

---

# **Mech-Mind Quick Guide**

**Mech-Mind**

**Dec 23, 2022**

## CONTENTS

1	Obtain the Camera and Accessories	2
2	Mount the Camera	4
3	Connect the Whole System	7
4	Installation and Connection	10
5	Your First Pick	15



This chapter will guide you to create an application project—box palletizing. You will start from camera installation and be able to control a real robot to pick boxes in the end.

## **OBTAIN THE CAMERA AND ACCESSORIES**

Mech-Mind Robotics provides basic camera accessories for our customers.

### **Standard Hardware**

Please check if all the devices and accessories are included in the box. All the devices and accessories you need are shown below.



Camera × 1



Industrial PC × 1



Dongle × 1



Flange × 1



Calibration Board × 1



Camera Power Cord × 1



Network Cable × 2



HDMI Cable\* × 1



- Dongle: used to decrypt Mech-Mind Software Suite
- Calibration Board and Flange: used to calibrate the camera
- Power Cord: used to connect the camera to the electricity supply
- Network Cable: one is used to connect the camera and IPC; and the other is used to connect the robot controller and IPC
- HDMI Cable: used to connect the computer monitor and IPC

## MOUNT THE CAMERA

Mounting the camera is like adding a pair of eyes to the robot.

The common mounting methods of the camera include:

- Eye To Hand (ETH): the camera is mounted on a stationary stand independent from the robot.
- Eye In Hand (EIH): the camera is mounted on the flange located at the end of the robot.

In this example project, since there is sufficient space and the objects to be picked are placed in a limited region, the camera will be mounted by using the ETH method. By using the ETH method, the camera needs to be attached to a fixed stand at a certain height above the object to be photographed, and the camera will not move with the robot.

Please mount the camera using ETH method as shown below.



#### Things you need to pay attention while mounting

1. Location selection: please make sure that the camera is mounted directly above the object plane, and there is enough space for the robot to perform a task without colliding with the camera, as shown below.

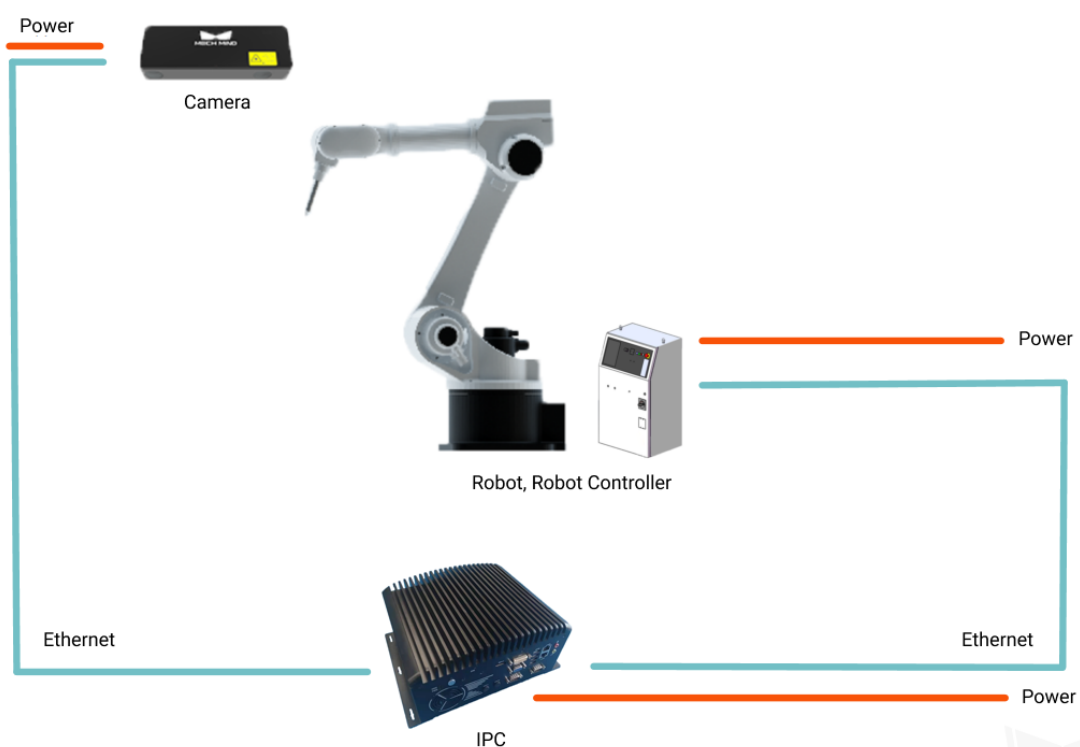


2. Camera stand: you can choose to use an appropriate camera stand based on the actual situation on site.
3. Camera Installation: please make sure that the motion of the robot and the vibration caused by other equipment will not affect the position of the camera. Please refer to camera\_mounting\_dimensions to learn more about fixation dimensions of all camera models.



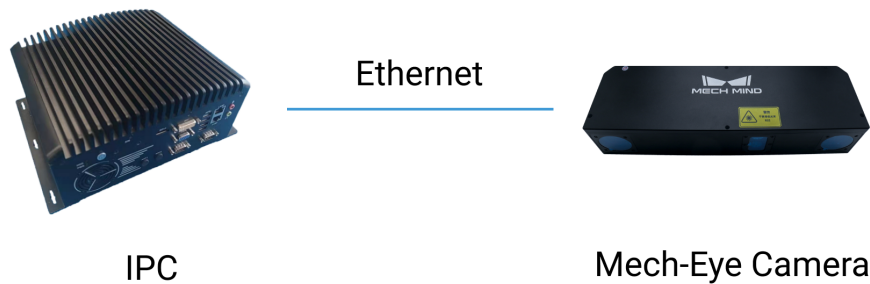
## CONNECT THE WHOLE SYSTEM

After mounting the camera and setting up the robot, you need to connect them to the IPC.



### 3.1 Connect the Camera and the IPC

Use the power cable and Ethernet cables provided by Mech-Mind Robotics to connect the **Camera** and **IPC**. An IPC may have multiple network interfaces, you can choose either one to connect. After connecting, you can change the subnetwork of the corresponding network interface according to actual needs.



## 3.2 Connect the Robot and the IPC

Use a Ethernet cable to connect the **Robot Controller** and **IPC** . After completing the connection, please turn on the robot controller and the robot teach pendant.

### Attention:

1. Please pay attention to which Ethernet port on the robot controller should be used. See `robot_integrations` for detailed information.
2. Please keep the cables organized. It is dangerous if the cables get tangled up with the robot.

## 3.3 IP address Setup and Program Loading

- In this project, the IPC should be connected with both the robot and camera, and therefore two IP addresses are needed. You can select the network interface and subnetwork according to actual needs. Please make sure that the **IP address of connected devices** and the **IP address of IPC** belong to the same subnetwork.
  1. In this project, the IP address of the network interface that connects robot to the IPC is 192.168.2.222.
  2. The IP address of the network interface that connects camera to the IPC is 192.168.2.66.
- To allow Mech-Mind software to communicate with the real robot, please follow the instructions of the robot brand you are using in `robot_integrations` to complete the relevant setup.

Now you have completed connecting the whole hardware system. Please check the circuit and power on. Then please read on to configure the software on the IPC.

## INSTALLATION AND CONNECTION



In order to enable the software to control the robot successfully, please operate according to the following steps.

1. *Select Robot Model*
2. *Connect and Test the Camera*

### 4.1 Select Robot Model

In order to connect to a real robot, you need to create a new project in Mech-Viz and add a robot model that corresponds with the real robot.

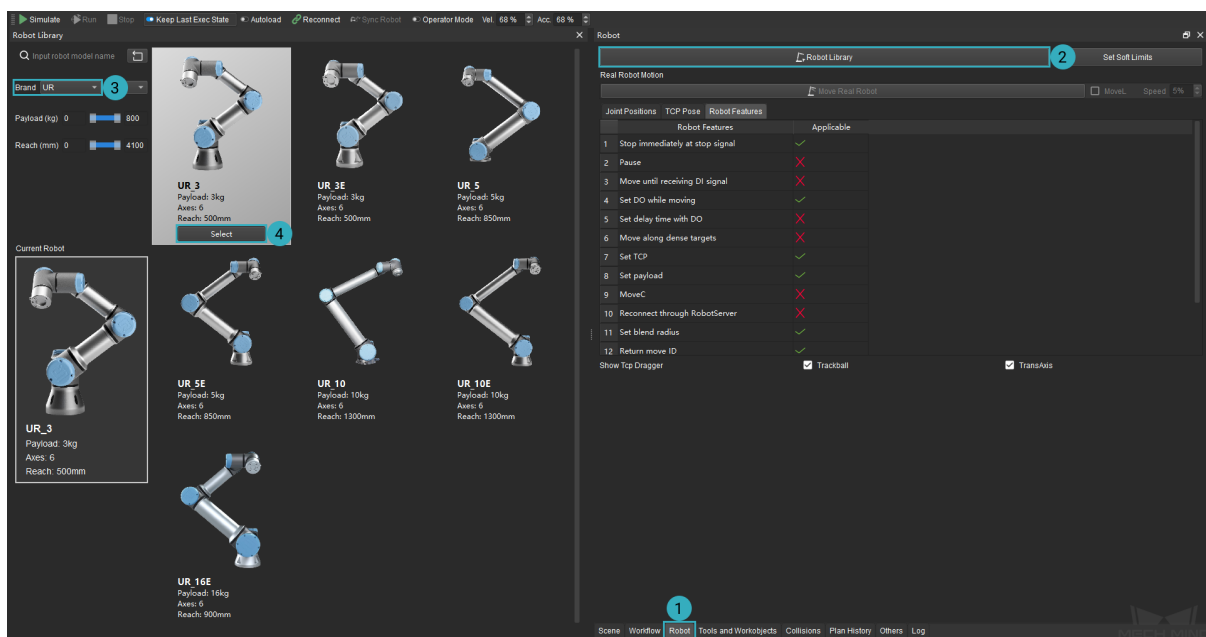
Mech-Viz is a cutting-edge **robot intelligent programming software** that has a visualized and coding-free programming workspace and can make a simulation with one click. It has built-in intelligent algorithms such as path planning, collision detection, etc., and has an adapted robot library including models of many major brands.

**Attention:** Starting from version 1.6.0, only robot models from the widely used brands are preloaded in Mech-Viz. All other robot models must be manually imported.

Please refer to [Robot Model Package](#) for instructions on downloading and importing robot model packages.

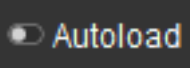
The example Mech-Viz project uses a UR\_3 robot. In a real application, please select the robot that you actually use on-site.

1. After downloading and importing the robot model package, click on the **Robot** tab in the lower right.
2. Click on *Robot Library* at the top of the tab.
3. On the left side, select the robot brand from the drop-down menu of **Brand**, or use the search box above it.
4. Select the robot model.



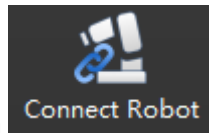
5. In the menu bar, click on *File* → *Save Project* or use the shortcut **Ctrl + S** to save the changes to the project.

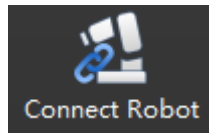
**Attention:** While building and configuring the project, remember to save the changes now and then!

6. Click on  in the toolbar.

**Note:** The project must be set to autoload for it to be registered in Mech-Center.

7. To control the real robot, complete the following in Mech-Center:



1. Open Mech-Center, and click on .
2. The robot is successfully connected if:
  - A message saying Robot: server connected to the robot shows up in the Log panel, and



-  shows up in the Service Status panel.

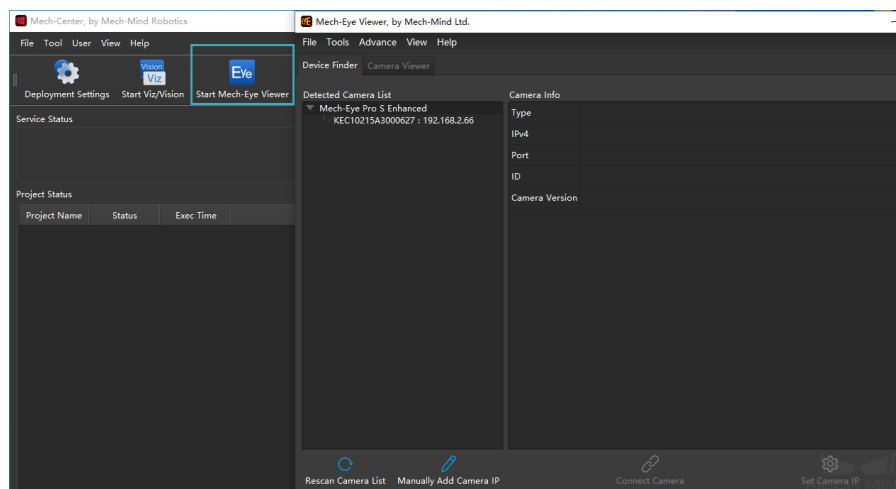
## 4.2 Connect and Test the Camera

This section will show you how to connect a camera via Mech-Eye Viewer and test if the camera can capture images correctly.

