grasps are not close enough to a perfect grasping pose. Or the object shape is not suitable under the current parameter settings. Try setting larger values for **sample_points**, **touch_threshold**, and **leaf_size** parameters, if recording grasps for small objects. Try increasing translation/rotation times for a better result (note that it will increase the computation time respectively).

13 MAINTENANCE MENU

Not assigned / future release.

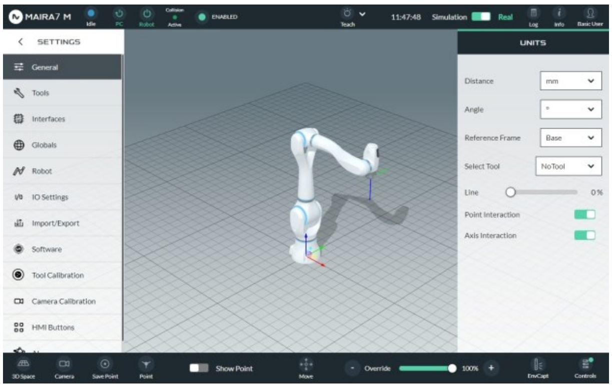
14 DASHBOARD MENU

Not assigned / future release.

15 SETTINGS MENU

In the **Settings** menu you can change all the system and robot settings.

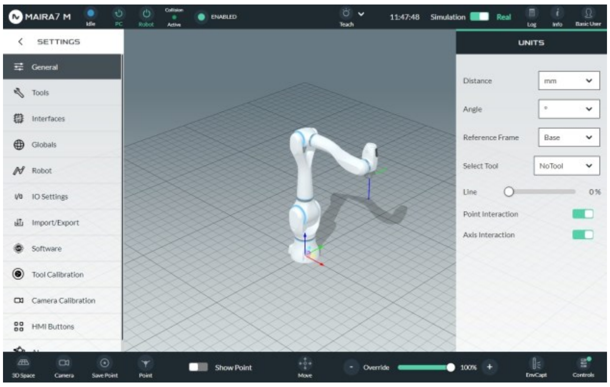
► Click the **Settings** button to open the menu.



Item	Meaning
General	Change general settings.
Tools	Add and edit tools.
Interfaces	Change interface settings.
Globals	Change global settings.
Robot	Change robot specific settings.
IO Settings	Change input and output settings.
Import/Export	Import/Export data from/to a USB flash drive.
Software	Install new software.
Tool Calibration	Add and edit tool calibration settings.
Camera Calibration	Add and edit camera calibration settings.
HMI Buttons	Change the HMI button actions.
AI	Change AI settings.

15.1 General

Item	Meaning
Distance	Set unit to mm or m.
Angle	Set unit to degrees or rad.
Reference Frame	Set to base (of robot), tool or world. The selected reference frame determines in which coordinate system incremental motions (Jog Mode) will be performed.
Select Tool	Select a tool that is globally considered (while saving a point, in **ZeroG** mode or Jogging, ...)
External FTS	Select an external force torque sensor (FTS) to be used with your robot. External FTSs can be added and configured via the Force Torque Sensor wizard (**Home > Wizard > Force Torque Sensor**).
Line	Use slider to set the percentage of the executed path to be shown. A visible line in the GUI will show the path of the TCP. The time until the disappearance of this line can be changed here.
Point Interaction	Enable or disable **Point Interaction**. If enabled, you can select a point in the 3D visualization space to open the **Goto Points** menu. A point can be selected by clicking on it.



E.g. **Point Interaction** – **P1** point selected

E.g. **Axis Interaction** – Joint **A4** selected

Axis Interaction	Enable or disable **Axis Interaction**. If enabled, you can select a robot joint in the 3D visualization space to open the **Move Joint** menu for the selected Joint. Select a joint by clicking on it.	E.g. trying to concurrently access the UI 
Allow Multiple UI Clients	If multiple user interface (UI) clients are allowed, you can concurrently control the robot from different UI instances (e.g. from the Teach Pendant and a browser on an external personal computer (PC) connected to the robot via Ethernet). If you disallow multiple UI clients, only a single UI instance is allowed to control the robot at a time. If you try to connect to the robot with an additional UI instance, you will be asked to terminate the existing one first.	

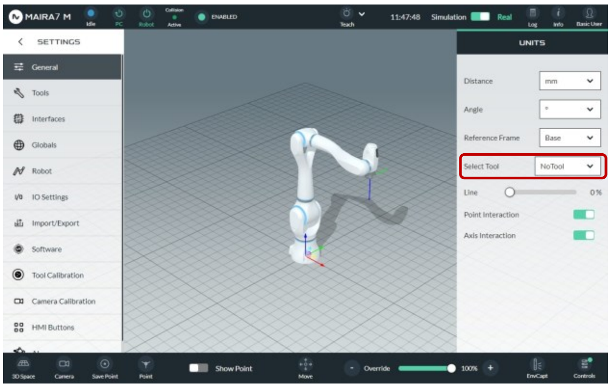
15.2 Tools

Edit the items.

Add and configure tools.

► Create a simple interface to various types of grippers and tools.

► Define the TCP position and dynamical parameters of the Tool.



The correct setup of the tool allows the operator to

► Teach pick & place operations with a gripper, e.g., for **Palletizing**.

► Achieve the desired motion and precision of the Tool Center Point (TCP).

► Operate the robot in **ZeroG mode**.

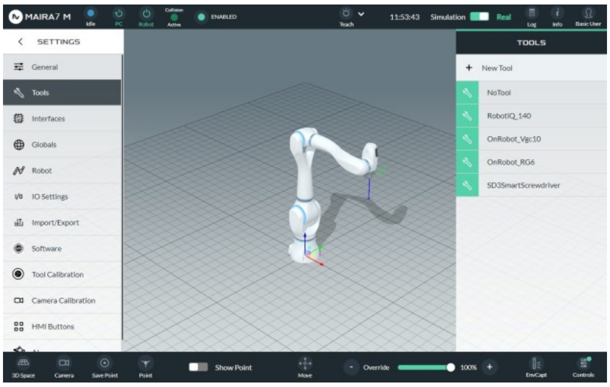
► Use the robot with a sensitive collision detection.

Add a new tool or edit an existing tool.

► Tap New Tool to add a new tool or select an existing tool to edit.

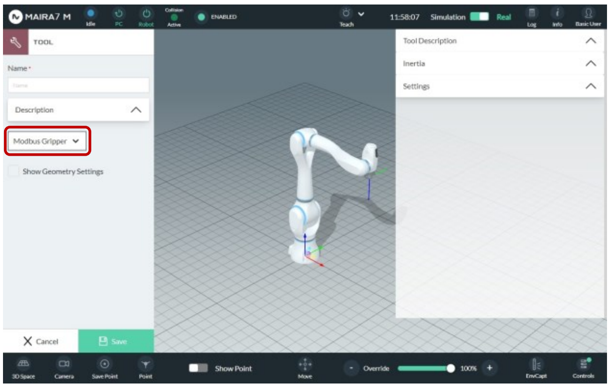
Edit a Tool

► Edit name and add a description.



15.3 Interfacing to Grippers

► For interfacing with external grippers two types of grippers are supported: **Standard Gripper** (General Purpose Input Output (GPIO)) Grippers and **ModbusRTU Grippers.** Choose your interface.



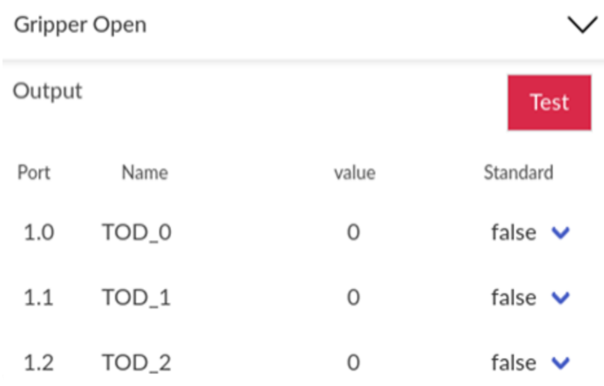
► **GPIO Grippers**

GPIO Grippers can be triggered using the **Analog or Digital Outputs** of either the **Tool Outputs** at the Robot's Flange or the **Controller Outputs** inside robot's Control Box. For detailed information for the Outputs, please refer to the robot User Manual.

Once one Output type is selected, you can define triggers for the **Gripper Open** and **Gripper Close** commands as required by the gripper in question. For this, please refer to the documentation of your gripper.

To test the correct behavior of the gripper, use the **Test** button to verify the communication.

The specified commands **Gripper Open** and **Gripper Close** are afterwards available in the Program Tree Generation.



NEURA ROBOTICS

► ModbusRTU Grippers

When selecting a **ModbusRTU Gripper,** more options to define the **Gripper Open** and **Gripper Close** States are possible.

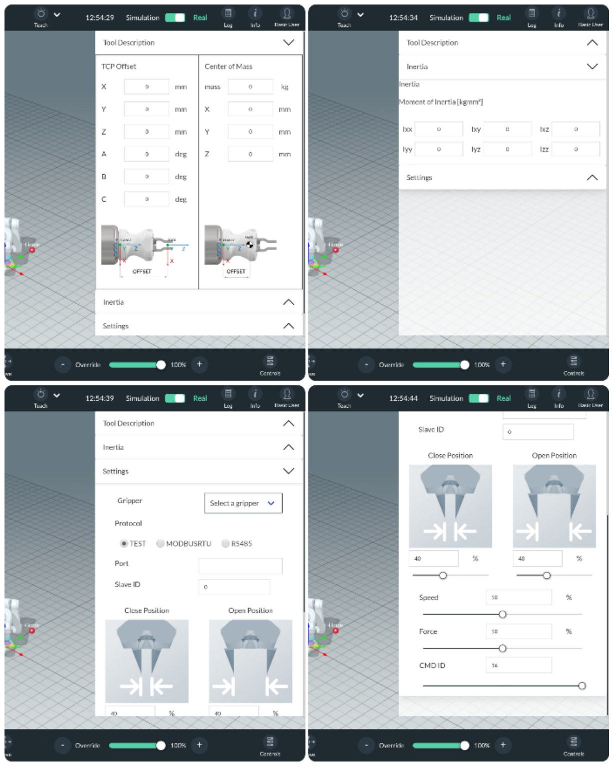
First, choose one of the supported Grippers from the dropdown menu.

Choose **MODBUSRTU** and insert the correct **Slave ID** according to the documentation of the Gripper. For example, the standard Slave ID for Robotiq Grippers is 9, and for OnRobot Grippers it is 65.

Additional settings from the Gripper can be triggered using the **CMD ID.** The ID is specified by the manufacturer of the Gripper.

The **Close Position** and **Open Position** of the gripper can be specified as a percentage of the maximum range, which depends on the gripper type. Additionally, the speed for opening and closing the gripper, as well as the force can be specified.

The available settings always depend on the gripper type and are given in the documentation of the gripper.



15.4 Kinematic & Dynamic Tool Description

► Enter the kinematic parameters for the **TCP Offset**.

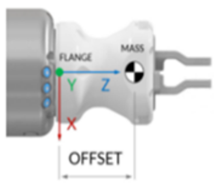
- The parameters **A B C** describe the rotation of the TCP frame with respect to the flange frame using the **roll**, **pitch yaw** or **Z, Y', X" Euler** convention.
- The **roll angle A** describes the rotation around the **X-axis** of the original robot flange frame, the **pitch angle B** describes the rotation around the **Y-axis** of the original robot flange frame, and the **yaw angle C** describes the rotation around the **Z-axis** of the original robot flange frame. The roll-pitch-yaw convention corresponds also to the **ZY'X" Euler convention**.
- An alternative way interpreting the angles **A, B,** and **C** is the **X, Y', Z" Euler** convention. Here, the rotation from robot flange to TCP coordinate system is described by a **rotation around the flange Z-Axis** with the **angle C**, followed by a **rotation around the new Y'-Axis** with the **angle B** and a **rotation around the twice rotated X"-**Axis with the **angle A**.
- The parameters **X Y Z** for the **TCP Offset** describe the **translatory offset** from the robot's flange frame to the

TCP frame in the three axis directions. The offset is described in the flange frame (so that the rotation parameters **A B C** do not influence the translatory offset).

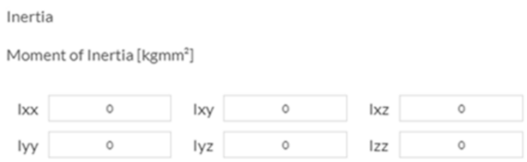
The kinematic Tool parameters are important for performing robot motions where the Tool Center Point is supposed to follow a trajectory precisely, or to reach a point precisely. Wrongly inserted values reduce the precision of the executed tasks.

► Enter dynamic parameters for the **Tool**.