Mech-Eye Viewer is a camera configuration software independently developed by Mech-Mind Robotics. You can use the software to connect to the camera and capture images.

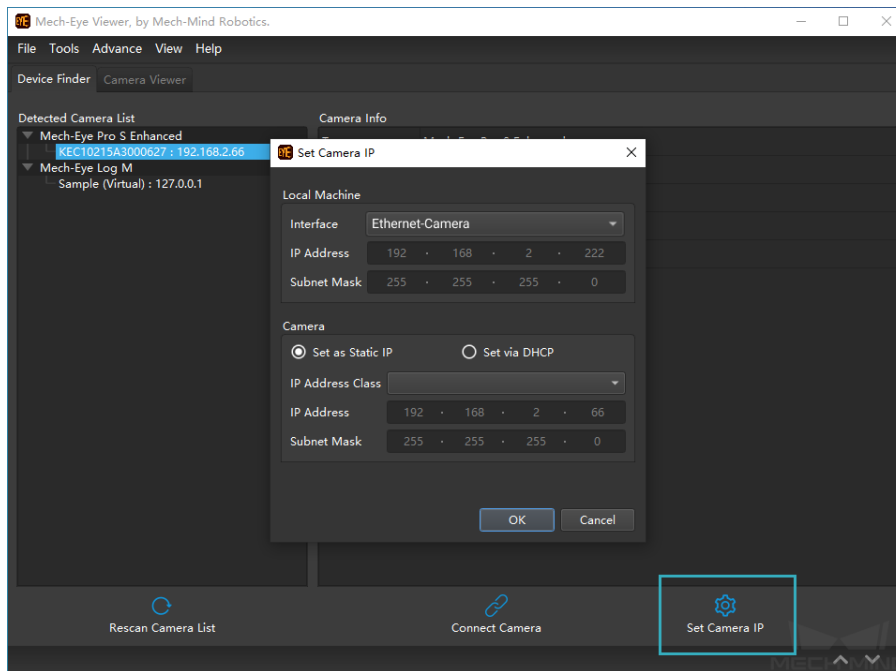
Please follow the steps below to connect to the camera:

1. Click on *Start Mech-Eye Viewer* in Mech-Center.

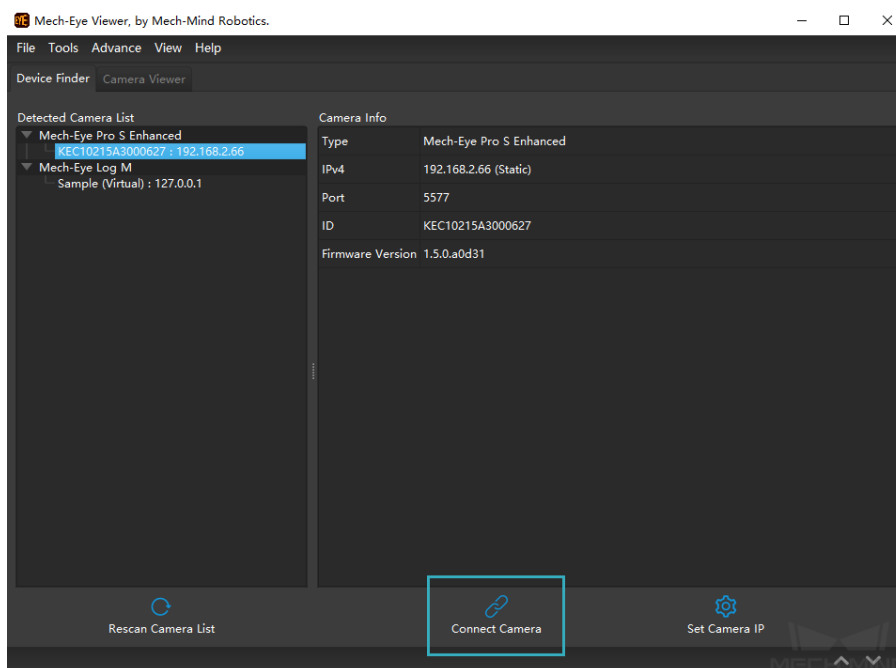


2. Set a static IP address for the camera.

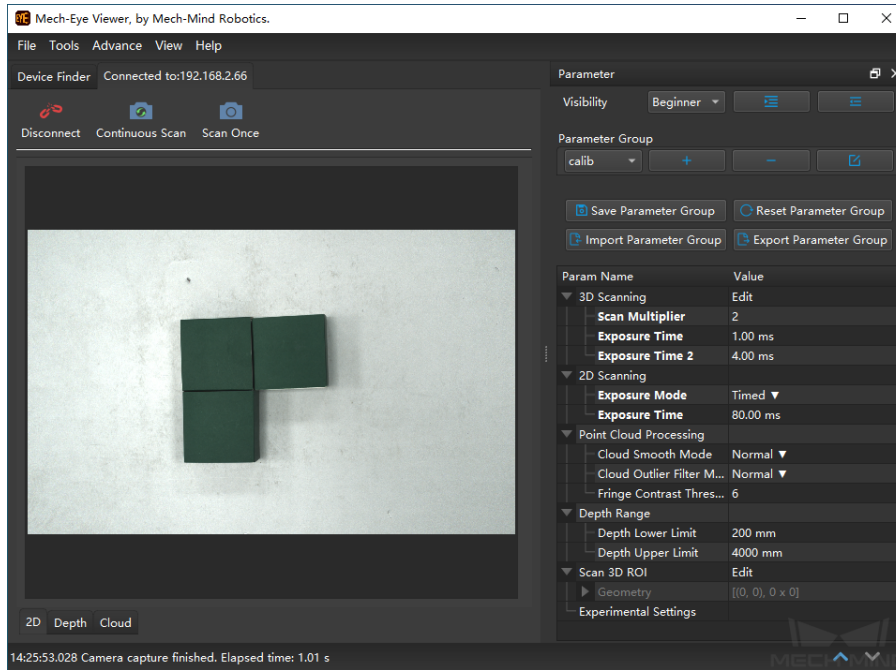
After selecting the camera to be connected (please do not select the virtual camera with a Virtual tag), click on *Set Camera IP* and select **Set as Static IP** in the pop-up window. Since the IP address of the IPC is 192.168.2.222, set the IP address of the camera as 192.168.2.66. Click on *OK* to save the IP address.



3. You can either double-click the camera IP on the left or click on the camera IP and then click on *Connect Camera* in the bottom center to connect the camera.



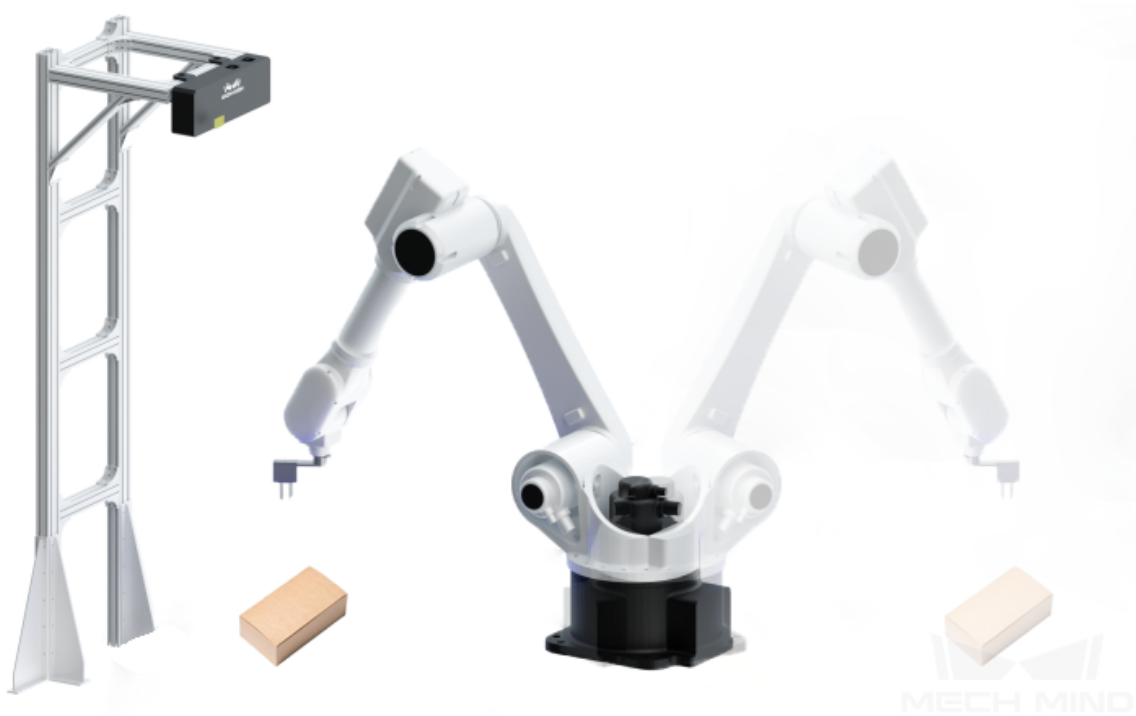
4. After connecting successfully, you will enter the camera viewer interface.  
Click on *Continuous Scan* to keep capturing images , and click *Scan Once* to capture one image at a time.



If the camera can capture images correctly, click on *Stop*. Now you have successfully connected and tested the camera. Please read the next section.



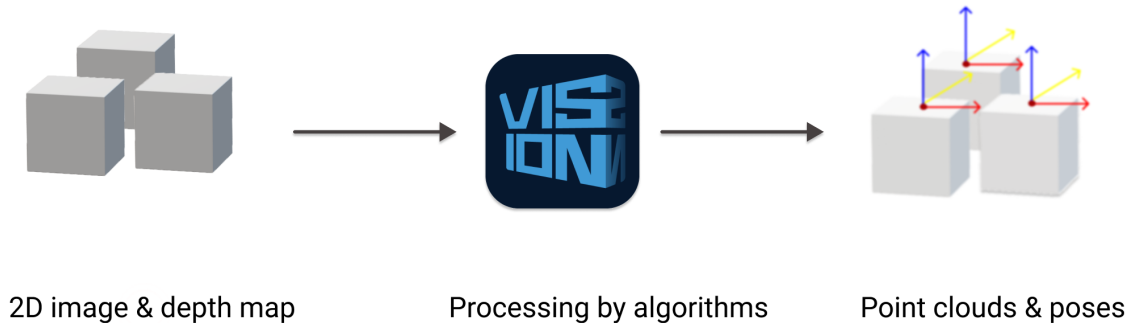
## YOUR FIRST PICK



Before reading this section, please make sure you have completed the configurations in the previous sections of this chapter.

In this section, you will use the Mech-Mind Software Suite to obtain the pose of a box, simulate the robot's trajectory, and let the physical robot complete picking and placing of the box.