- Enter the **mass m** of the tool in [kg]. The mass needs to be positive.
- Insert the properties **X, Y, Z** of the center of mass of the tool in [mm].
- The coordinates of the center of mass are defined from the flange frame to the center of mass and are given in flange coordinate system. The TCP offset does not influence the center of mass.



The **moment of inertia** of the tool in [kgmm²] can also be specified if available. It describes the tool's tendency to resist rotation.



If available, the parameters **Ixx, Ixy, Ixz, Iyy, Iyz, and Izz** can be inserted. The parameters are given in the flange frame and eventually need to be transformed to the frame.

The dynamic tool parameters heavily influence the performance of the Cobot. If the dynamic parameters of the tool are not properly set up, the robot's path following accuracy is reduced, the ZeroG mode might show a drift, and the collision detection might not be as robust or sensitive as desired. Not inserting the correct dynamic tool parameters is similar to a constant force acting on the robot, that can lead to a falsely identified collision or a motion during ZeroG mode. It is the responsibility of the integrator to ensure the proper operation of the robot, including the correct specification of the tool parameters for a safe interaction between operator and Cobot.

► Alternative to manual insertion of the dynamical tool parameters: the **Dynamic Parameter Identification Wizard (DPI)**

- DPI identifies all dynamical parameters (masses, center of masses, ...) of the entire robot.
- It can be used to identify the tool in complex load situations, e.g., when there are cables mounted onto the robot and the identification of the tool parameters is difficult. It is also recommended for simple tools.
- Note: when using DPI to estimate the tool parameters, the manually inserted values for the dynamical parameters need to be set to 0. The tool parameters are included into the model of the robot. When unmounting the tool, DPI needs to be executed again to identify the robot parameters without the tool once again.
- DPI is password protected and the responsibility of the integrator.

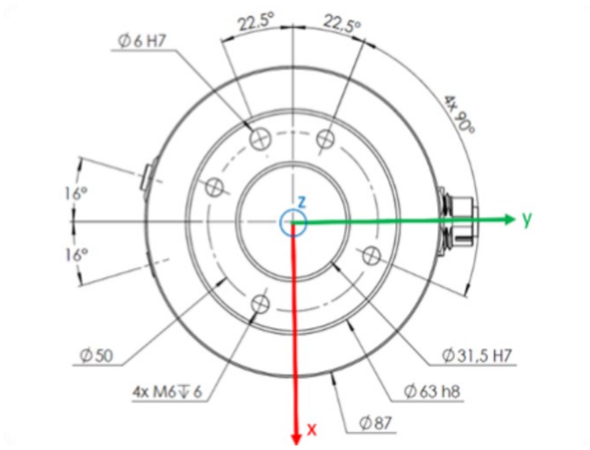
For more information, see the section of the DPI wizard.

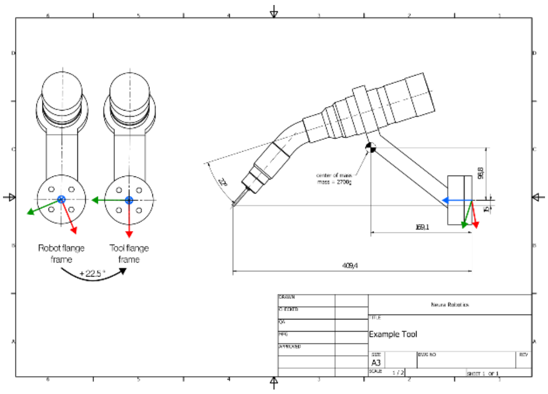
15.5 Example of Tool Description Settings

This section will illustrate the correct setup of the kinematic and dynamic tool parameters based on the following example tool given in the figure XX. Additionally, consider the dimensions of the flange.

► We will start with entering the values for the **TCP Offset**

- From the Example Tool drawing, one can quickly see that the **Z** value of the TCP offset is **Z = 409,4 mm**.
- It is important to note that the hole pattern of the flange is rotated by 22.5° compared to the X-Axis of the robot's flange.
- Thus, the x- and y-coordinate axes of the robot's flange do not align with the x- and y-coordinate axes of the tool flange and the offset of 15mm needs to be distributed among the robot's flange x- and y-axes.
- In the tool flange drawing, the positive x-axis of the tool flange points upwards. Considering the hole-pattern as a pre-rotation, it results in the **kinematic tool offset X = cos(22,5°) * -15mm = -13,85mm.**
- The positive y-axis of the flange points to the left of the figure, thus the **kinematic tool offset Y = sin(22,5°)*-15mm = -5,74mm.**
- For the choice of the tool coordinate axes directions, we will now adapt **ZYX Euler parameters C, B, A.**
- It is desirable to choose the tool z-Axis in the same orientation as the tip of the illustrated welding tool. This would require a rotation around the z-axis of the flange by 22,5° in the negative direction to compensate for the misalignment of the holes and the flange coordinate system. Thus, the parameter **C = -22.5°.**
- Next, we will perform a rotation of B **= -55°** around the new y-Axis. Now the z-axis of points in the same direction as the TCP tip which is the desired outcome. A rotation





around the new X”-axis is not necessary; thus, we choose **A = 0°.**

- Now we will determine the **dynamic parameters** for the tool
  - The mass can be read from the tool drawing and is **mass = 2700g = 2.7kg.**
  - Again, the value for the center of mass in Z direction can be directly read from the drawing: **Z = 169,1mm.**
  - Similarly, as for the TCP Offset, we need to project the value of 98,8mm onto the x- and y-Axes of the flange using trigonometric function sine and cosine due to angle of 22.5° between flange x-axis and holes in the robot-flange.
  - The center of mass in X-direction of the flange frame can be computed by **X = cos(22,5°) * 98,8mm = 91,28mm.**
  - The center of mass in Y-direction of the flange frame can be computed by Y **= sin(22,5°) *98,8mm = 37,81mm.**
  - Due to the lack of information regarding the moments of inertia, they will be omitted in this example.

15.6 Interfaces

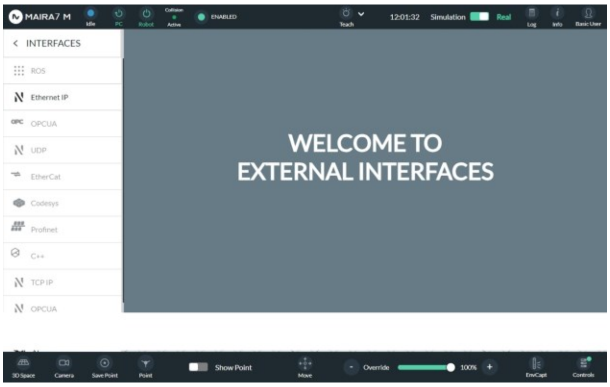
The **Interfaces** menu allows the user to activate and configure different external interfaces.

► To access the interface menu, navigate to **Home >> Settings >> Interfaces** via the main menu

➔ The **INTERFACES** menu is open, presenting a list of interfaces (greyed out options not available yet or further configuration necessary (e.g. upload of configuration files to the robot controller via NeuraUSB))

► To activate/configure a certain interface, select it from the list

➔ The corresponding settings page opens



NOTICE	With external interfaces activated (except for Ethernet IP), the robot can no longer be used by GUI (e.g. jogging or playing programs). Deactivate external interfaces to regain control via GUI.

NOTICE	External interfaces work in both simulation and real robot mode. Safety-related, the user will not be able to switch between modes while an interface is activated. The user is supposed to deactivate the interface, then switch modes and reactivate the interface after the switch.

NOTICE	Interface configurations cannot be altered while interfaces are active. Interfaces must be deactivated first to change settings and reactivated afterwards.

► Configure the desired interface according to your needs

- ROS
- Ethernet IP
  - .yaml config file must be uploaded to robot via NeuraUSB first
- OPCUA
  - Currently the robot cannot be controlled by the OPCUA interface.

Only robot state readings are obtained.

- UDP
- EtherCat
  - The robot can be controlled via EtherCat by using a Beckhoff EL6695 bridge terminal. A configuration file must be uploaded to the robot in advance.
- Codesys
- Profinet
- C++
- TCP IP

15.7 Globals

Change or delete the global settings like **Points** and **Recorded Paths**.

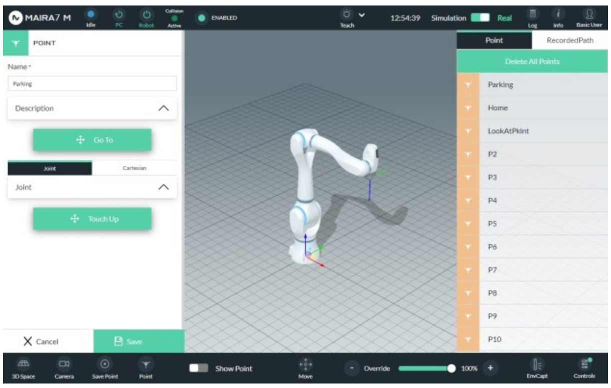
15.7.1 Points

Edit or **delete** global points.

► Choose a point.

► Select the **Edit** button to edit it or the **Waste bin** to delete it. Deleting a point will permanently delete this point, there is no recovery possible.

► Select **Delete All Points** to delete all points in the list. Deleting all points will permanently delete points, there is no recovery possible (Parking and Home points cannot been deleted).

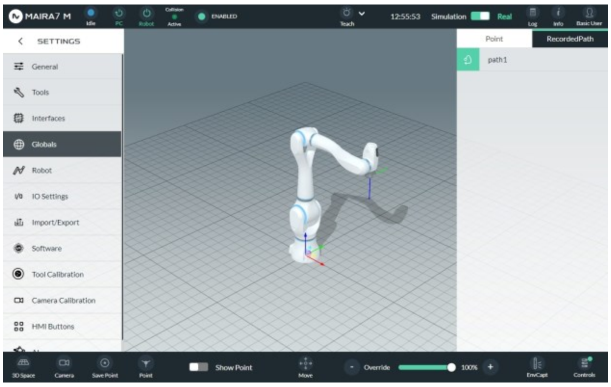


15.7.2 Recorded Path

Edit or **delete** recorded paths.

► Choose a recorded path.

► Select the **Edit** button to edit it or the **Waste bin** to delete it. Deleting a recorded path will permanently delete it, there is no recovery possible.



15.8 HMI Buttons (Touch Buttons)

Change the function allocation of the HMI buttons (touch buttons) on the top of the robot's head (link 6).

1. Select one of the buttons on the screen to configure.
2. Select an action from the dropdown menu.
    - **Gripper Open**
    - **Gripper Close**
    - **ZeroG On**
    - **ZeroG Off**
    - **Save Point**
    - **Delete Point**
    - **none**
3. Select **none** if no action shall be assigned to the button.
4. Select one of the arrows to allocate a swipe gesture into the corresponding direction. Same actions as for the buttons can be assigned.
5. Select **Save** to save your settings or select **Reset to Default** to assign the default setting.

NOTICE	Use the arrows depicted in the GUI representation of the button layout to identify the physical touch buttons on the robot's head. The camera in the robot's head must be pointed into viewer direction to match layout orientation. See overlay of physical and GUI representations to the right. Selected buttons/arrows are indicated by a green color.

Camera

15.9 Robot

Change robot specific settings for Joint, Cartesian, Robot Parameters and Gravity Vector.

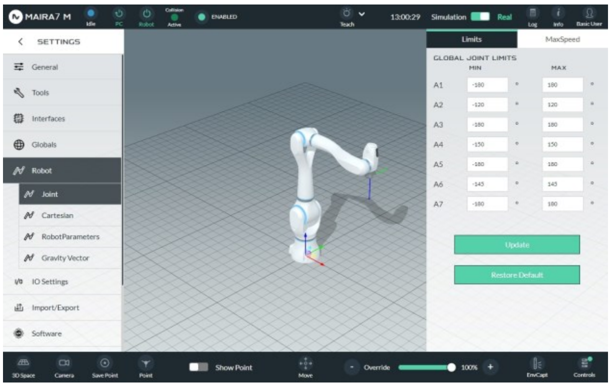
15.9.1 Joint Limits

Change the settings for all joints.

► **Joint Limit Settings**

- The joint limits determine the range of motion of the robot.
- Edit the joint limits to restrict the robot’s range of motion in restricted workspaces.
- The limits are considered in all motion types, e.g., in Jog mode, or in Program mode.

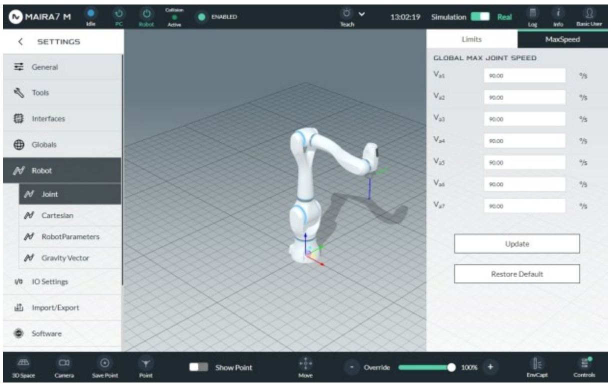
Click **Update** to save any changes made or **Restore Default** to reset the parameters.