## 5.1 Use Mech-Vision to Obtain the Box Pose



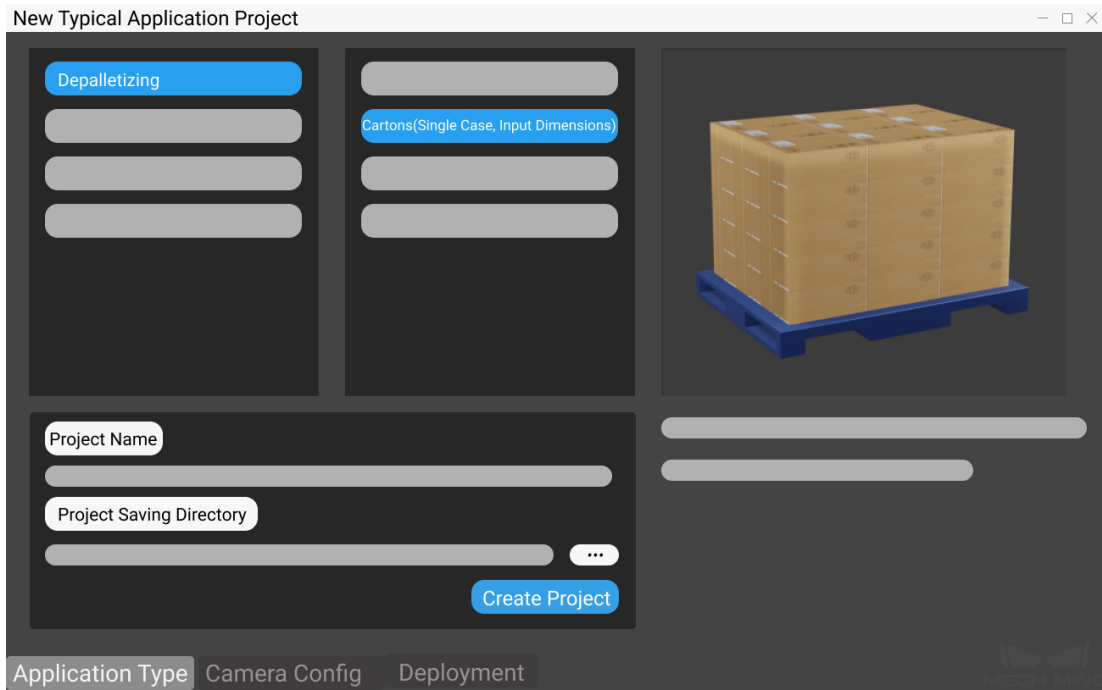
In this section, you will learn how to use Mech-Vision to obtain the pose of a box.

### Preparation

1. Get two boxes of the same dimensions and record the dimensions (length, width, and height).
2. Get a calibration board provided by Mech-Mind Robotics (please make sure the dots on the calibration board are clear and free of obvious scratches, and the calibration board does not have deformation).

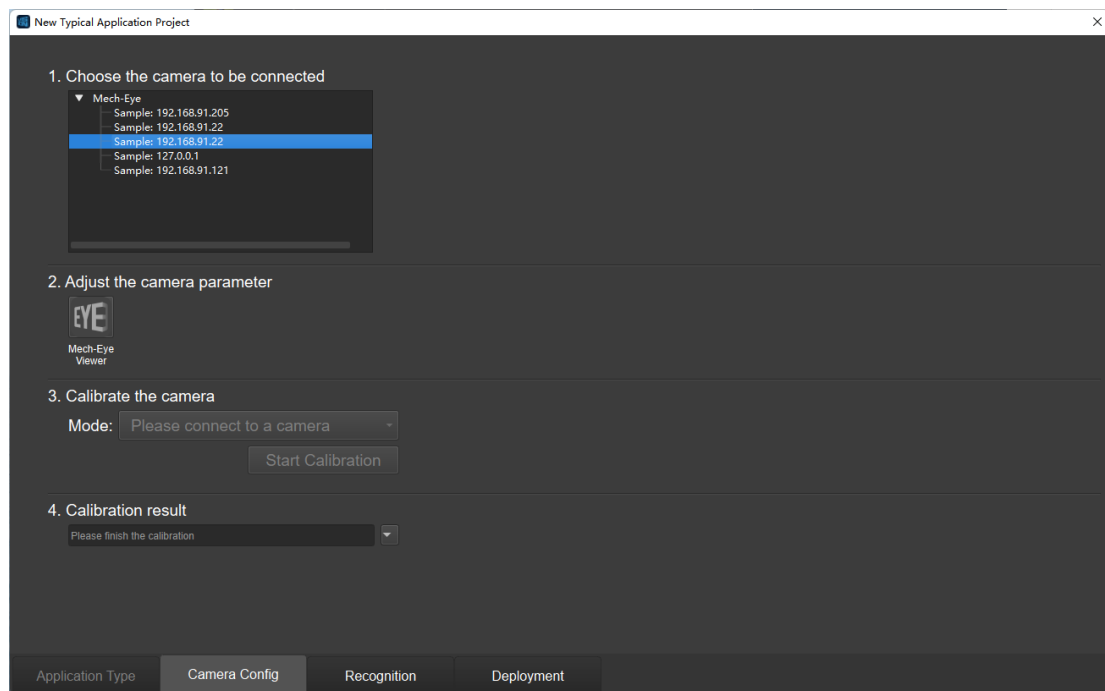
1. Create a new project via New Typical Application Project.

Open Mech-Vision, and click on *Typical Applications* → *New Typical Application Project* in the menubar on the top. After the window *New Typical Application Project* pops up, select *Depalletizing* → *Boxes (Single Case)*, and set the project name and file location to finish creating the project.



2. Complete camera configuration and calibration.

Please follow the instructions on the page to complete the following steps:



1. Select the camera to connect:

Double-click the camera to be connected in the camera list to create a connection with

the real camera. After the connection is established, the interface of the remaining steps will be activated.

2. Adjust camera parameters:

Select the default parameter group built into the Mech-Eye Viewer.

3. Calibrate the camera:

**First mount the calibration board to the end of the robot arm.**

Click on *Start calibration* and follow the step-by-step instructions in *calibration\_guide* to complete the calibration.

**After the calibration is completed, remove the calibration plate and mount the end effector.**

4. Calibration result:

Load the calibration results from the previous step to finish all the steps on the camera configuration page.

Click *Recognition* at the bottom of the interface to go to the next step.

3. Set the parameters related to object recognition.

Please place the carton in the center of the camera's field of view in a random orientation before proceeding, as shown in the figure below.



**Attention:** Please Make sure that only the target object to be recognized is in the camera's field of view, which you can confirm by capturing images in Mech-Eye.

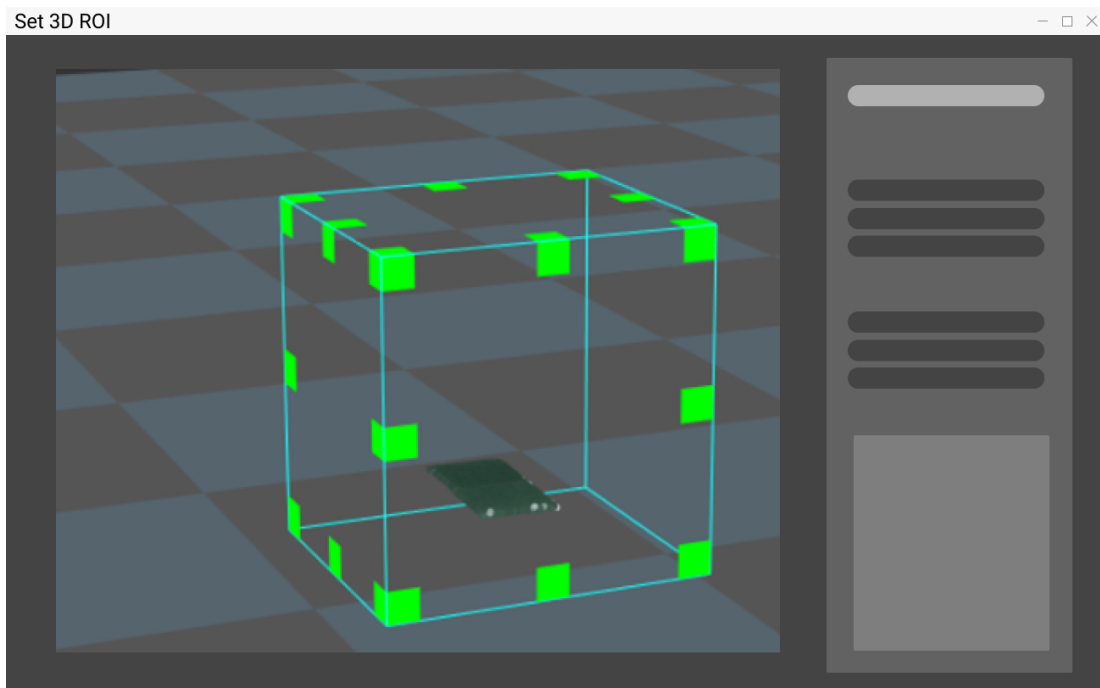
1. Box dimensions:

Fill in the box dimensions in the field of box dimensions on the right side of the interface. The model path and configuration file path are preset by the software, and you do not need to change them.

2. 3D ROI:

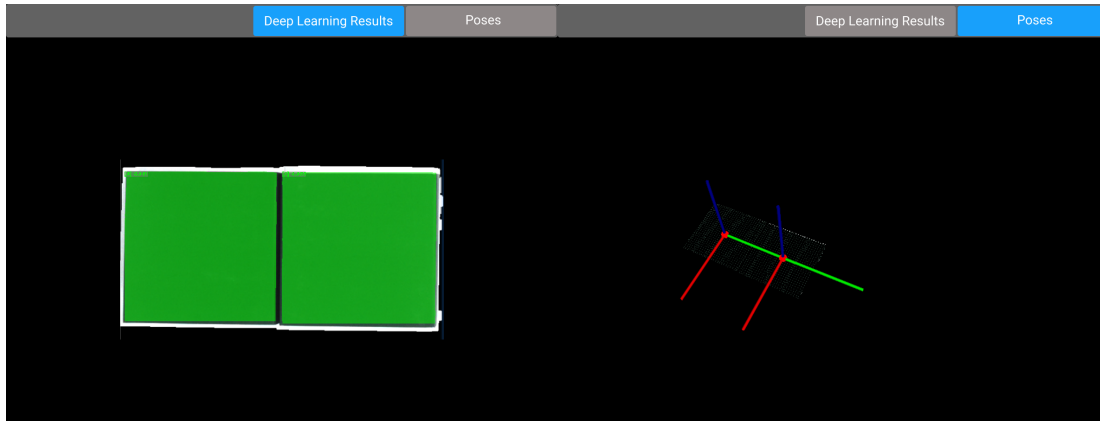
Click on *3D ROI* in the upper left corner of the window to open the window of setting the ROI (region of interest) and complete the ROI settings. Press Ctrl and left-click on the little green squares on the edge of the cube that represents the ROI to adjust the ROI.

**Attention:** The cube representing the ROI should contain all the objects to recognize. The cube walls should not go through the object point clouds.



3. Deep learning results & poses:

After you click on *Generate pose*, the project starts running and outputs the deep learning results and object poses, as shown below.



If there are X, Y, and Z coordinates marked on a box, the box has its pose output from the project.

When going through this step, if the two boxes are not segmented in the deep learning results or no pose is output, please try the following measures to solve the issue.

- Confirm whether the box dimensions filled in are the same as the actual dimensions.
- Re-set the 3D ROI.
- Adjust the parameters of the camera.

Click on **Deploy** at the bottom of the interface to proceed to the next step.

4. Complete the deployment and view the final running result.

At this point, you have completed all the settings of the project. Please click on *Finish* to finish deploying the project and enter the custom editing mode.

You have done everything needed in Mech-Vision, please proceed to the next section.

## 5.2 Use Mech-Viz to Plan a Motion Trajectory for the Robot



In this section, you will learn how to use Mech-Viz to plan robot's trajectory. Please open Mech-Viz.