► Joint MaxSpeed Settings

- The joint maximum speed limits determine the speed of each axis of the robot.
- Edit the joint limits to restrict the maximum speed at which the robot performs its tasks.
- The limits are considered in all motion types, e.g., in Jog mode, or in Program mode. The global limits overwrite other local settings, e.g., in Jog mode.

Click **Update** to save any changes made or **Restore Default** to reset the parameters.



15.9.2 Cartesian Limits

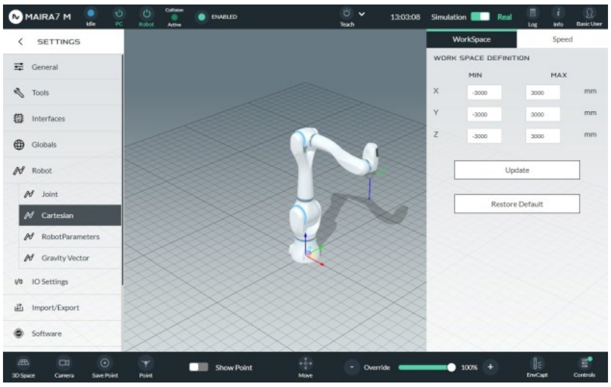
Change the parameters for the cartesian space.

Edit cartesian settings.

► **Cartesian Workspace Settings**.

- Create a box-shaped maximum cartesian workspace for the TCP of the robot by specifying the maximum and minimum values for the three axes directions.
- The maximum and minimum values are with respect to the base frame of the robot.
- The specified Cartesian Workspace is inclusive, meaning that the robot cannot leave the workspace.
- The workspace is **active** in **Joint** and **Cartesian Jog Mode**.

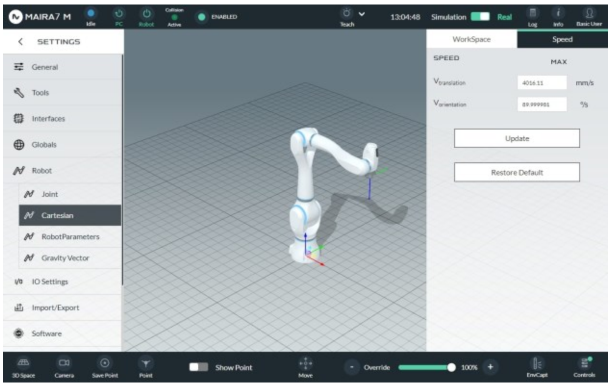
Click **Update** to save any changes made or **Restore Default** to reset the parameters.



► **Cartesian Speed Settings**.

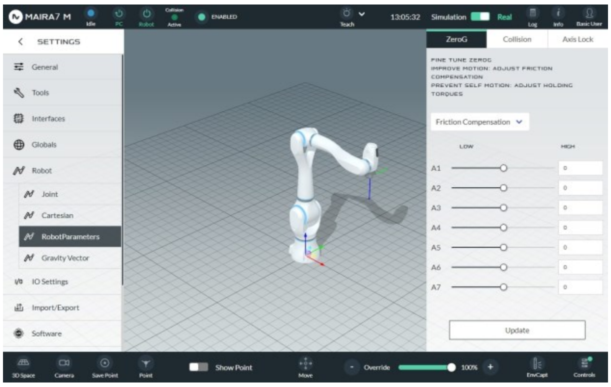
- Specify the desired maximum velocities for the robot's TCP in all Cartesian directions.
- It is possible to define the maximum velocities in all three translatory directions by inserting the desired values for **Vx, Vy**, and **Vz**.
- The maximum **angular velocities** of the TCP in x, y, and z-axis directions can be specified using **Va**, **Vb**, and **Vc**.
- The specified values are the global settings that cannot be exceeded in any mode by the robot. They also constitute the maximum values in **MoveL** and **MoveC** motions.

Click **Update** to save any changes made or **Restore Default** to reset the parameters.



15.9.3 Robot Parameters

In this tab, different Robot Parameters can be set. This includes adjusting the **ZeroG mode**, **Collision parameters** and to **lock Axis** for ZeroG mode.



► **Fine Tuning ZeroG mode Parameters**.

- Using the available slider in **Friction Compensation** and in **Holding Torques** allows the operator to make the ZeroG mode more robust or smoother.
- The **Friction Compensation Sliders** can be used to make the motion in ZeroG mode smoother or stiffer. High Friction Compensation leads to smoother motion. However, increasing the value can lead to unwanted self-motion. It is the responsibility of the operator to choose proper values.
- The **Holding Torques Sliders** can be used in case the robot moves on its own when turning on ZeroG mode, e.g., when a tool is not properly set up. Increasing the Holding Torques is recommended when the robot moves downwards.

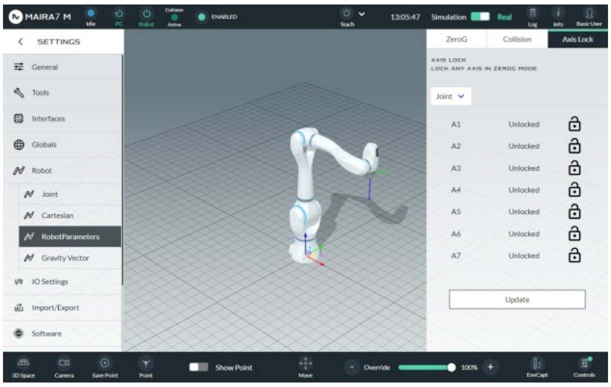
CAUTION

When setting ZeroG too soft, robot might slowly collapse or continue movement even when operator releases from hand.

Click update to save any changes made.

► **Fine Tuning Collision Detection Parameters**.

- This password protected Setting can be used to adjust the collision detection parameters to either make the collision detection more robust or to increase the sensitivity.



CAUTION

When collision is set too hard, this could harm operator when a collision is not detected.

- When collision is set too soft, a collision might be "detected" even when no physical collision happened.

Click update to save any changes made.

► **Axis Lock for ZeroG mode**.

- This setting allows to lock one or several axes in **ZeroG mode.**
- Tick the axis to be locked and update the settings by confirming with the password.
- Note: Only lock an axis when **ZeroG mode** is not activated.

Click update to save any changes made.

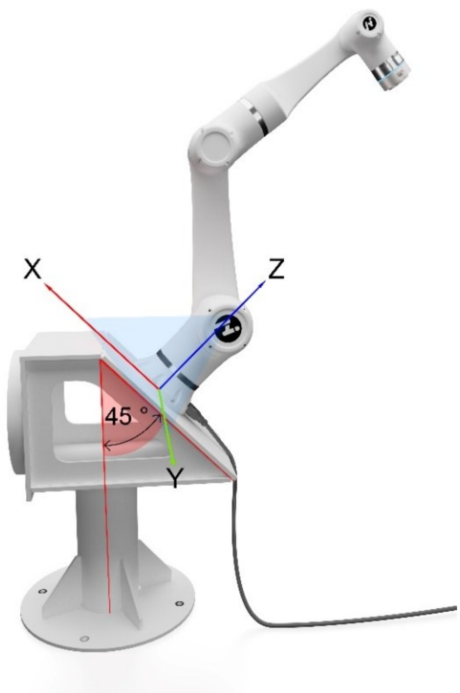
15.9.4 Gravity Vector

Setting the **gravity (acceleration) vector** is necessary when the robot is mounted upside down, on a wall or at an angle.

► Based on the specified values, the **orientation of the base of the robot** is adjusted.

► Settings the gravity vector is necessary for

- **ZeroG mode**
- **Collision Detection**
- High path and absolute **accuracy**



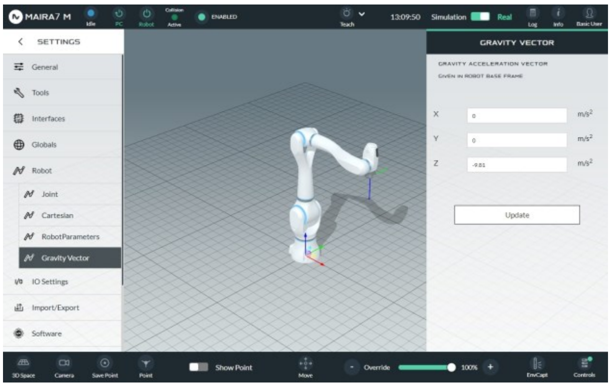
► The **standard gravity vector** is **[0.0, 0.0, -9.81] m/s²** when the robot is **mounted upright on the floor**. In this case the gravity acceleration of the earth points in the negative z-Axis of the robot's base coordinate system.

► The norm of the gravity vector (sqrt(x^2+y^2+z^2)) needs to 9.81m/s^2. The norm of a vector can be computed as: norm ([x y z]) = sqrt(x^2+y^2+z^2).

► When the robot is mounted upside down, the gravity vector is [0.0, 0.0, -9.81] m/s^2 because the earth's gravity points in the direction of the positive z-Axis of the robot's base.

► When the robot is mounted on a stand with its cable pointing towards the floor at a 45° angle, the gravity vector is [-sin(45°)*9.81, 0.0, -cos(45°)*9.81] m/s^2 because the earth's gravity points in the direction of both the negative x- and y-Axes of the robot's base to equal parts. There is no gravitational acceleration acting in the direction of the robot's base z-Axis.[6]

► Changing the gravity vector is password protected and is the responsibility of the authorized integrator. An incorrectly set gravity vector leads to undesirable behavior in **ZeroG** mode and collision detection.



15.10 AI

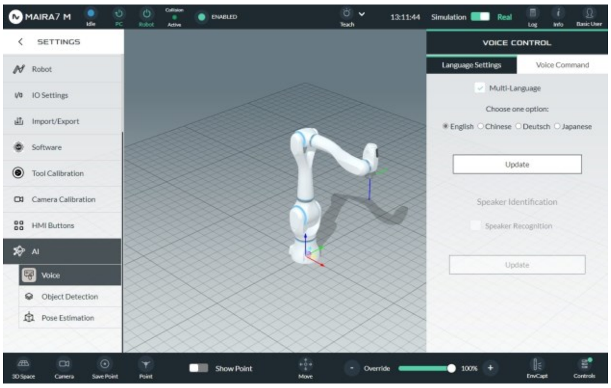
Change the settings of the Artificial Intelligence features.

15.10.1 Voice

Change language or voice command settings.

Language Settings:

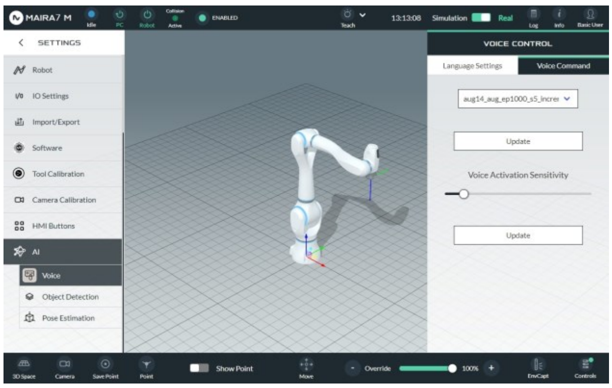
1. Check the **Multi-Language** box to enable language selection. This allows the user to change the language via the voice command "change language".
2. Check a radio button to select a language. This defines the default language pack loaded when powering up the robot.
3. Select **Update** to apply the settings.



Voice Command Settings:

1. Select a voice command model.
2. Select **Update** to apply the settings.

The default model is well trained and is used to activate the voice commanding system after turning it on from GUI. The model is sensitive to *'Hey MAiRA'* activation word and does not depend on the selected language.



1. Use the slider to change **Voice Activation Sensitivity**

[6] Figure shows LARA, same for MAiRA.

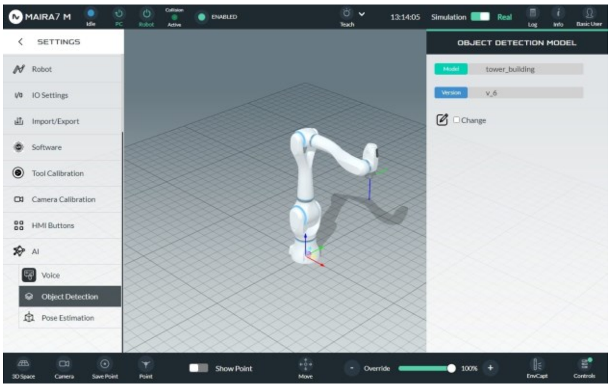
2. Select **Update** to apply the settings.

15.10.2 Object Detection

The name and version of the object detection model currently used are shown in the corresponding fields.

To change the object detection model:

- ► Press the **Edit** button (pencil icon)
- ► Select the **Change** checkbox
- ► Select a **Model** from the dropdown menu
- ► Select a **Version** for the model from the dropdown menu
- ► Press **Update** to apply settings

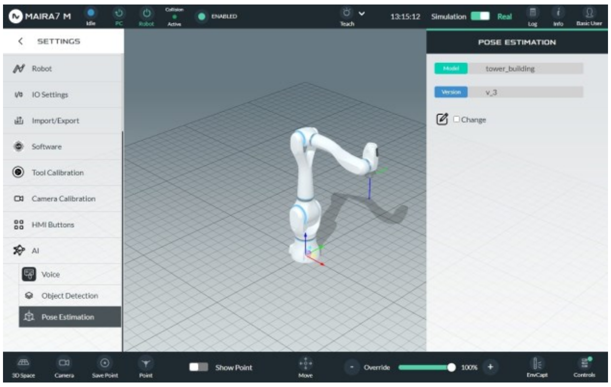


15.10.3 Pose Estimation

The name and version of the pose estimation model currently used are shown in the corresponding fields.

To change the pose estimation model:

- ► Press the **Edit** button (pencil icon)
- ► Select the **Change** checkbox
- ► Select a **Model** from the dropdown menu
- ► Select a **Version** for the model from the dropdown menu
- ► Press **Update** to apply settings



15.11 I/O Settings

15.11.1 Controller Digital

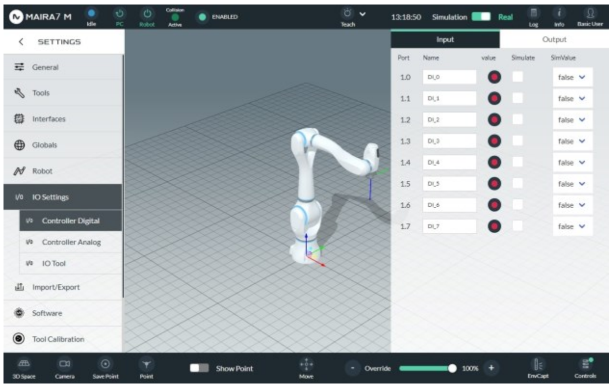
Change settings for the digital controller board. The inputs can be used to trigger a program e.g. if the program waits for an input to be set.

Outputs can be used to trigger external equipment and can be set in the program (**SetVariable**) or for testing via the Settings menu **Settings >> IO Settings >> Controller Digital**.

15.11.1.1Input

► Set digital input ports

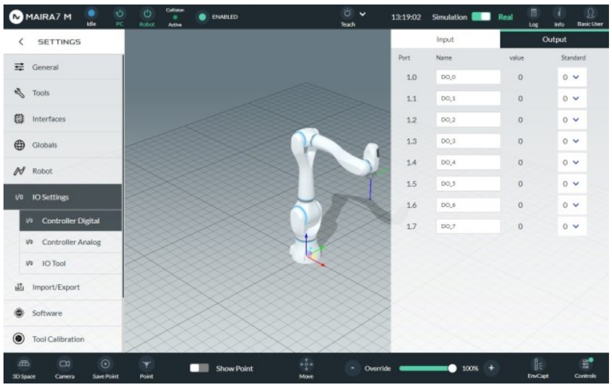
Item	Meaning
Port	Port number
Name	Edit name if required
Value	Current value
Simulate	Check box to simulate port
SimValue	Select simulation value



15.11.1.2 Output

► Set digital output ports

Item	Meaning
Port	Port number
Name	Edit name if required
Value	Current value
Standard	Select standard value



15.11.2 Controller Analog

15.11.2.1 Input

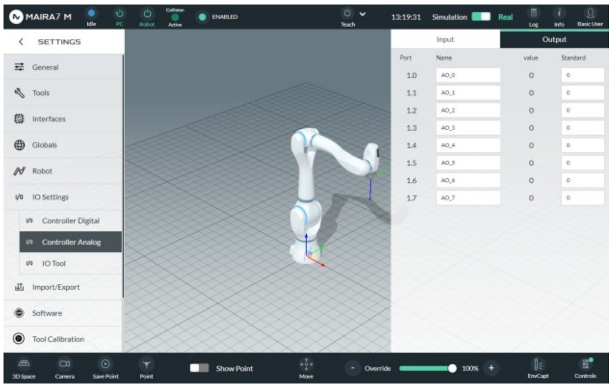
► Set analog input ports

Item	Meaning
Port	Port number
Name	Edit name if required
Value	Current value
Simulate	Check box to simulate port
SimValue	Set simulation value.

15.11.2.2 Output

► Set analog output ports

Item	Meaning
Port	Port number
Name	Edit name if required
Value	Current value
Standard	Set standard value

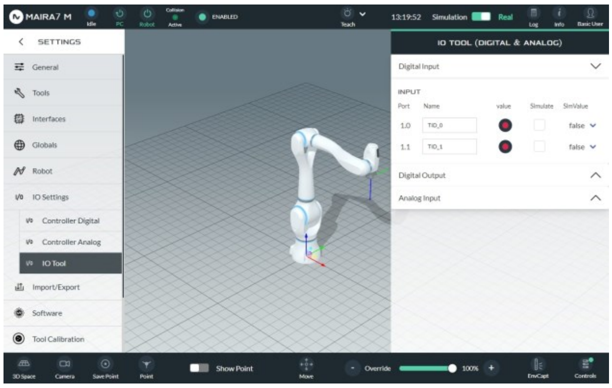


15.11.3 IO Tool

15.11.3.1 Digital Input

► Set digital input ports

Item	Meaning
Port	Port number
Name	Edit name if required
Value	Current value
Simulate	Check box to simulate port
SimValue	Select simulation value.



15.11.3.2 Digital Output

► Set digital output ports

Item	Meaning
Port	Port number
Name	Edit name if required
Value	Current value
Standard	Select standard value

15.11.3.3 Analog Input

► Set analog input ports

Item	Meaning
Port	Port number
Name	Edit name if required
Value	Current value
Simulate	Check box to simulate port
SimValue	Set simulation value

15.11.4 Extended IO

The Extended IO feature lets you connect additional IO devices to the robot.

This feature requires additional hardware. Currently only the IFM AL1342 (Modbus TCP) device is supported. For this device, a standard configuration file is prepared on the robot. This configuration file lets you use 4 inputs (robot: EDI_1, EDI_2, EDI_3 EDI_4; IFM device: X01, X03, X05, X07) and 4 outputs (robot: EDO_1, EDO_2, EDO_3, EDO_4; IFM device: X02, X04, X06, X08) with your robot and the IFM device. The standard configuration file requires the IFM device IP to be 192.168.2.250. Only one extended IO device can be used at a time.

Note on the upload IO feature: the upload IO feature is not yet available. In a future release it lets you upload custom configuration files for extended IO devices to your robot.

To use an extended IO device with your robot:

- ► Set the extended IO device IP to 192.168.2.250 by following the manufacturer's instructions
- ► Connect your sensors or actuators to the extended IO device by following the manufacturer's instructions.
- ► Connect the extended IO device to your robot's controller by using an Ethernet cable (M12 to RJ45), follow the manufacturer's instructions for the extended IO device and the robot's user manual.
- ► Power on the extended IO device.
- ► In the robot's graphical user interface (GUI), navigate to **Home > Settings > IO Settings > Extended IO**.
- ► Tap your IO device in the device list to select it.
- ► Tap the **View** icon.
- ► Tap **Connect** in the I/O Info box.

► Successful connection will be confirmed with a pop-up message.

➔ The extended IO device is now connected to the robot and ready to be used.

To disconnect an extended IO device:

► In the robot's graphical user interface (GUI), navigate to **Home > Settings > IO Settings > Extended IO**.

► Tap your IO device in the device list to select it.

► Tap the **View** icon.

► Tap **Disconnect**.

► Successful disconnection will be confirmed with a pop-up message.

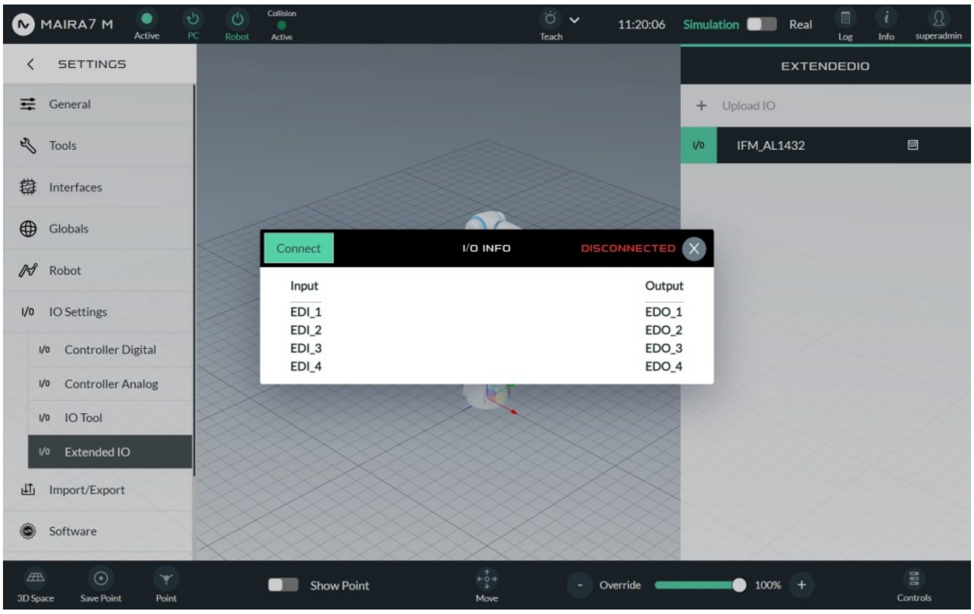
➔ The extended IO device is now disconnected from the robot and cannot be used anymore.

To view an extended IO configuration:

► Navigate to **Home > Settings > IO Settings > Extended IO**.

► Tap your IO device in the device list to select it.

► Tap the View icon.



15.12 Import/Export

15.12.1 Creating a NeuraUSB/SecureSync flash drive

NOTICE

From software version v4.19.0-alpha.16, the name that must be given to flash drive was changed from **NeuraUSB** to **SecureSync**. Please name the device as required by your software version.

- Name the flash drive **NeuraUSB** for software versions till v4.19.0-alpha.15 (and below).
- Name the flash drive **SecureSync** for software versions from v4.19.0-alpha.16 (and above).

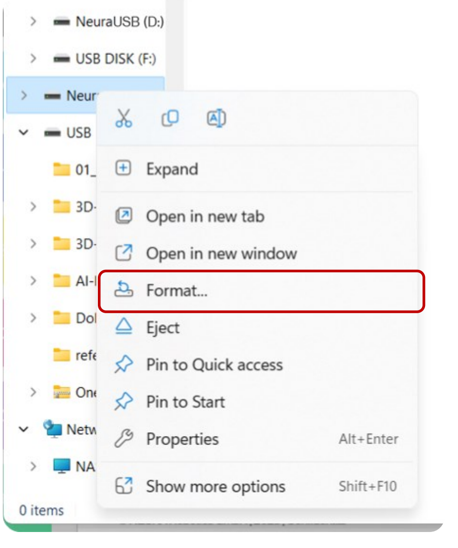
A **NeuraUSB/SecureSync** flash drive must be used for various file operations, like creating or restoring backups, installing software updates or importing external programs.

The following section explains how a **NeuraUSB/SecureSync** flash drive is created by formatting and renaming a flash drive using Windows 10.

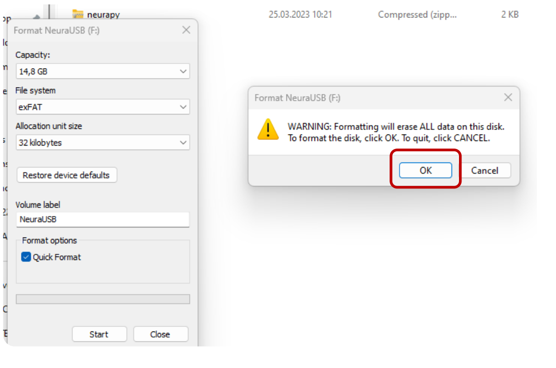
- Insert a USB flash drive into your Windows 10 machine.

The file format of the flash drive must be **exFAT** to work with the robot control PC. Therefore, the flash drive will be formatted with **exFAT** in the next steps.
Be aware that the USB flash drive will be completely wiped within this process.

- Open **Windows File Explorer**.
- Search for the inserted USB flash drive in the left overview pane of the **Windows File Explorer**.
- Right-click on the USB flash drive and select **Format...**.