### 5.2.1 Configure Robot Tools and Workobjects

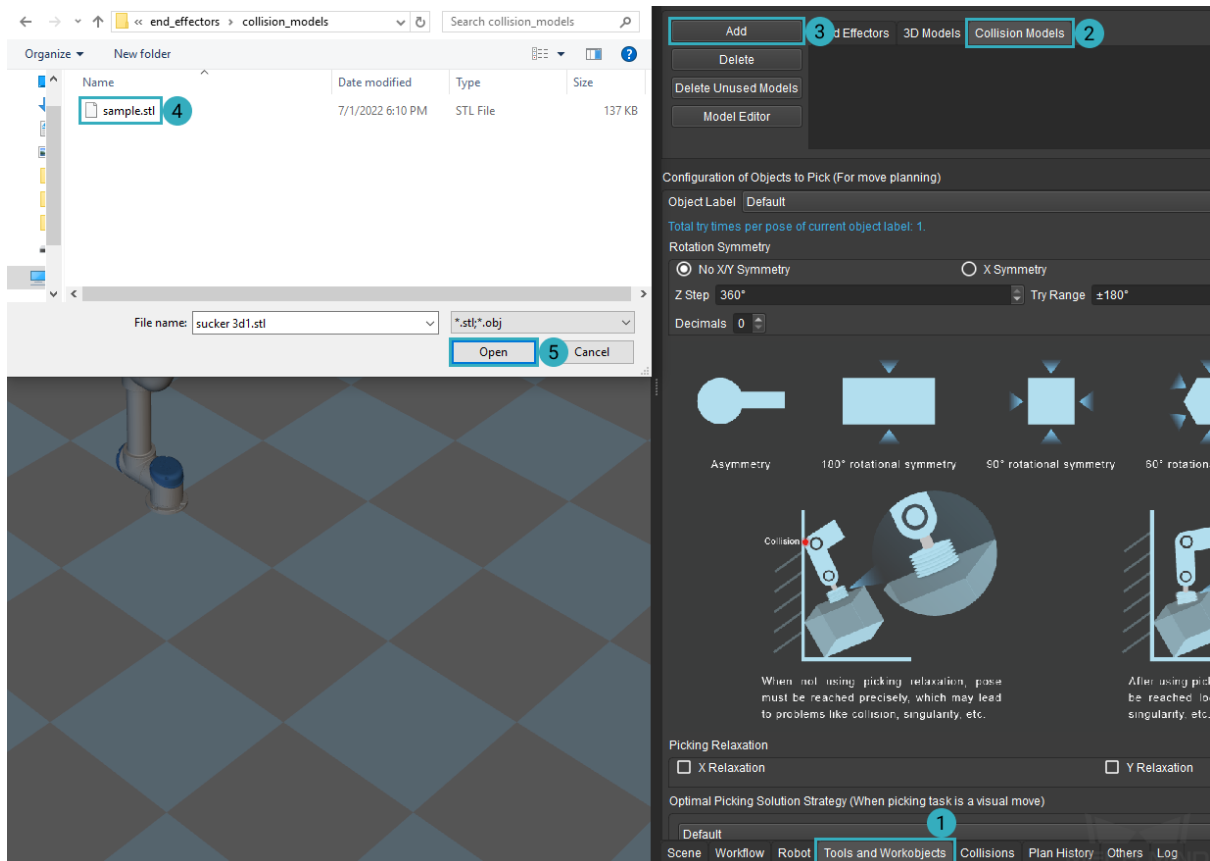
Once you finish the configurations in this chapter, you will be able to:

- Have the robot tool displayed in the 3D simulation area.
- Add robot tools to collision detection.
- Allow Mech-Viz to plan the path and picking strategy for the robot.

### Add Collision Model

The robot tool model file used in the example project is located in `\sampleviz\end_effectors\collision_models`.

1. Click on the **Tools and Workobjects** tab in the lower right.
2. Click on the **Collision Models** tab at the top.
3. Click on **Add** on the left.
4. Select the **sample** STL file from the above location.
5. Click on **Open**.

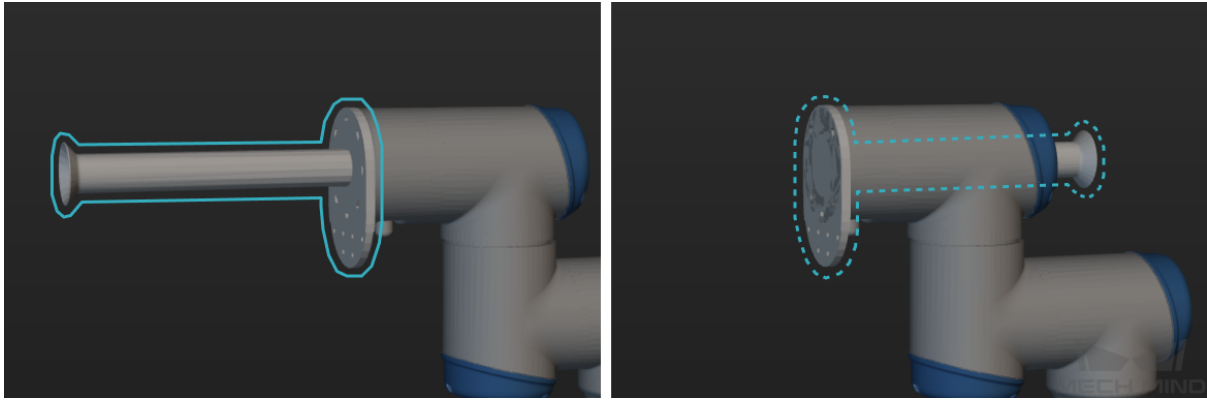


### Adjust Model Position

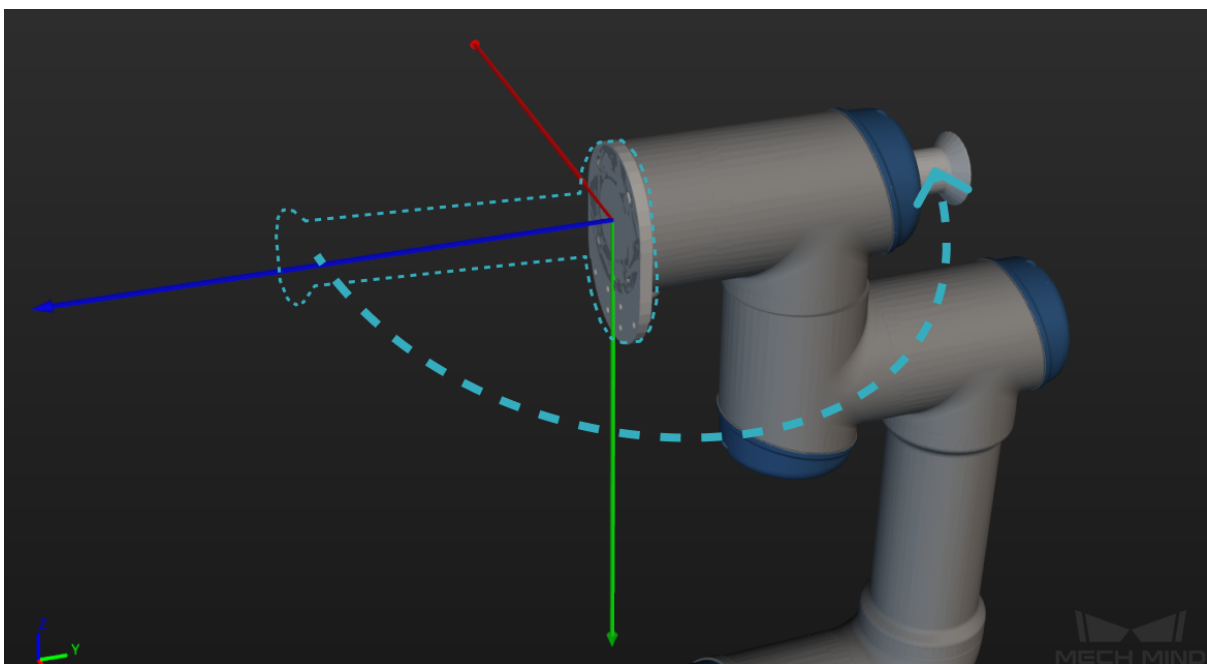
The added tool model may not be in the correct position and thus requires adjustment.

In the figures below, the left one is a correctly positioned tool model, but the model that you just added probably looks like the one on the right.



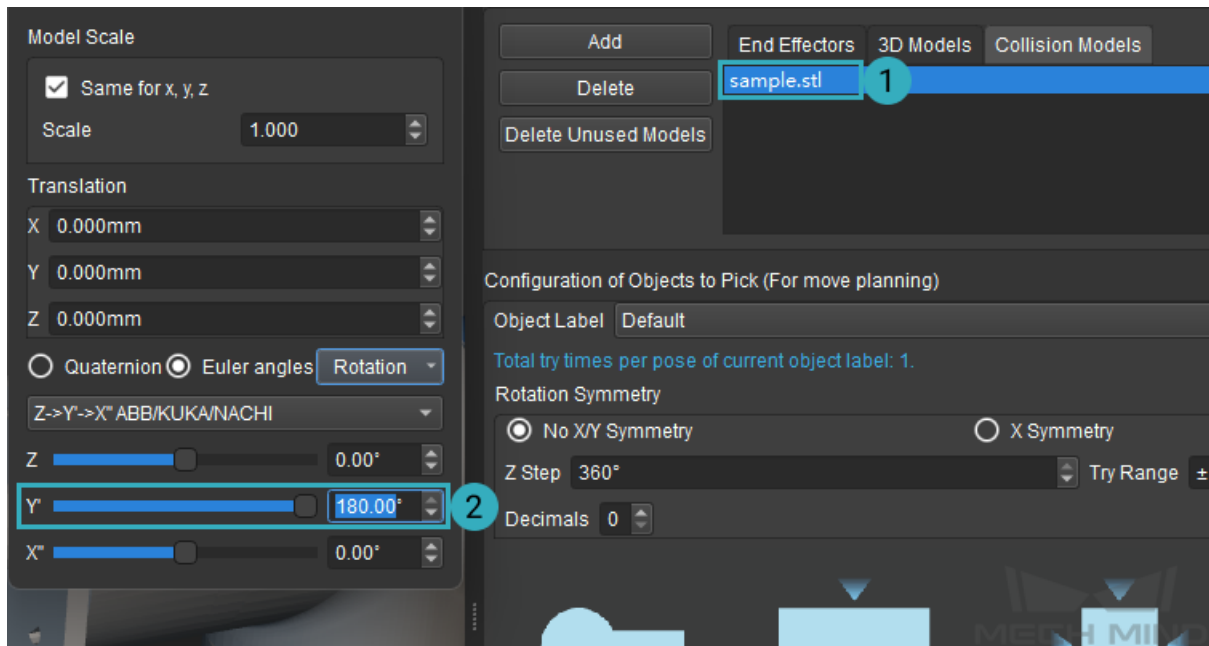


With the axes displayed, you can see that the tool model is flipped about its Y-axis (green).



So in this case, all you need to do is flip it around.

1. Double-click on the model name in the **Collision Models** tab.
2. In the pop-up window, change the value of **Y'** to 180°.




---

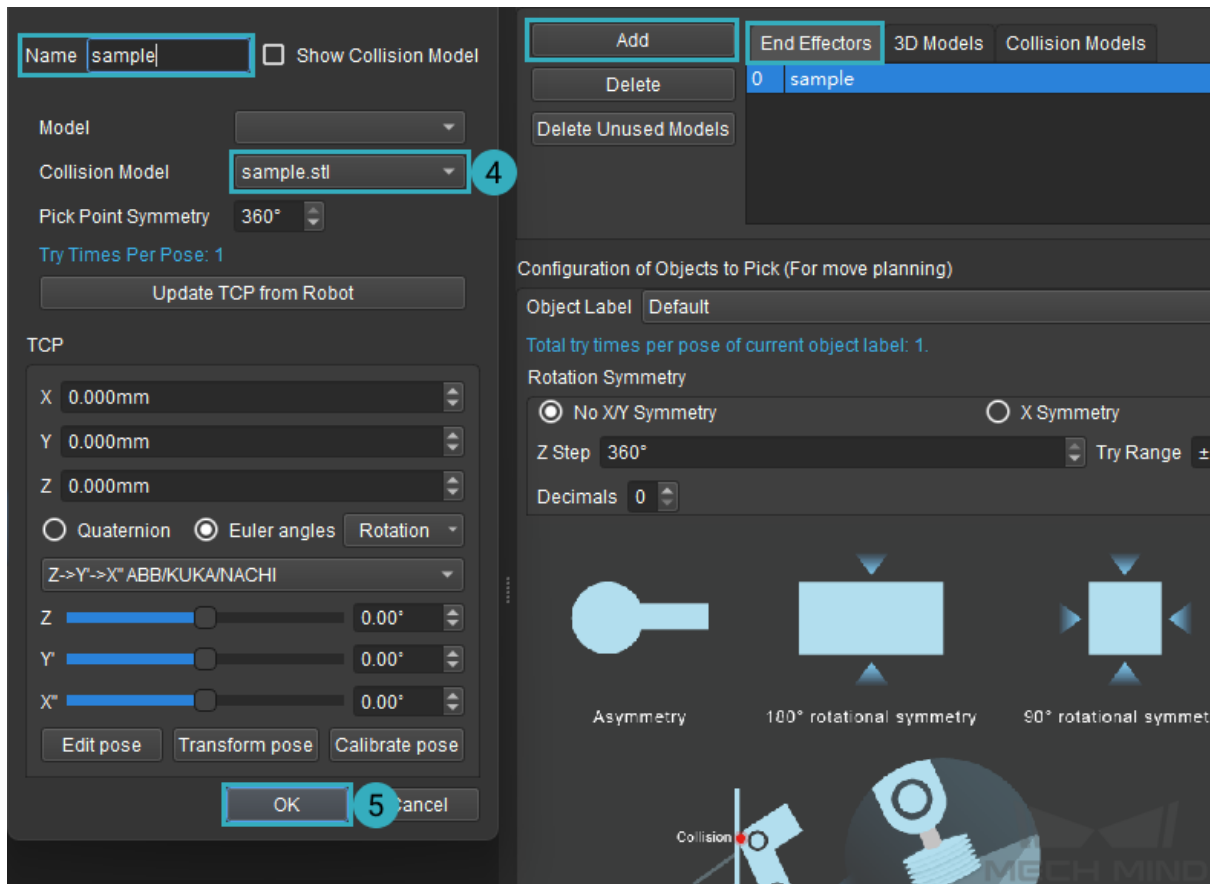
**Hint:** In this window, you can adjust the scale, position and rotation of the model. In real applications, please adjust these parameters according to your actual needs.