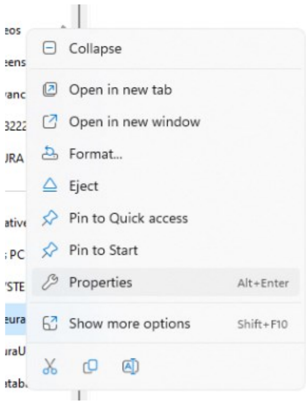
- The File system must be set to **exFAT**, so please select **exFAT** from the File system dropdown menu.
- Start the formatting process by clicking **OK**.



▶ Confirm the Warning Message about the wiping of all the data on the flash drive with **OK**.

▶ A popup message will confirm the successful formatting of the USB flash drive.

🗐 The name of the USB flash drive must be exactly **NeuraUSB** (till software version v4.19.0-alpha.15 and below) or **SecureSync** (from software version v4.19.0-alpha.16 and above). The drive will be renamed in the next steps.



▶ Again, search for the inserted USB flash drive in the left overview pane of the **Windows File Explorer**.

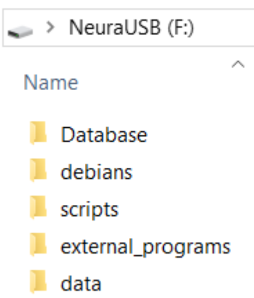
▶ Right-click on the USB flash drive and select **Rename**.

▶ Type in **NeuraUSB** (till software version v4.19.0-alpha.15 and below) or **SecureSync** (from software version v4.19.0-alpha.16 and above), note capitalization and press **Enter.**

➔ Your USB flash drive should now be named **NeuraUSB**.

➔ The flash drive is now ready to be used with the robot control PC.

Possible folder structure of the **NeuraUSB/SecureSync** flash drive:



Folder	Content
Database	Storage for backup data
debians	Storage for software update files
scripts	Storage of software update installation scripts
external_programs	Storage for external programs

data	Storage of metatdata
data/process	Storage for external source files

The **Database** folder will be automatically created in the root directory of your **NeuraUSB/SecureSync** flash drive when creating a robot backup via the Teach Pendant for the 1st time. See 15.12.2.

The **debians** and **scripts** folder are related to software updates. For software updates those two folders will be provided and must be copied from your PC to your **NeuraUSB/SecureSync** flash drive (root directory) before mounting it to the control box. See 15.13.

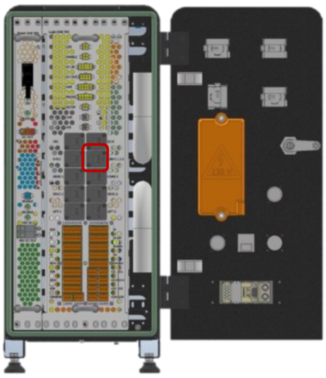
The **external_programs** folder must be created manually on your PC if you want to import Python scripts onto your robot. Place your scripts into the **external_programs** folder before mounting the USB flash drive to your control box. See 15.12.4.

The **data** folder and **process** subfolder are used for **ExternalSource** (see 15.14) configuration files. Those two folders must also be created manually on your PC. Place your configuration files into the **process** subfolder before mounting the **NeuraUSB/SecureSync** flash drive to your control box.

15.12.2 Export (Backup robot data)

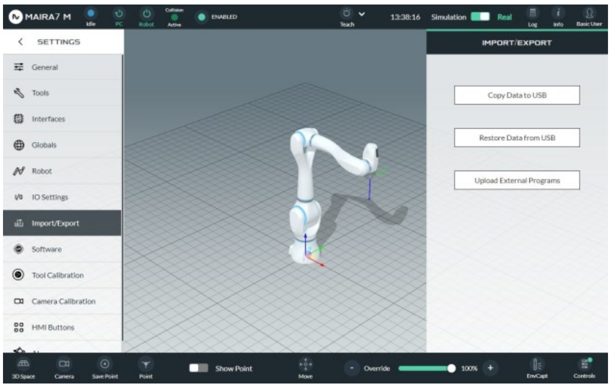
The **Export (Copy Data to USB)** feature enables the user to create backups of robot data onto a USB flash drive.

The following section explains the procedure by using a prepared **NeuraUSB/SecureSync** flash drive (see 15.12.1).



- 🗐 The import/export feature for creating backups of your robot data only works with USB flash drives that have the required name (**NeuraUSB/SecureSync**) and file format (**exFAT**). For instructions on how to prepare your USB flash drive please refer to paragraph 15.12.1.

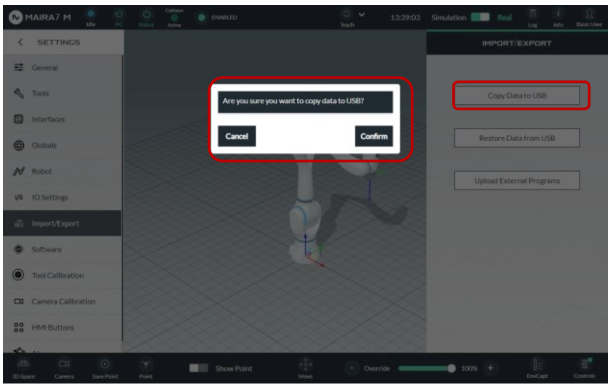
- 🗐 The **NeuraUSB/SecureSync** flash drive must be plugged into the control box. Use the right one of the two USB-C ports within the control box to connect your **NeuraUSB/SecureSync** flash drive (use a suitable USB port adapter if necessary).



- ▶ To create a backup of your robot data, use the Teach Pendant and tap on **Settings** in the left-hand menu pane. Then tap on **Import/Export**.

- ▶ You will be prompted to enter your user password.

- ▶ Type in your user password and tap on **Validate** to confirm.

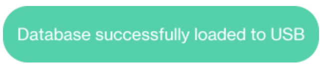


- ▶ To create a backup of your robot data, tap on **Copy Data to USB** in the right-side settings pane.

- ▶ You will be asked if you are sure you want to copy data to USB.

- ▶ Proceed with the backup process by tapping **Confirm**.

- ➔ The database export will finish with the popup message **Database successfully loaded to USB**.



NOTICE	Backups are not incremental. A new backup on a USB flash drive with an existing backup will override the existing one. To keep the initial backup, please use your PC to extract the backup folder from the Database folder on the USB flash drive. Rename the backup folder with a suitable name and timestamp. Then create your new backup.

15.12.3 Import (Restore robot data)

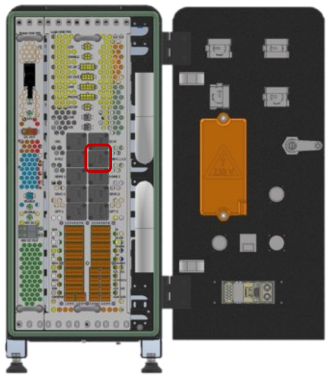
The **Import (Restore Data from USB)** feature enables the user to restore backups of robot data from a USB flash drive.

The following section explains the procedure by using a prepared **NeuraUSB/SecureSync** flash drive (see 15.12.1) containing backed-up robot data (see 15.12.2).

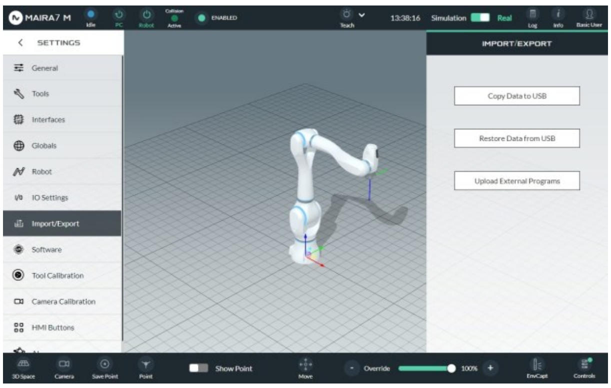
To restore a backup, the **NeuraUSB/SecureSync** flash drive containing a backed-up database must be plugged into the control box. Otherwise, it's impossible to perform the import operation.

For instructions on how to create a backup of your robot database, please refer to paragraph 15.12.2

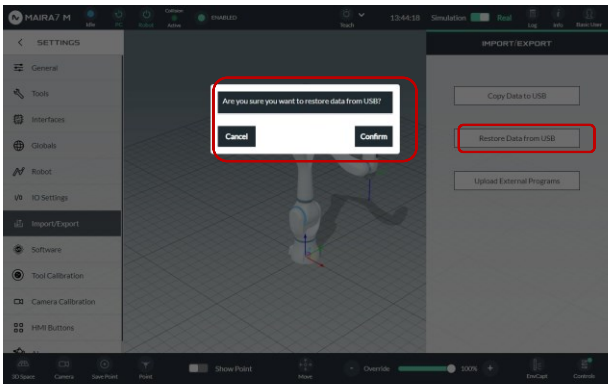
Use the right one of the two USB-C ports within the control box to connect your **NeuraUSB/SecureSync** flash drive (use a suitable USB port adapter if necessary).



- To restore a backup of your robot data, use the Teach Pendant and tap on **Settings** in the left-hand menu pane. Then tap on **Import/Export**.

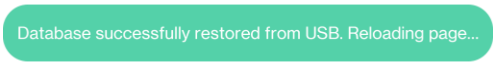


- You will be prompted to enter your user password.
- Type in your user password and tap on **Validate** to confirm.
- To restore a backup of your robot data, tap on **Restore Data from USB** in the right-side settings pane.



- You will be asked if you are sure you want to restore data from USB.
- Proceed with the restore process by tapping **Confirm**.

➔ The database import will finish with the popup message **Database successfully loaded from USB. Reloading page….**



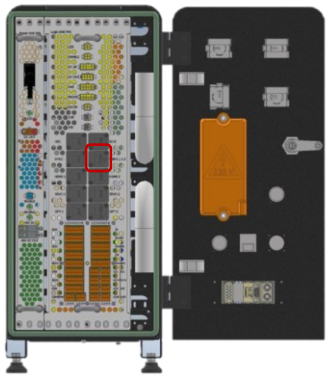
15.12.4 External Programs

The **External Programs** feature enables the user to run custom Python programs from the user interface.

Only the import of external programs will be shown here.

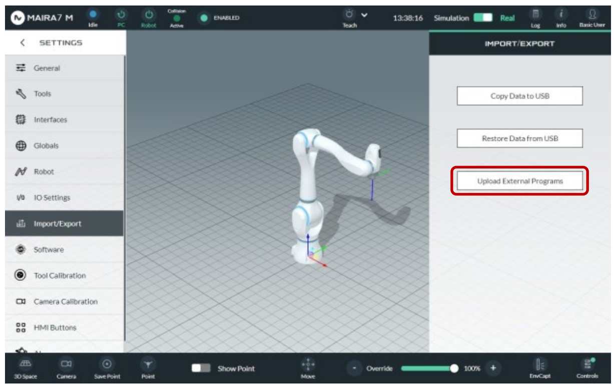
For information on how to program **MAiRA** using **NeuraPy**, please refer to the additional API documentation.

🗐 The External Programs feature for importing custom Python programs only works with a **NeuraUSB/SecureSync** flash drive. For instructions on how to create a **NeuraUSB/SecureSync** flash drive, please refer to paragraph 15.12.1.



🗐 Your external programs must be placed in a folder called **external_programs** on the **NeuraUSB/SecureSync** flash drive which must be plugged into the control box. Use the right one of the two USB-C ports within the control box to connect your **NeuraUSB/SecureSync** flash drive (use a suitable USB port adapter if necessary).

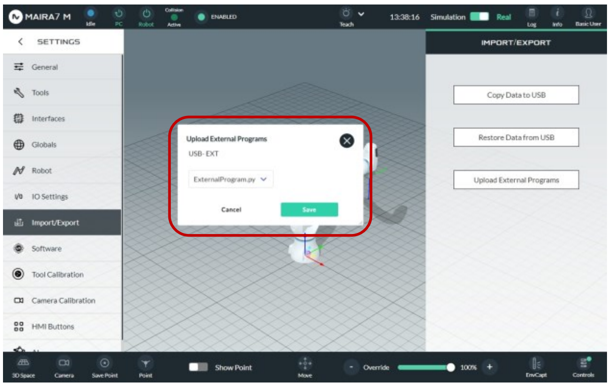
- ▶ To import custom Python programs, use the Teach Pendant and tap on **Settings** in the left-hand menu pane. Then tap on **Import/Export**.



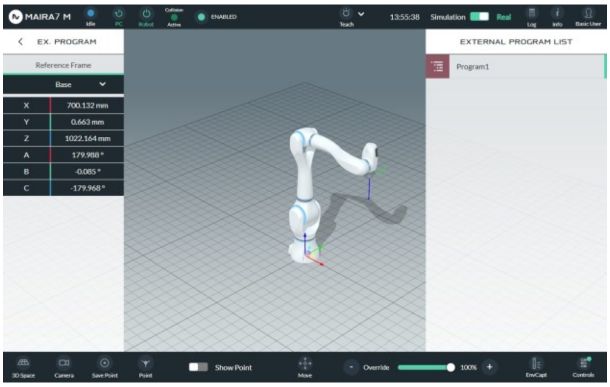
- ▶ You will be prompted to enter your user password.
- ▶ Type in your user password and tap on **Validate** to
- ▶ Tap on **Upload External Programs** in the right-side settings pane.

► Choose the programs you want to import.

► Click on **Save** to complete the import.



➔ The imported programs are now available under the **Ex. Program** tab in the main menu and can be executed like standard programs.



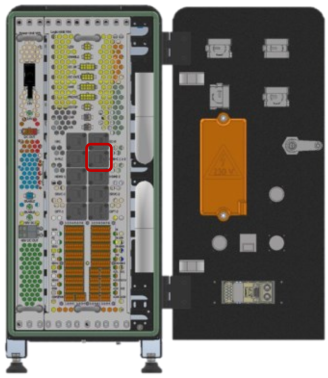
15.13 Software

The **Software Update (Install Software)** feature enables the user to install a new robot software version from a USB flash drive.

The following section explains the procedure by using a prepared **NeuraUSB/SecureSync** flash drive (see 15.12.1) containing the software installation files.

🗐 To install a new robot software version, the **NeuraUSB/SecureSync** flash drive containing the new software files must be plugged into the control box.

Use the right one of the two USB-C ports within the control box to connect your **NeuraUSB/SecureSync** flash drive (use a suitable USB port adapter if necessary).



Without a properly prepared **NeuraUSB/SecureSync** flash drive and the software update files in the required location, it's impossible to perform the update operation.

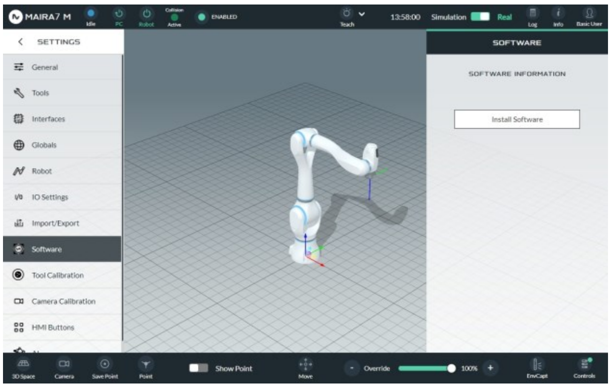
For instructions on how to create a

NeuraUSB/SecureSync flash drive, please refer to paragraph 15.12.1.

- You must backup your robot data onto the **NeuraUSB/SecureSync** flash drive before updating the software (see paragraph 15.12.1 on how to do so).

- Power off the robot by following paragraph 7.4.

- To perform a software update, tap on **Settings** in the left-hand menu pane. Then tap on **Software**.



- You will be prompted to enter your user password.

- Type in your user password and tap on **Validate** to confirm.

- To install new robot software, tap on **Install Software** in the right-side settings pane.

- You will be asked if you are sure you want to install the software.

- Proceed with the installation process by tapping **Confirm**.

- The software update process will take up to 10 minutes.
  Afterwards the GUI will prompt you with a message to restart the robot controller.

➔ The software update process is now completed.

- To restore your robot data from the backup created earlier, please refer to paragraph 15.12.3.

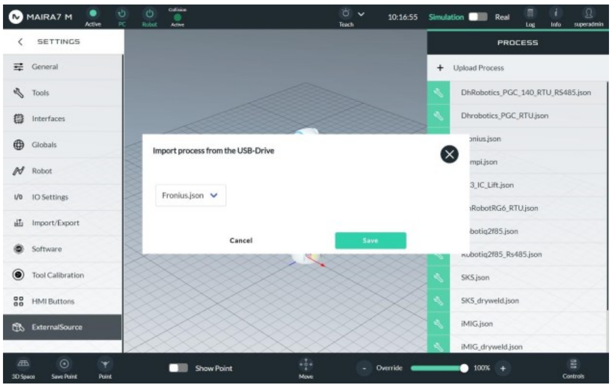
15.14 External Source

The **External Source** tab within the **Settings** menu allows you to upload configuration files to establish communication with external devices, e.g. welding power sources. To ease the creation of configuration files NEURA Robotics provides the standalone application “**External Source File Configurator**”.

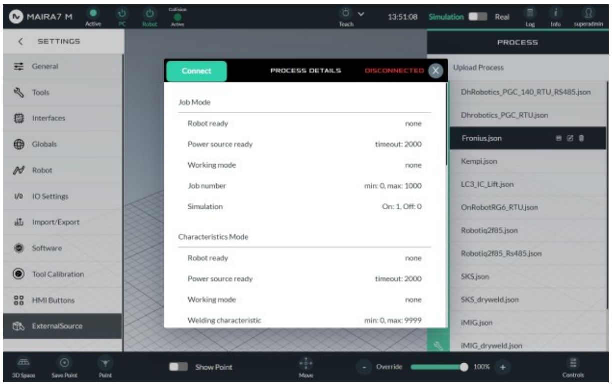
To obtain the “**External Source File Configurator**” and the associated documentation, please contact your NEURA Robotics Business Partner or Project Manager.

To load a configuration file or to create or edit parameter sets from the robot GUI, please follow the instructions below:

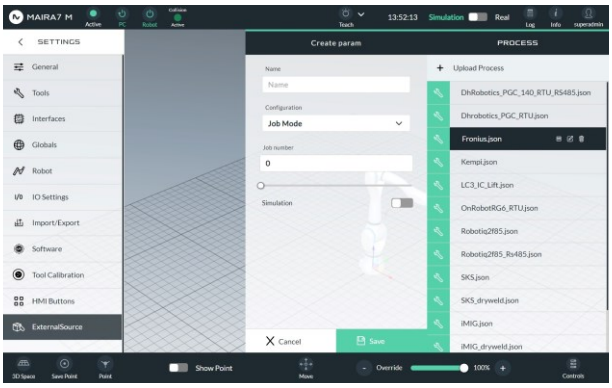
1. You need a **NeuraUSB/SecureSync** flash drive to upload configuration files (see 15.12.1).
2. In the root folder of the **NeuraUSB/SecureSync** flash drive, create a folder called “**process**” if it doesn’t’ already exist.
3. Store your configuration file(s) in the “**process**” folder.
4. Mount the **NeuraUSB/SecureSync** flash drive to the control box of your robot.
5. Go to **Settings** / **External Source** via the main menu.
6. Tap on **Upload Process**, select the configuration file to upload from the dropdown menu and confirm with **Save**.
7. Successful upload is confirmed with a popup message.
8. Click on **Done** to exit the Import dialog or **Go to list** to get back to the dropdown menu to upload additional configuration files.



9. To view your configuration files, select them from the Process list and press the icon. A popup window will show the defined configuration.
10. From the popup window you can also test the connection to the external device by pressing **Connect**. A successful connection is indicated by the popup message “Connection to External Source was successful”.



11. To create predefined parameter sets for a configuration file, you can either use the "**External Source File Configurator**" or the robot GUI.



12. To define parameter sets via GUI, select your configuration file from the process list and click the edit icon .

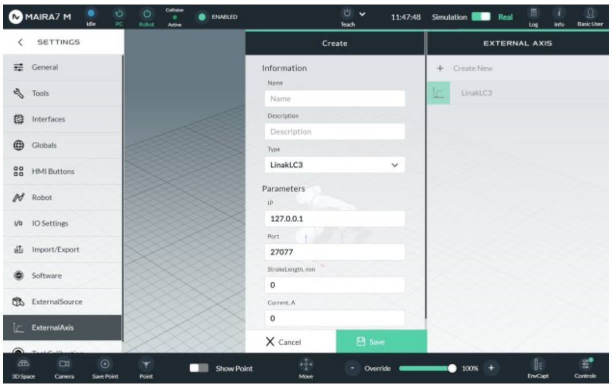
13. Specify a name for your parameter set and define the parameters accordingly. The parameters which can be defined depend on the configuration file selected. Use the "**External Source File Configurator**" to create your configuration files.

14. To use an external source within a program, use the **CustomApp** block from the **TreeEditor** and configure the instruction block accordingly (see 11.7.5).

15.15 ExternalAxis

This settings menu allows for configuration of external axes (currently only Linak vertical lift axes).

1. Press **+ Create New** to create a new External Axis
   a. Or select an existing axis from the list to edit by pressing the **Edit** icon
2. Edit name and add description if wanted.
3. Edit items and save settings.



Item	Settings
Type	Select an axis type from the dropdown menu.
IP	Set the IP address of the linear axis.
Port	Set the communication port of the linear axis.
StrokeLength [mm]	Set the maximum allowed movement distance of the linear axis.

Current [A]	Power setting to accommodate for heavier lifts / faster movement speeds.
Speed [mm/s]	Set the speed for axis movement in mm/s.

15.16 Tool Calibration

In the process of tool calibration, the kinematic tool parameters (TCP offset, TCP orientation) of the attached tool to the robot are estimated to achieve high accuracy on the robot's End-Effector. It is recommended to perform a tool calibration, if the tool parameters are not known, unclear, or the real tool did change. NEURA Robotics provides several methods to identify the tool parameters. In general, no extra device is needed, only a rigid distinctive point within the workspace.

When to perform the Tool Calibration?

- After the real tool with known parameters was modified
- For unclear kinematic tool parameters

How to perform the Tool Calibration? The following methods are available:

- Three-Point Calibration for the TCP-offset
- One-Click-Orientation Calibration for the TCP-orientation
- Advanced-Orientation Calibration for the TCP-orientation
- Automated tool calibration will be available in future software release

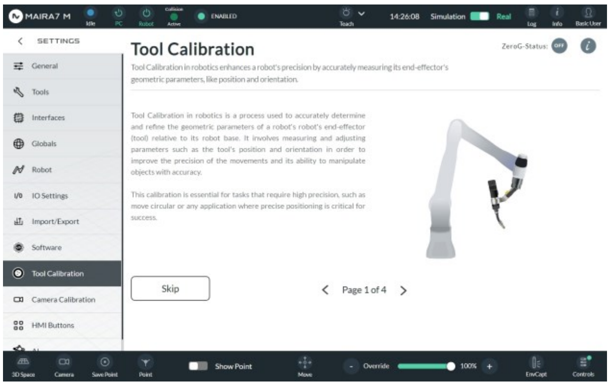
Detailed instructions for each mode can be found in the respective descriptions below.

Useful Tips:

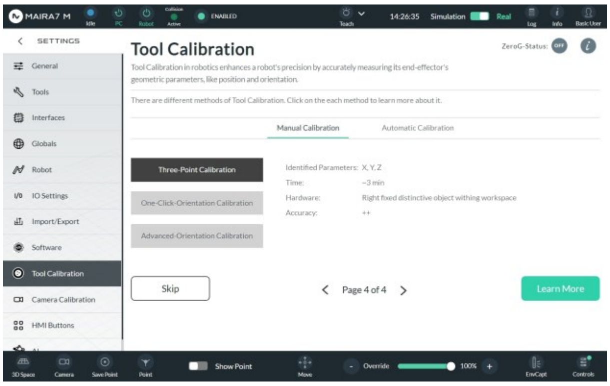
- Approaching the reference point could be done with the freedrive mode.
- To fine adjust the position of the robot with the freedrive mode, slightly clapping onto the robot can change the position.
- Mount the tool securely to the flange. Make sure to not change the static reference point while doing a calibration.

15.16.1 General Process of Tool Calibration

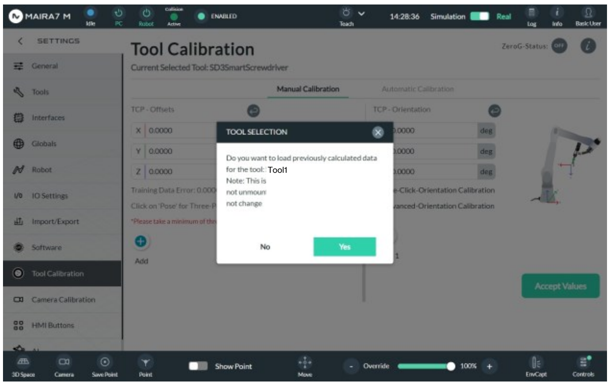
1. Enter the Tool Calibration Wizard via **Settings / Tool Calibration**. A series of onboarding screens will provide crucial background information.