---

### Add End Effector

After adding the tool model to **Collision Models**, you need to add it to **End Effectors** as well for adjusting the TCP and switching tools when multiple tools are used.

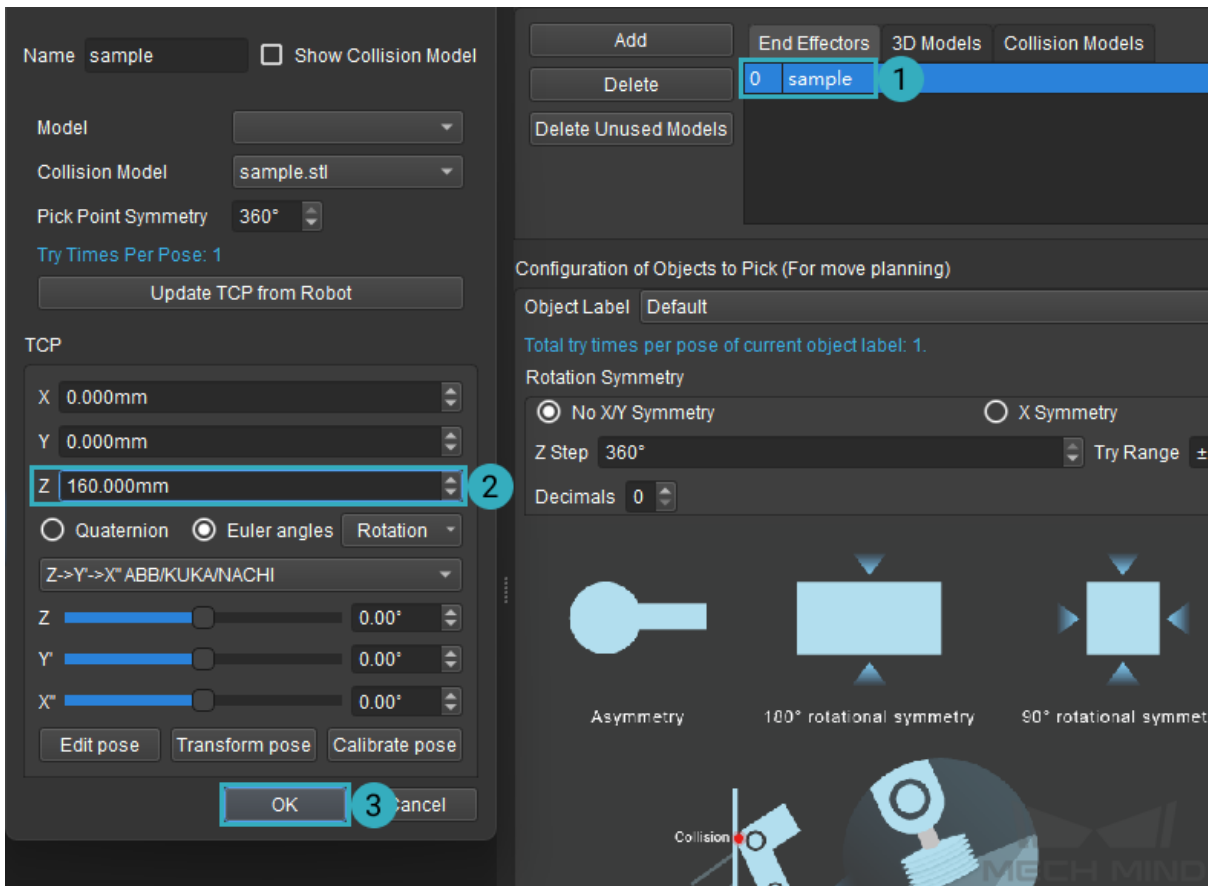
1. Click on the **End Effectors** tab to the left of **Collision Models**.
2. Click on *Add*.
3. In the pop-up window, type in a name for the robot tool.
4. Select the corresponding model from the drop-down menu of **Collision Model**.
5. Click on *OK* at the bottom to save the changes.



### Adjust Tool Center Point (TCP)

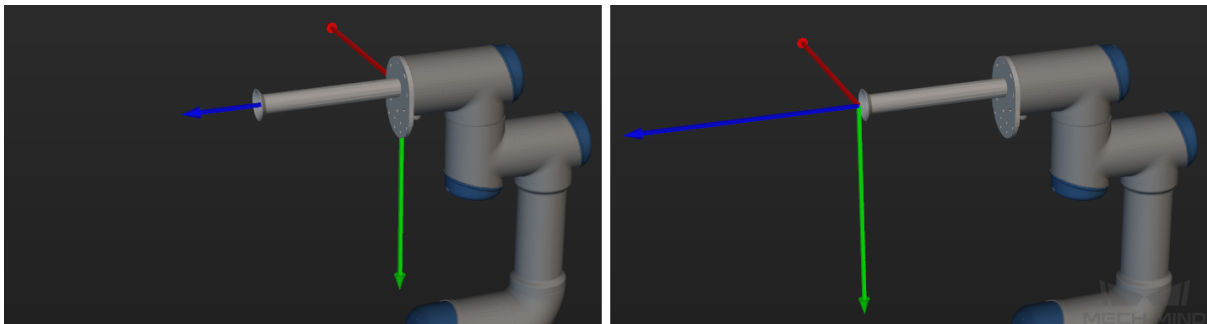
In the 3D Collision Area, the TCP is represented by the origin of the axes. Its default position is at the center of the robot flange. After adding a robot tool, you'll need to move the TCP to where the tool actually interacts with workobjects.

1. Double-click on the model name in the **End Effectors** tab.
2. In the pop-up window, increase the value of Z until the TCP is at the tip of the tool.
3. Click on *OK* to save the changes.



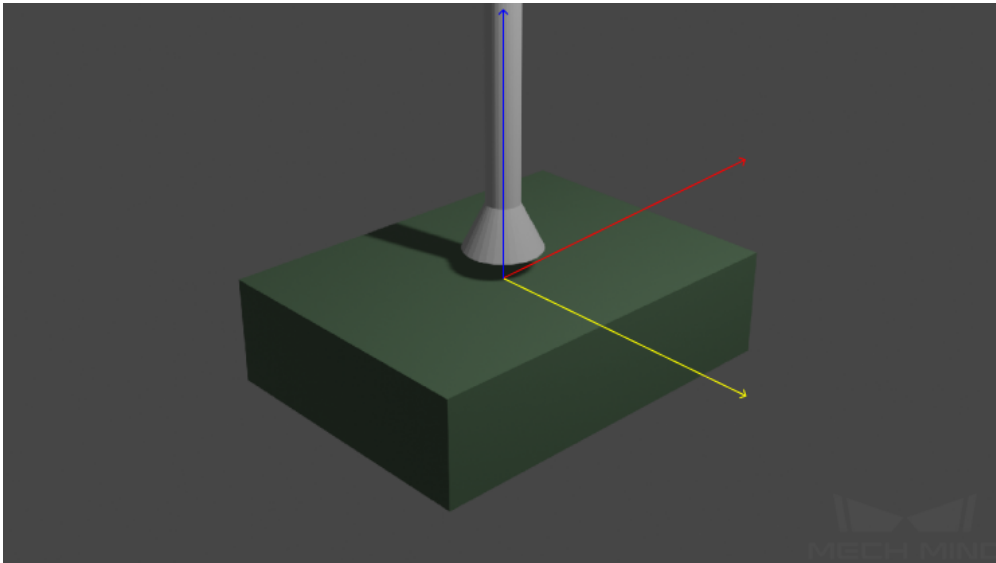
**Note:** In real applications, please input the calibrated TCP values. You can also use the *Update TCP from Robot* button to automatically fetch the current TCP values set on the robot.

The figure below shows the TCP before (left) and after (right) the above adjustment.



## Configure the Workobject

After configuring the robot tool, you'll need to configure the workobject (in this case, the carton) so that Mech-Viz can plan the robot path and picking strategy according to the characteristics of the workobject.



The following sections in the **Tools and Workobjects** tab can be configured:

- **Rotation Symmetry:** Here you can set the rotational symmetry of the workobject. In the example project, the carton's top surface is to be picked by the robot. This surface is a rectangle, and it overlaps with itself if you rotate it 180° about the its Z-axis (blue). Since in our simple example, the carton will be placed upright and picked by its top surface, you can think of the carton as not having X- or Y-axis symmetry. Therefore, the options to choose are: **No X/Y Symmetry**, **180° for Z Step**, and **±180° for Try Range**.
- **Picking Relaxation:** In carton picking scenarios, usually no picking relaxation is set. Because the rim of the suction cup has to be parallel to the carton surface, thus ensuring a firm grasp on the carton. Therefore, the robot must reach the picking pose precisely.
- **Optimal Picking Solution Strategy:** This parameter determines how much the robot tool is rotated during picking. In our example, set it to **Default** to avoid unnecessary rotation after the carton is picked up, which may result in dropping of the the carton.

If picking relaxation is set, the robot doesn't have to reach the picking pose precisely.

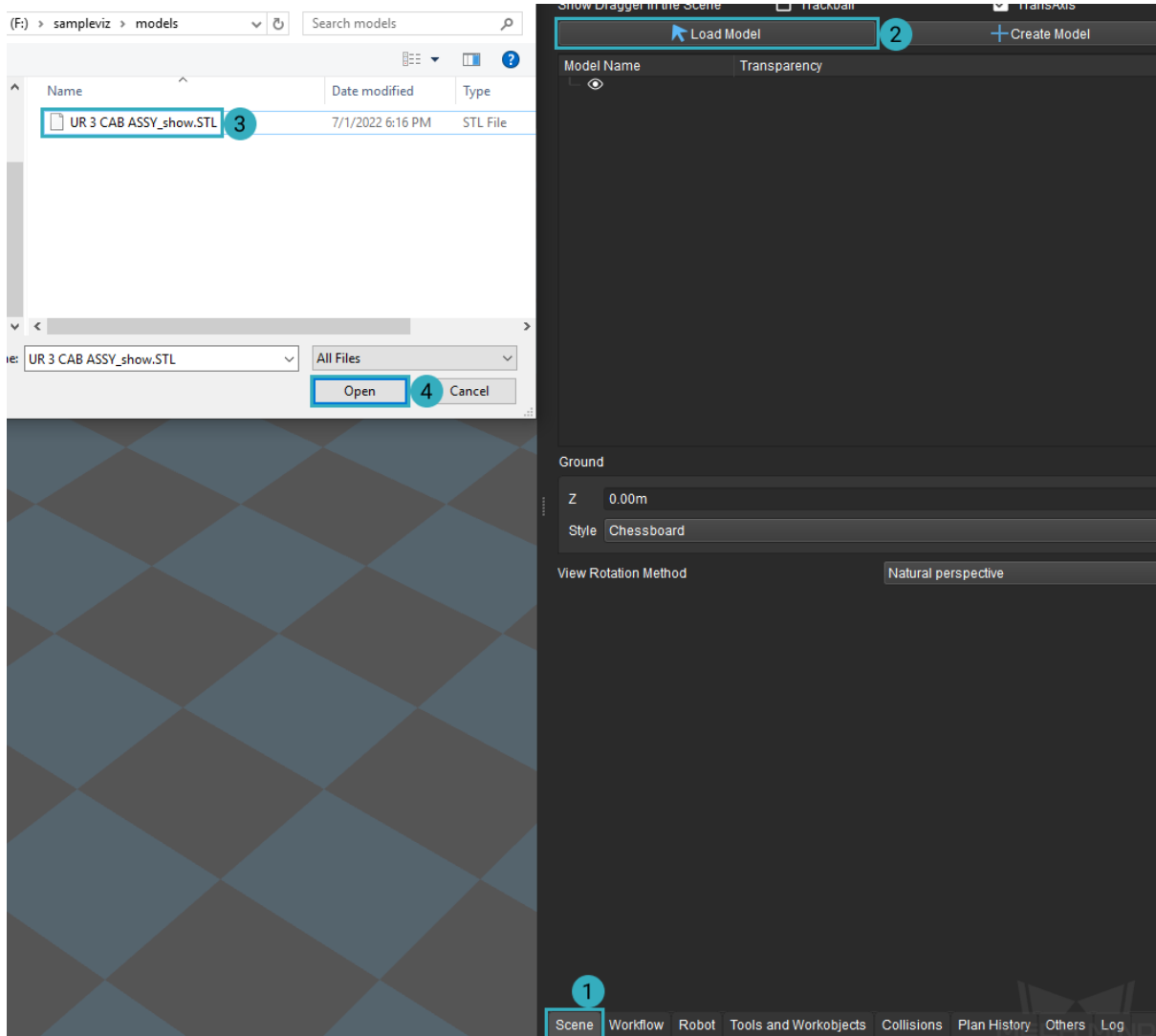
### 5.2.2 Configure Scene Object Models

With models of scene objects added, scene objects can be taken into consideration during collision detection, affecting the planning of path and picking strategy for the robot. These models are also displayed in the 3D simulation area, reproducing the actual scene more realistically.