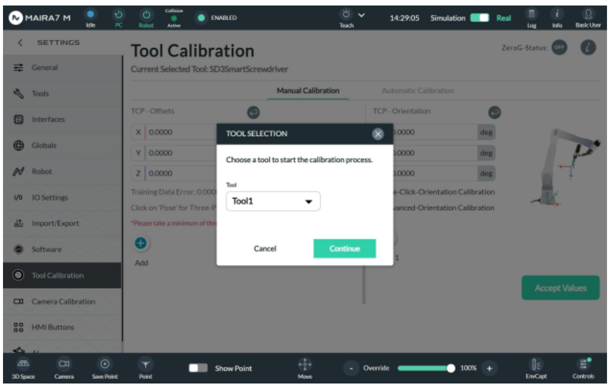
2. In 4 Pages, the general methods are explained:
   - Page 1: General information about tool calibration
   - Page 2: Relation between Flange-frame and TCP-frame
   - Page 3: Guide to calibrate different tools
   - Page 4: Overview about available tool calibration methods
     - Details for each method can be obtained by selecting the method and pressing the **Learn More** button
   - Automatic calibration will be available in a future software release



3. To enter the actual Tool Calibration Screen, press the **Skip** button to close the current onboarding screen.



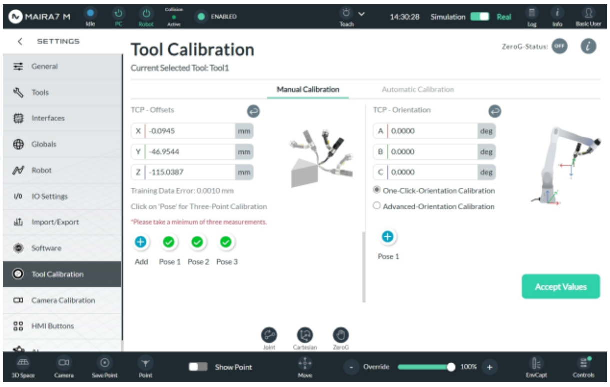
4. A popup dialog will remind the operator that dynamic tool parameters must be set properly before using ZeroG mode to perform Tool Calibration (see 15.2 and 12.1). If dynamic tool parameters have been set properly, please press the **Continue** button to proceed with the Tool Calibration.

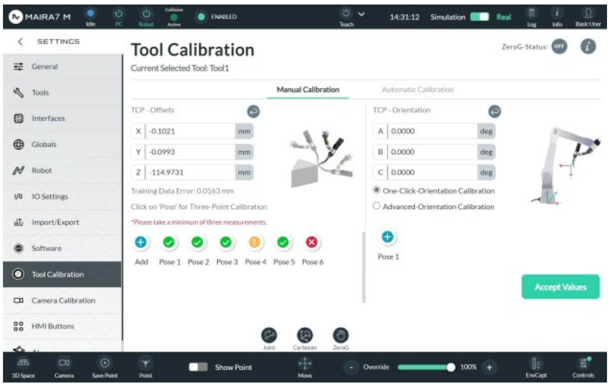


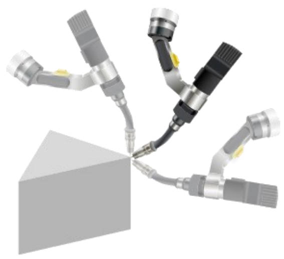
5. In the next dialog, the user can decide to continue a previous calibration by loading the already available tool data by pressing the **Yes** button or to start a new calibration from scratch with pressing the **No** button.

NOTICE	Continuing a previous calibration is only recommended if the tool was not unmounted and the reference position did not change.

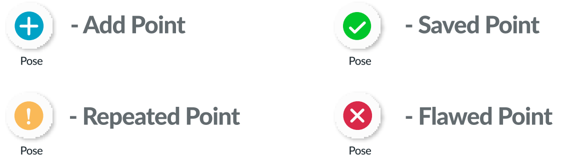
6. If a new calibration is started by pressing the **No** button, the mounted tool should be selected in the dropdown menu. By pressing **Continue**, the main calibration page will be shown.
7. The main page is divided into 2 sections:
    - On the left side, there is the offset calibration section (with the **Three-Point Calibration** method)
    - On the right side there is the section with the orientation calibration methods (**One-Click-** and **Advanced-Orientation Calibration**)
8. **Three-Point Calibration** (for TCP offset)
    - This method identifies the X, Y, Z Offset Parameters from the flange to the TCP
9. Recording calibration data:
    - Move the tool's tip (TCP) to a specific point in the workspace
    - Click the **Add** icon to record a pose
    - Repeat the procedure with at least 3 measurements in different orientations
    - Add more poses for higher accuracy
    - The tool parameters (X, Y, Z) are calculated with each new measurement
    - Each recorded pose is evaluated by its measurement quality, only green ticked measurements are used for calibration
10. Deleting calibration data:
    - Click on a pose to delete it permanently
    - Click on the reset button ↩ to delete the method permanently
11. Useful hints:
    - The validation status of poses might change while recording additional poses, e.g. a pose might initially be tagged invalid and retagged valid while adding additional poses





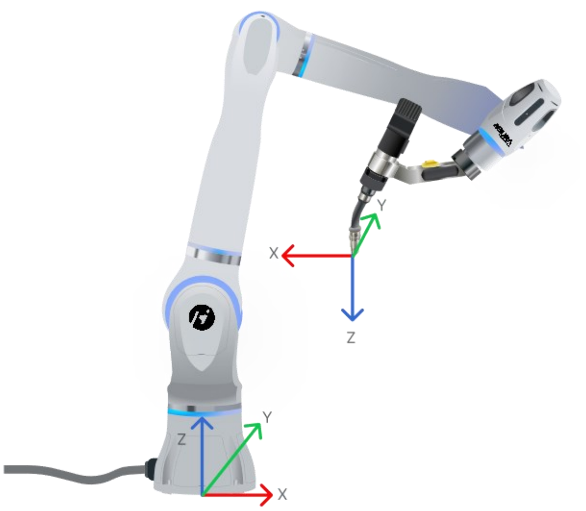


- The displayed training error only displays the quality of the input poses, this is not the real error!



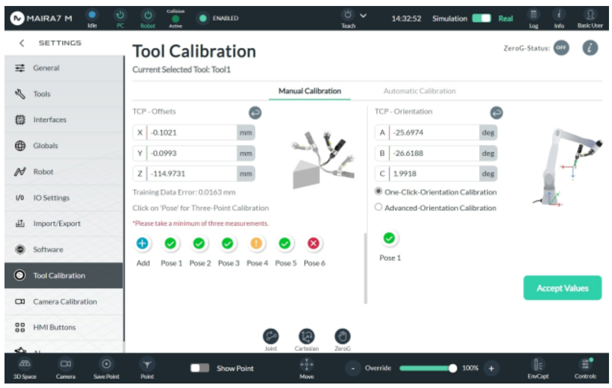
12. One-Click-Orientation Calibration (for TCP orientation)

- This method identifies the A, B, C Orientation Parameters from the flange to the TCP
- The parameters and the result of this method are only visible, if the method is selected, the reset button ↩ only affects the selected method



13. Recording calibration data:

- Move the tool's tip (TCP) to the following specific poses:
    - Align the X-axis of the tool with the negative X-axis of the base
    - Align the Y-axis of the tool with the positive Y-axis of the base
    - Align the Z-axis of the tool with the negative Z-axis of the base
- Click the Add icon to record each pose

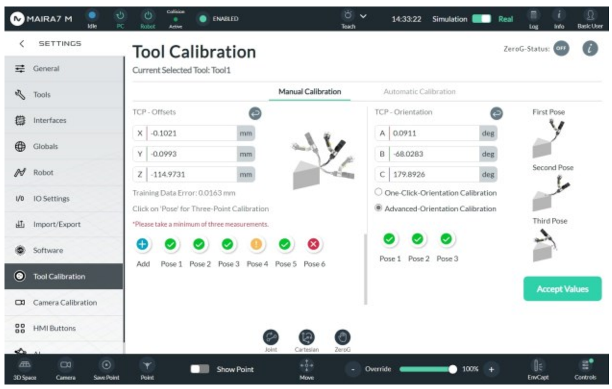


14. Deleting calibration data:

- Click on a pose to delete it permanently
- Click on the reset button ↩ to delete the method permanently

15. Advanced-Orientation Calibration (for TCP orientation)

- This method identifies the A, B, C Orientation Parameters from the flange to the TCP
- The parameters and the result of this method are only visible, if the method is selected, the reset button ↩ only affects the selected method



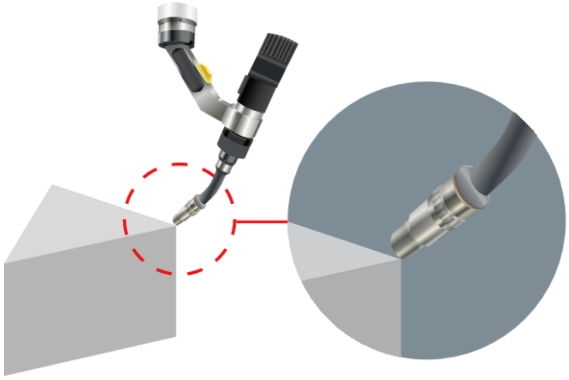
- This method involves guiding the tool around the reference point in three (or at least two) poses
- This method provides a generally higher accuracy than the One-Click-Orientation calibration

16. Recording calibration data:

Move the tool to the following poses and record them with the corresponding buttons:

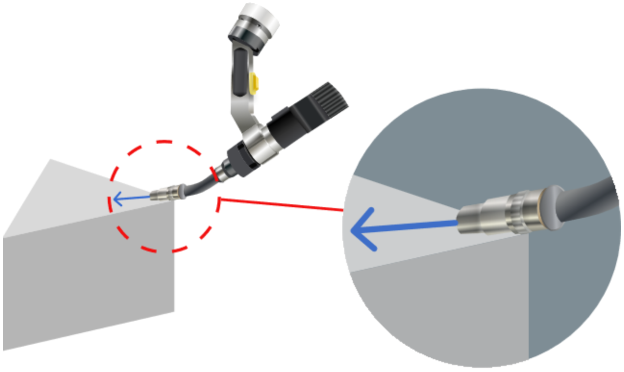
First Pose:

Align the tool with a reference point and an edge representing the Z-axis of the tool and record this pose. It does not need to match the previous calibrated TCP.



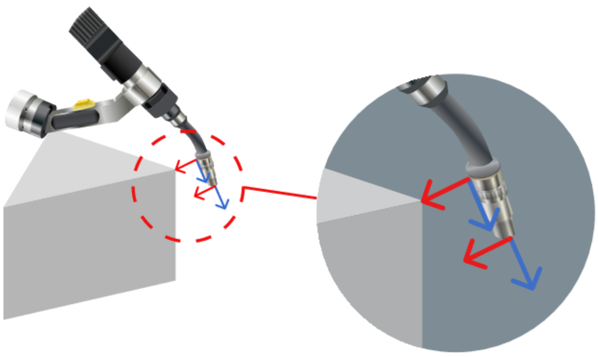
Second Pose:

Align the tool with the reference point along the negative Z-axis direction and record this pose. The parameters A, B and C for the tool orientation are obtained from this measurement.



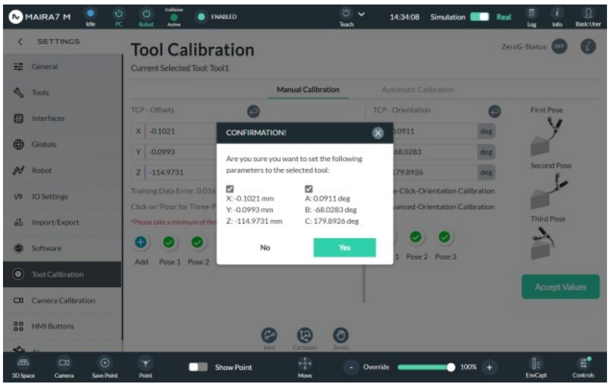
Third Pose (optional):

Align the tool in positive X-direction of the tool from defined Z-axis. This measurement is only needed to define the specific X-and Y-direction of the tool frame (e.g. for non-rotary-symmetrical tools)



17. Accepting parameters:

- To set the parameters, click on **Accept Values**
- Make sure the selected tool (see upper left corner) is the correct one
- Tick the parameters to be permanently set for the tool
- Click **Yes** to set them



NOTICE	The displayed values are cut to 4 digits. The parameters will be set with the highest accuracy though. Check if the parameters were set properly in the tool settings (see 15.2).

15.17 Camera Calibration

Camera calibration should be performed whenever the camera's position relative to the end effector changes. This may occur after a collision or following camera repair or replacement. The calibration process can also be executed, if the actual camera pose seems to be incorrect.

15.17.1 Calibration Board

For camera calibration, a CharuCo calibration board of size 600x400, Pattern Coarse is needed.

Please check CharuCo Targets for more information (https://calib.io/collections/products/products/charuco-targets?variant=9400454938671).

15.17.2 Preparation

► Activate the camera (e.g. wrist) to be calibrated (see 8.7)

► Place the CharuCo calibration board on a table

  ○ Ensure the calibration board is clearly visible by the camera to be calibrated

► Ensure the workspace is clean and collision-free

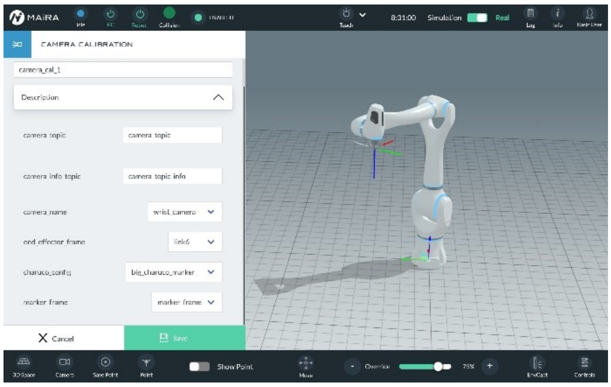
15.17.3 Calibration procedure

NOTICE	Do not move the calibration board once the calibration process has been started. Otherwise, calibration will be impaired or fail.

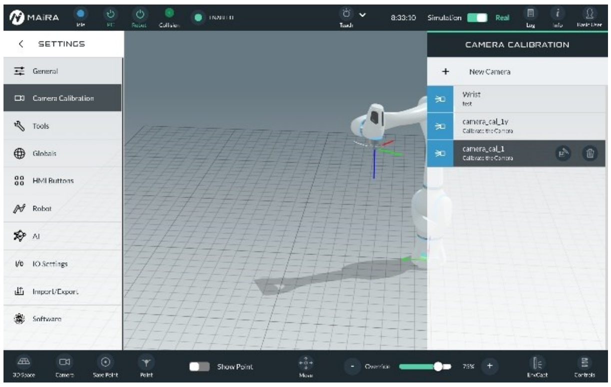
🗐 Preparation steps (see 15.17.2) must have been completed

🗐 The robot must be powered on (**ENABLED**)

- ▶ Go to **Settings >> Camera Calibration** (enter password if required)
- ▶ Set the **end_effector_frame** to **link6** or **link7** according to your camera configuration
- ▶ For **link6**, set the **camera_name** to **wrist_camera**
- ▶ For **link7**, choose **dentist_camera**
- ▶ Leave the remaining parameters unaltered
- ▶ Press **Save**

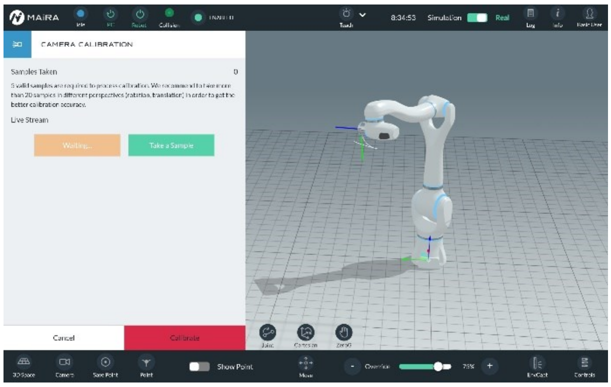


- ▶ Select **camera_cal_1** from the **CAMERA CALIBRATION** list in the right-hand menu pane
  - o If you altered the **Camera Name** parameter, select the altered name respectively
- ▶ Press the **Edit** button
- ▶ Scroll down to **New Calibration** and press **Start**
- ▶ Press **Run**



- ▶ Take samples of the calibration board in different orientations
  - o Use ZeroG to move the robot and change camera perspectives
  - o Press the **Take a Sample** button to capture a sample image

NOTICE	Take at least 5 samples. The more samples, the more accurate the calibrated pose of the camera. 20 samples from different perspectives (rotation, translation) are recommended.



🗐 You have taken at least 5 (recommended 20) sample pictures of the calibration board

▶ Press **Calibrate** to run the actual calibration with the acquired samples

➔ Calibration results will be shown after completion

▶ Press **Finish** and confirm with **Yes** to exit camera calibration

➔ Camera calibration has been performed

16 VOICE CONTROL

The Voice Control system allows users to control MAiRA by their voice without having to physically interact with any device. The Voice Control system consists of two components.

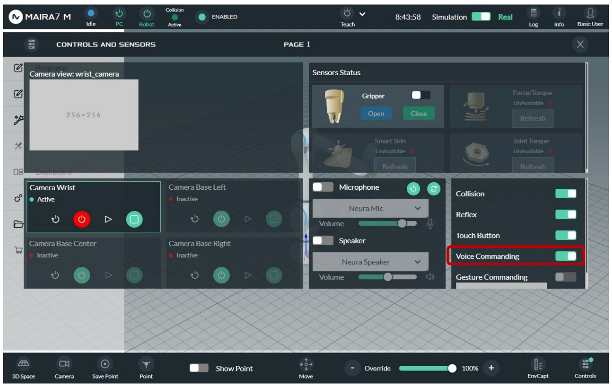
1. **Voice Activation**
   - To invoke the voice engine by saying "**Hey MAiRA!**".
   - Activation will be confirmed with a feedback sound and the question "**Hi, how can I help you?**".
2. **Voice Commanding**
   - To control the activated robot by voice, e.g. by saying the command: "**Move axis five thirty degrees.**".
   - Predefined commands in multiple languages can be found in 16.2.1.

16.1 Enabling voice control

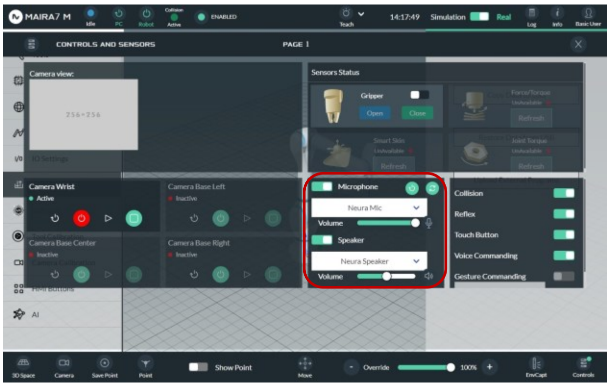
Voice control can be enabled via the **Controls** button in the bottom menu bar.

- ► Use the slider to activate **Voice Commanding**
  - If the **Voice Commanding** slider is grayed out (not available), please wait for two minutes. This usually happens when the robot has just started and all available functionalities are checked.

➔ Once activated, you will hear the "**Voice Commanding is activated**" feedback



- ► By default, the integrated microphones and speaker are selected
- ► To select other input/output devices, activate the slider for microphone/speaker and select the appropriate devices from the dropdown menu



16.2 Voice commands

Voice Control uses pre-defined commands to execute defined actions.

16.2.1 List of commands

A list of all the currently available commands is given below.

English	Chinese	German	Japanese
Move Linear	線性移動	Bewege linear	ジョイントを移動
Move Joint	移動關節	Bewege Gelenk / Gelenk bewegen	直線移動
Free Drive Mode	無重力模式	freier Bewegungsmodus	ダイレクトティーチングモード
Pick	抓取	Greife ein Objekt	取ってください
Open Gripper	打開抓手	Greifer öffnen	グ リッパーを開いて
Close Gripper	關閉抓手	Greifer schließen	グ リッパーを閉じて
tic-tac-toe	井字遊戲	tic-tac-toe	三目並べで遊びましょう
Scan	掃描	Scannen	スキャン
Play Trajectory	重複上一個動作	Gebe Bewegungsaufnahme wieder	動作を再生
Record Action	紀錄動作	Aktion aufnehmen	動作を保存
Change Language	切換語言	Sprache ändern	言語を変更する
Did you eat	你吃了嗎	Hast du gegessen	ご飯食べました
How are you	你好嗎	Wie geht es dir	お元気ですか
MAiRA meaning	麥拉的含意	Was bedeutet Maira	マイラとは
I love you	我喜歡你	Ich liebe dich	好きです
Tell a joke	講個笑話	Erzähle einen Witz	冗談を言う

16.2.2 Staged Commands

All the commands listed in 16.2.1 are considered staged commands. To complete a command, users must follow the robot's instructions to specify the required parameters.

- Example 1: "**Hey MAiRA!**" → "**Move joint**" → "**Point one**"

- Example 2: “**Hey MAiRA!**” → “**Pick**” → “**Any object**”

16.2.3 Chained Commands

When we combine a command with its parameters in one go, it is considered a chained command. The user can specify a complete command in one sentence.

- Example 1: “**Hey MAiRA!**” → “**Move axis one by thirty-five degree.**”
- Example 2: “**Hey MAiRA!**” → “**Pick any object.**”

16.2.4 Double Check Operation safety feature

To avoid unintended actions, Double Check Operation (DCO) repeats the commands spoken by the user to verify the integrity of the command.

- Users will hear the feedback: “**Do you mean...**” after each command is spoken.
- The user must respond with “**yes** or **no**”.
    - **Yes**: continues with the next operation, e.g. requesting further parameters or executing the command.
    - **No**: reverts to the previous operation (robot will ask previous question).

DOC is an important safety feature, especially for commands that will move the robot.

16.3 Settings

The voice control settings can be found in the main menu via **Settings >> AI >> Voice**. Please refer to 15.10.1 for details.

17 FILEMANAGER MENU

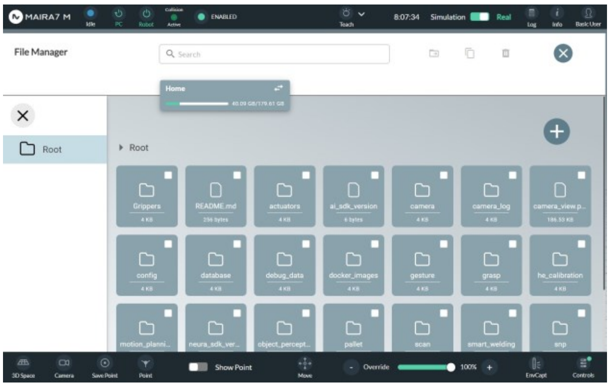
The robot's GUI provides a basic file manager for file manipulation. It allows for standard file operations like moving, copying, deleting or adding files and folders.

To access the file manager:

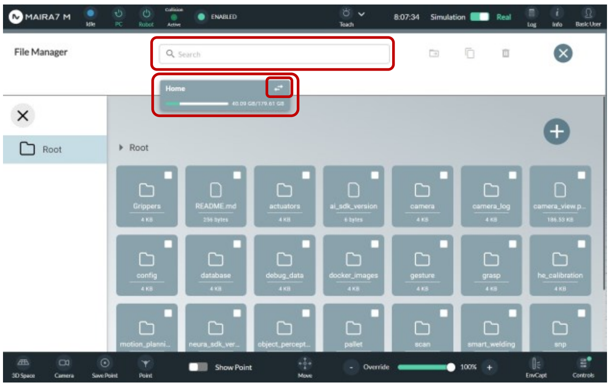
- ► Press on **Filemanager** in the main menu
- ► Enter a password if required

➔ The file manager is open and ready to perform file operations

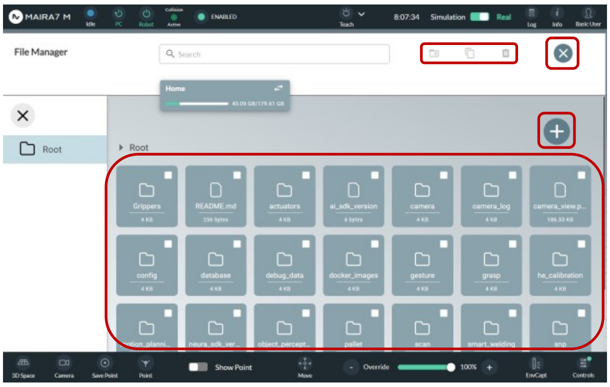
The left menu pane allows for folder navigation.
The main menu pane shows subfolders and files.



- ► Use the search bar to search for files/folders
- ► The box underneath the search bar indicates the current storage locations and indicates memory usage (Default is the **Home** folder of the AI controller PC)
- ► Press on the **2 small opposing arrows** to select a different storage location



- ► Press the **+** icon to add a file/folder
- ► Press on a file/folder icon (or multiple) from the list to select it
- ► Press the **Move** (*folder with arrow*), **Copy** (*2 files*), or **Delete** (*waste bin*) icon to perform the corresponding action with the selected files/folders
- ► Press the **X** button in the top right corner to exit the file manager



➔ The file manager is closed and you are back in the main GUI view

© NEURA Robotics GmbH
Gutenbergstraße 44
72555 Metzingen
Germany

Phone: +49 (0) 7123 87970 0
E-Mail: info@neura-robotics.com
www.neura-robotics.com

Document title:	MAiRA Software Manual
Document edition:	v2.1/ 12-2024
Document number:	100001000014
Document language:	English

COPYRIGHT, CHANGES AND LIABILITY