In the **Scene** tab, you can create simple models directly with *Create Model*, or import already made models with *Load Model*.

In the example project, you can import a ready-made model from `\sampleviz\models`.

1. Click on the **Scene** tab in the lower right.
2. Click on *Load Model* at the top of the tab.
3. Select the **UR 3 CAB ASSY\_show.STL** file from the above location.
4. Click on *Open*.

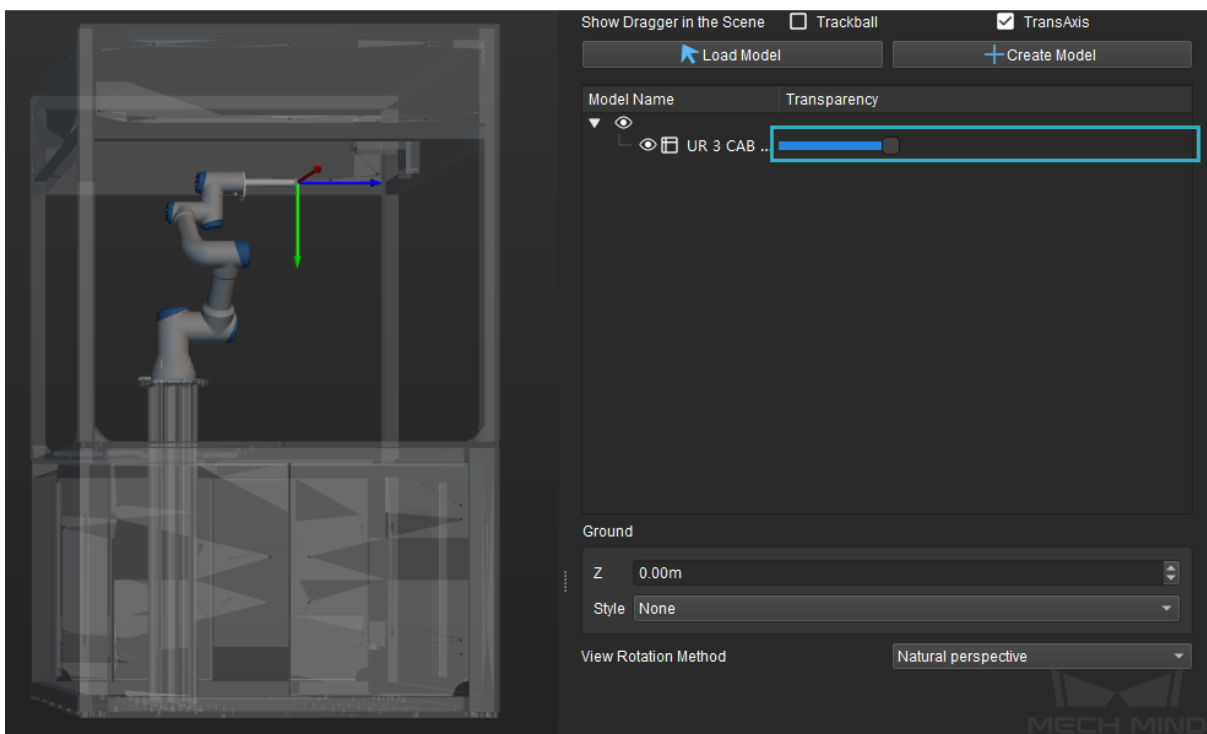
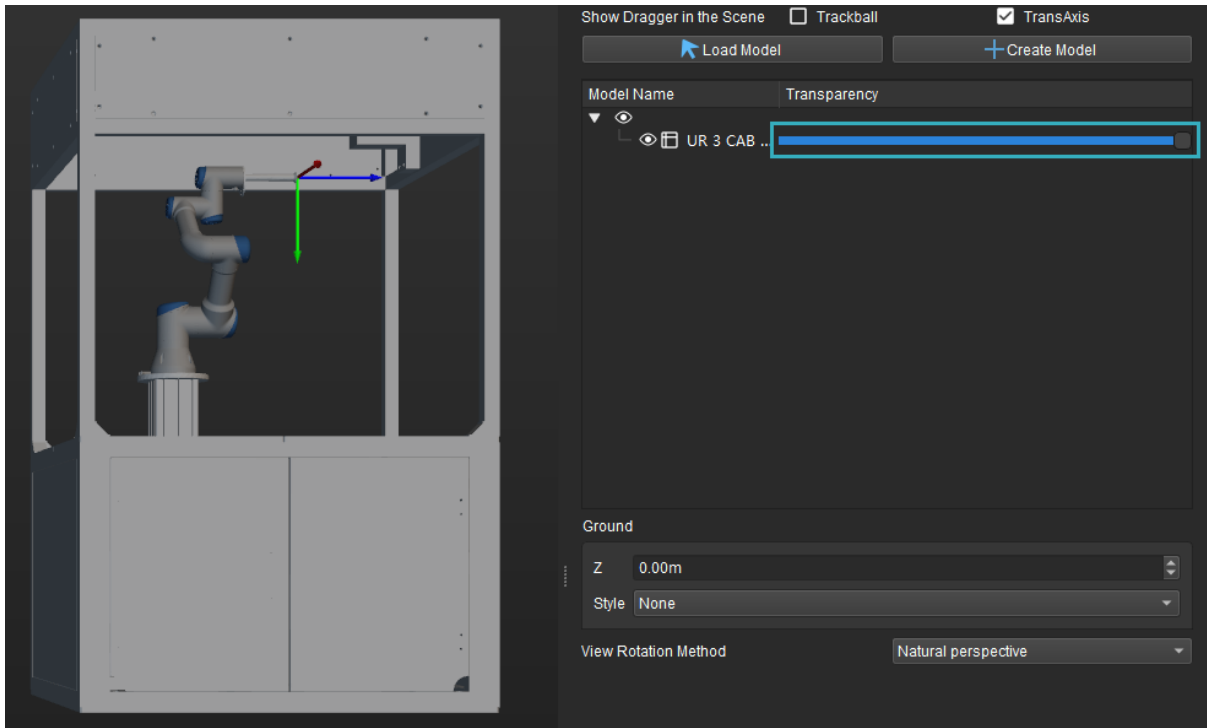


**Hint:** If the position of the added model is incorrect, you can adjust it by the following methods:

- Double-click on the model name and change the position and/or rotation values in the pop-up window.
- Check the **Trackball** and/or **TransAxis** options at the top of the **Scene** tab to display the translation/rotation manipulators of the model. Hold down **Ctrl** and click on a manipulator to drag it.

Scene object models might block your view of the planned path, poses, other objects, etc., in the 3D simulation area.

In such cases, you can adjust the opacity of the model by the slide bar to the right of the model name.

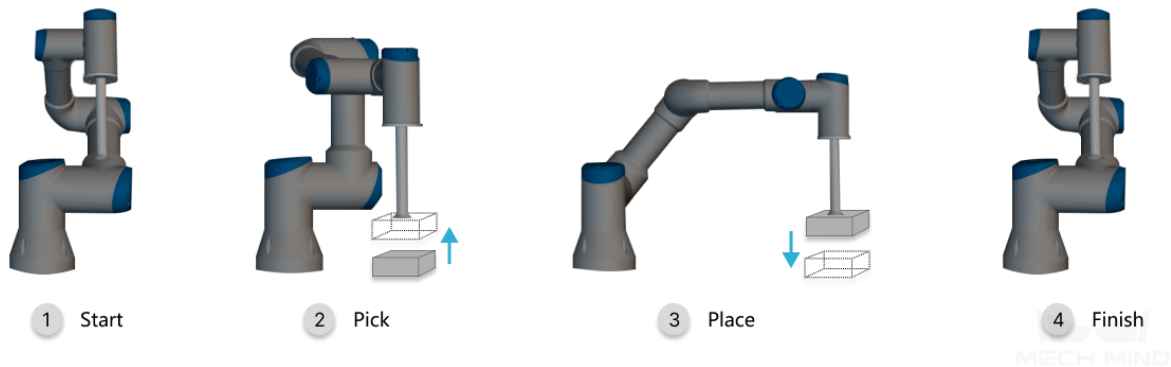


### 5.2.3 Configure the Workflow

Workflow is the core of a Mech-Viz project. It is where you configure what the robot does at what point.

In the **Workflow** tab, you'll add Tasks to the workflow, connect the Tasks, and adjust the parameters of Tasks to achieve the desired actions.

In the example project, you will create the following simple pick-and-place workflow.



#### Build the Workflow

##### Set Home Position

The home position is where the robot starts to move and where it will return at the end. The home position should be relatively removed from other objects, and the robot shouldn't block the camera when it's at the home position.

The move Task can be used to define a target pose on the robot's path and set how the robot moves to this pose.

---

**Note:** With the real robot connected, you can sync the pose of the real robot to the simulated robot in Mech-Viz.

---

1. In Mech-Viz, click on *Sync Robot* in the toolbar. The simulated robot will move in sync with the real robot.
2. Move the real robot to the desired home position. Check the simulated robot to make sure the pose of the real robot has been synced.
3. From the **Basic Move** category in the Task library, drag the move Task to the programming area. The current pose of the simulated robot will be automatically set as the target.

---

**Note:** If not connected to a real robot, you can still adjust the target pose of the move Task in the parameter panel. The simulated robot will display the adjusted pose simultaneously.

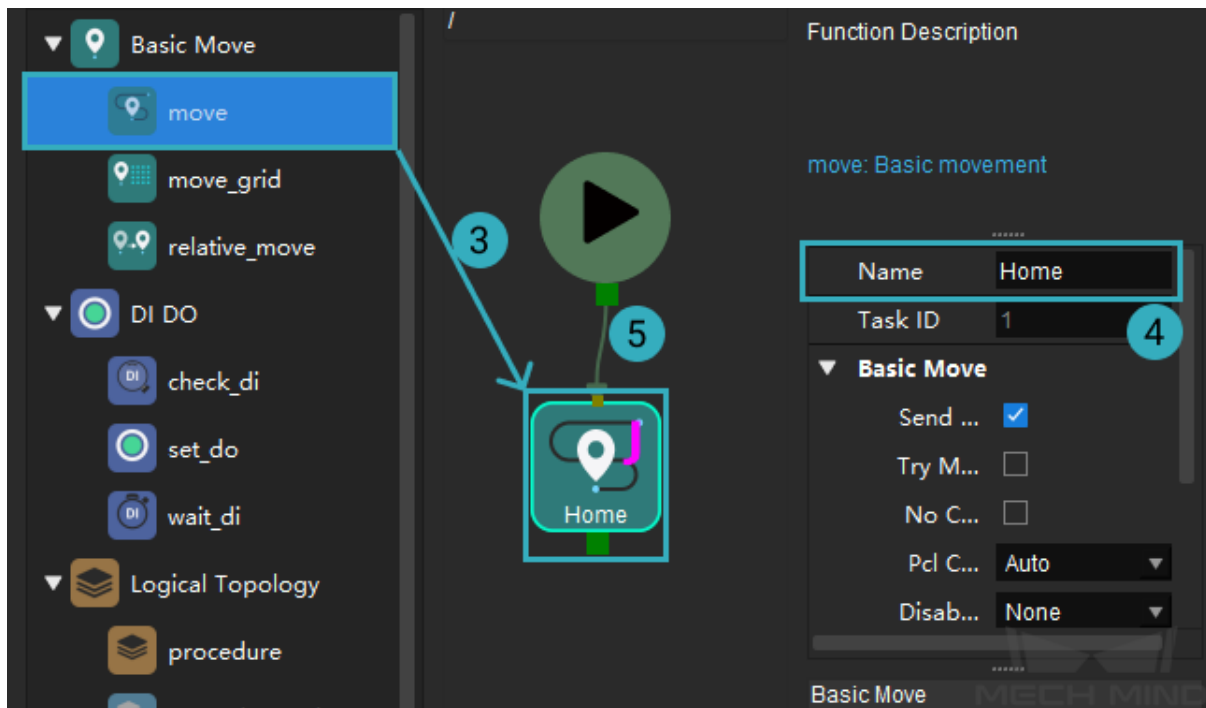
---

4. In the upper part of the parameter panel, rename the Task to **Home** in the **Name** field.



5. Connect the start icon to **Home**.

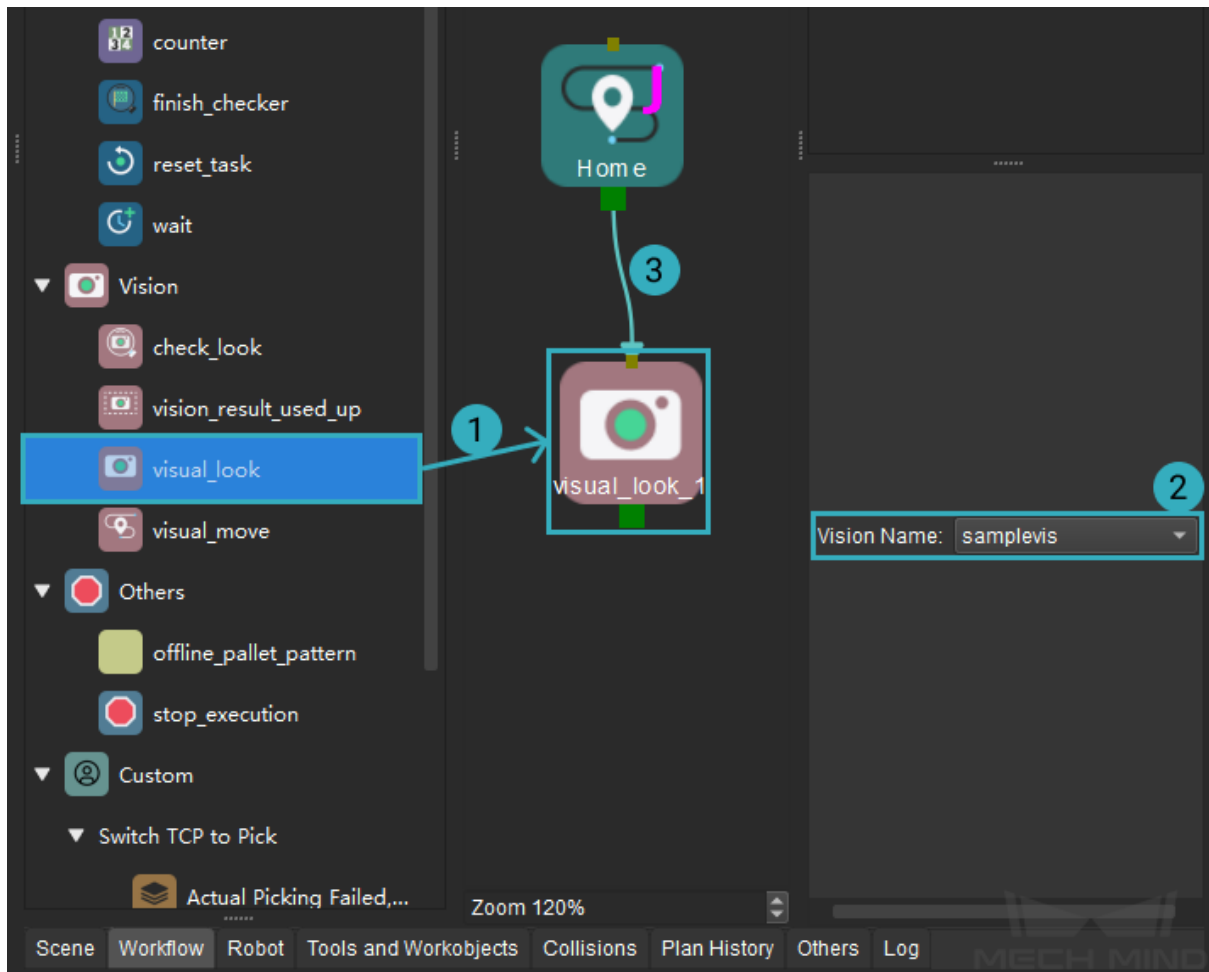
**Hint:** A connection is added by holding and dragging an exit port onto an entry port.



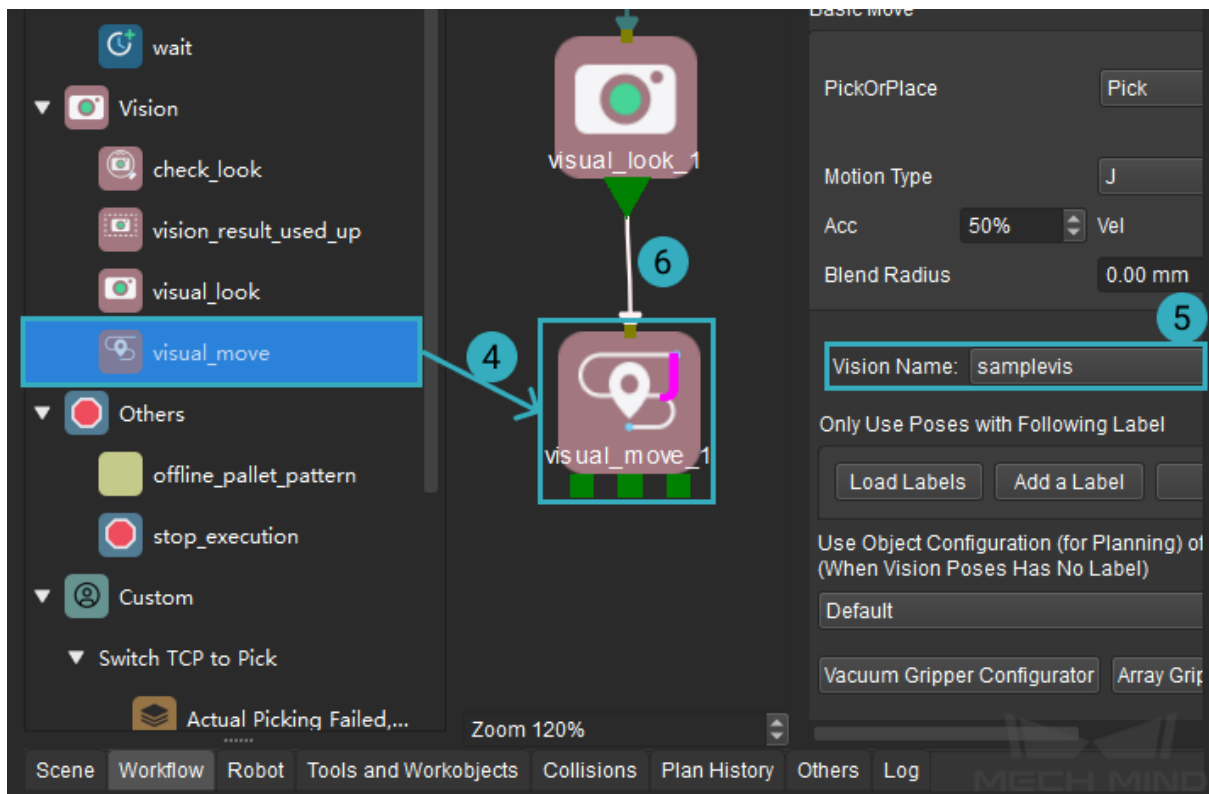
### Set Picking Target

Mech-Viz receives the workobject pose from Mech-Vision, and you'd want the robot to move according to this pose. This can be accomplished with the `vision_look` and `vision_move` Tasks.

- `vision_look` Task obtains the workobject pose from the corresponding Mech-Vision project.
  - `vision_move` Task moves the robot according to the received workobject pose.
1. From the **Vision** category in the Task library, drag the `vision_look` Task to the programming area. This Task is automatically named `visual_look_1`.
  2. In the lower part of the parameter panel, select **samplevis** (the example Mech-Vision project) from the drop-down menu of **Vision Name**.  
This will enable the Task to obtain workobject poses from the Mech-Vision project named **samplevis**.
  3. Connect **Home** to `visual_look_1`.



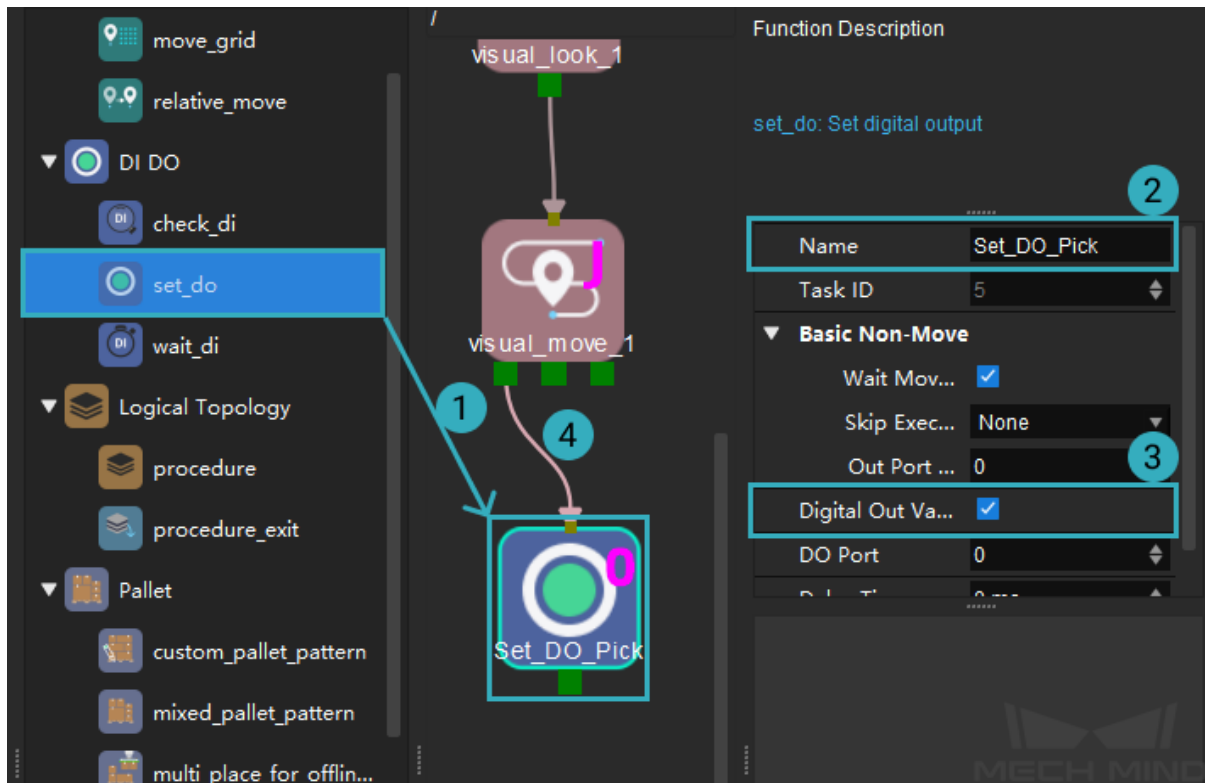
4. From the **Vision** category in the Task library, drag the `vision_move` Task to the programming area. This Task is automatically named `visual_move_1`.
5. In the lower part of the parameter panel, select `samplevis` from the drop-down menu of **Vision Name**.
6. Connect the exit port of `visual_look_1` to the entry port of `visual_move_1`.



### Set DO for Picking

Once the robot reaches the picking target, the robot has to operate its tool to pick up the workobject. This is achieved with the set\_do Task, which instructs the robot to operate its tool.

1. From the **DI DO** category in the Task library, drag the set\_do Task to the programming area.
2. In the parameter panel, rename the Task to **Set\_DO\_Pick** in the **Name** field.
3. Check the checkbox of **Digital Out Value**, which tells the robot to start generating vacuum.
4. Connect the left-most exit port of **visual\_move\_1** to **Set\_DO\_Pick**.



### Set Placing Target

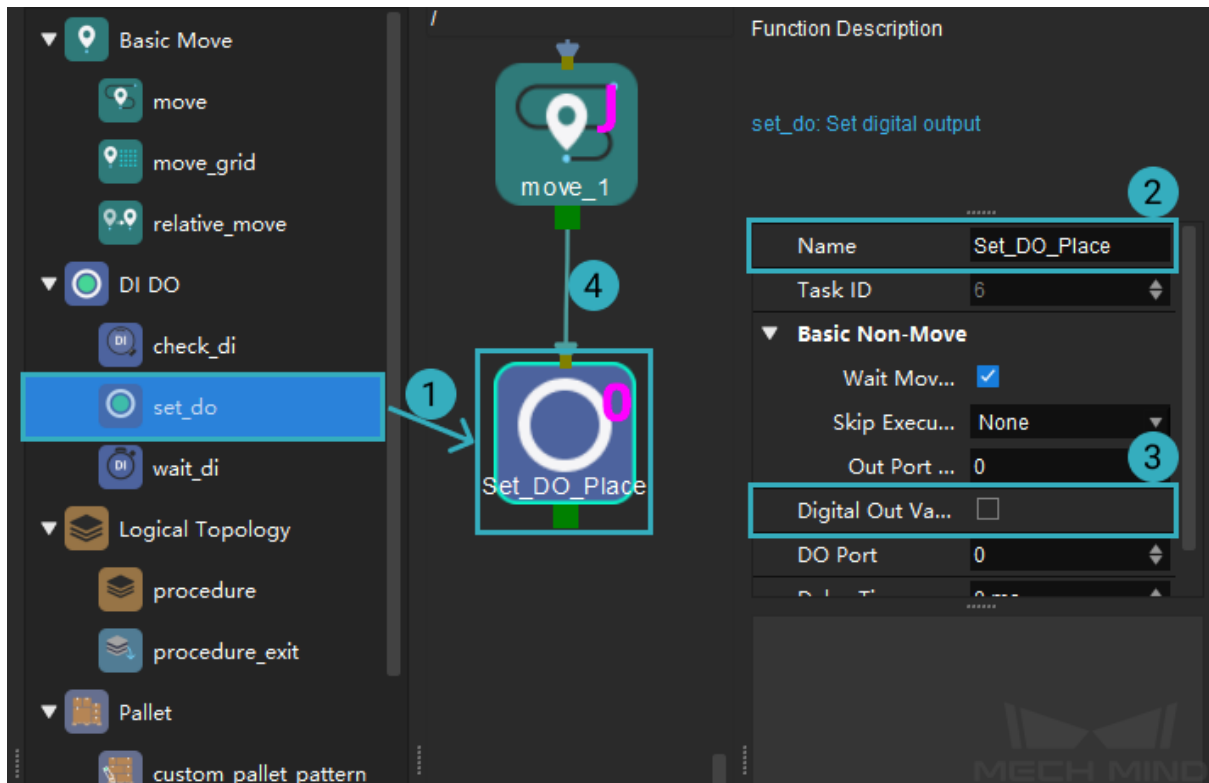
After picking up the workobject, the robot should move to a predetermined position to drop off the workobject. Once again, the move Task can be used to define the placing target.

1. Move the real robot to the desired placing target. Check the simulated robot to make sure the pose of the real robot has been synced.
2. From the **Basic Move** category in the Task library, drag the move Task to the programming area. This Task is automatically named `move_1`.
3. Connect `Set_DO_Pick` to `move_1`.

### Set DO for Placing

Once the robot reaches the placing target, the robot has to once again operate its tool to drop off the workobject, which is achieved with the `set_do` Task.

1. From the **DI DO** category in the Task library, drag the `set_do` Task to the programming area.
2. In the parameter panel, rename the Task to `Set_DO_Place` in the **Name** field.
3. Uncheck the checkbox of **Digital Out Value**, which tells the robot to release.
4. Connect `move_1` to `Set_DO_Place`.



## Return to Home

After the robot drops off the workobject, the next step is to return to its home position. As you already set the home position in **Home**, now you just need to copy and paste this Task.

1. Right-click on **Home** and select **Copy**, or left-click on **Home** and then use the shortcut Ctrl + C.
2. Navigate down to the end of the workflow. Right-click on the empty area and select **Paste**, or use the shortcut Ctrl + V.
3. This pasted Task is automatically named **move\_2**. Rename it to **Return\_Home** in the **Name** field in the parameter panel.
4. Connect **Set\_DO\_Place** to **Return\_Home**.

The entire workflow now looks like this:



**Hint:** Right-click in the programming area and select **Format** to automatically align the Tasks and adjust their spacing.

---

Click on *Simulate* in the toolbar, the simulated robot will move according to the workflow.

You can see that although we built the workflow according to the correct pick-and-place logic, there are several problems:

- The robot motion from the home position to the picking target is too large.
- Moving right after setting DO signals doesn't provide enough time for the robot to pick/place.
- It would be safer if the robot moves vertically down and up right before and after it performs picking and placing.

Now let's improve the workflow to resolve the problems.

## Improve the Workflow

### Add Waypoint

Add a waypoint between the home position and the picking target to break down the motion into two parts.

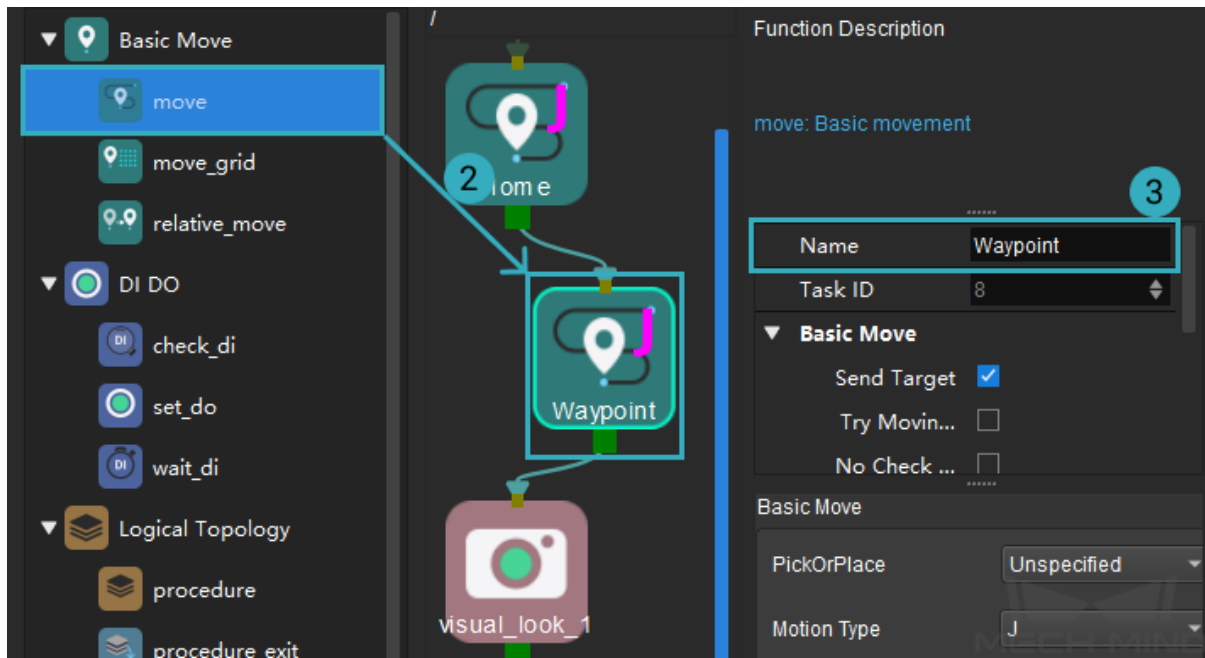
1. Move the real robot to the desired waypoint. Check the simulated robot to make sure the pose of the real robot has been synced.

---

**Note:** The robot should not block the view of the camera when its at this waypoint. You can capture an image to check if the robot is in the way.

---

2. From the **Basic Move** category in the Task library, drag the move Task to the programming area.
3. In the upper part of the parameter panel, rename the Task to **Waypoint** in the **Name** field.
4. Connect **Home** to **Waypoint** and **Waypoint** to **visual\_look\_1**.



### Add Relative Targets for Picking

The more practical picking motion should be: the tool moves vertically down to the picking target, and then moves vertically up after it picks up the workobject.

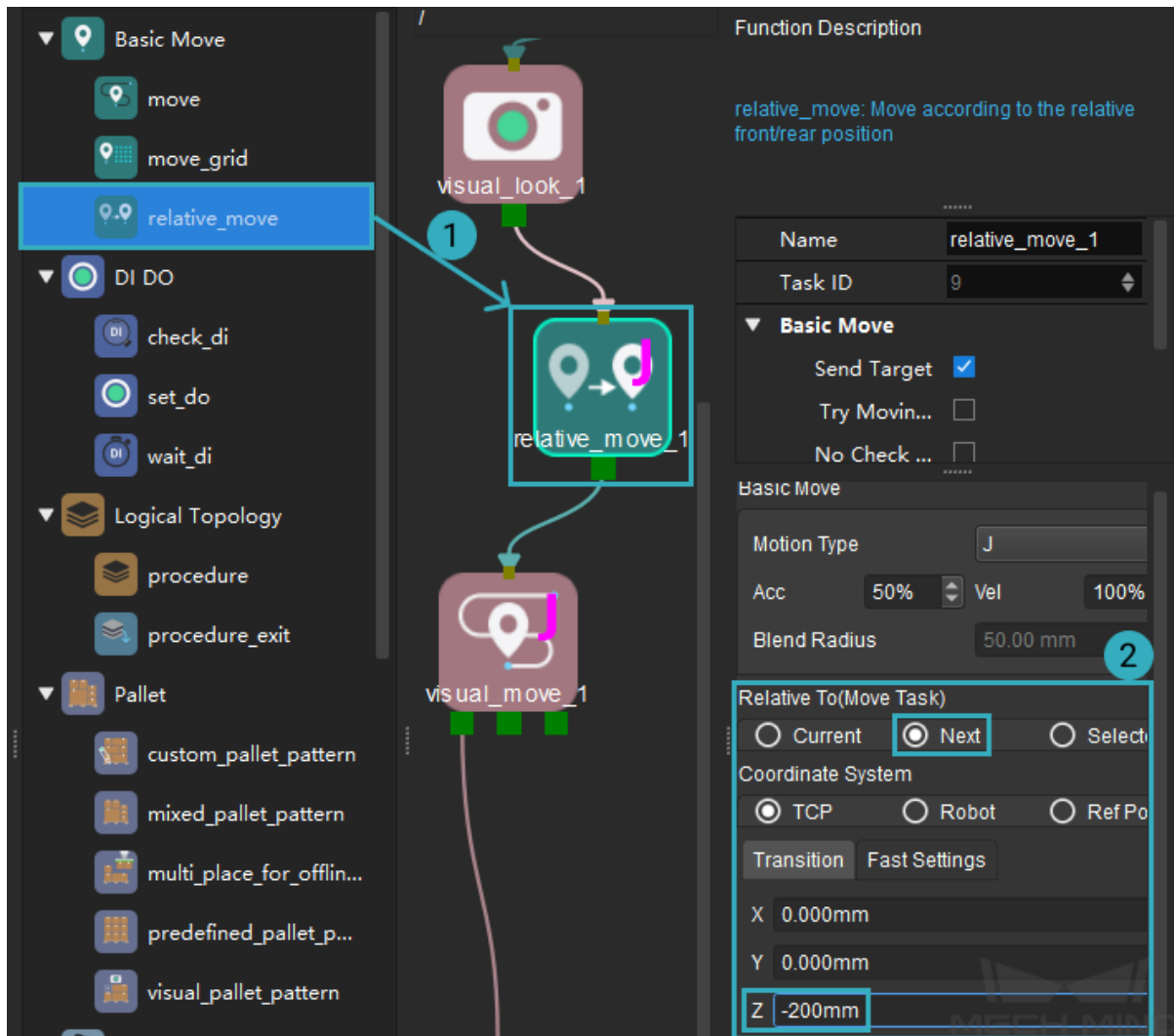
The `relative_move` Task can be used to add targets relative to other targets.

### Approaching the Picking Target

Move the robot to 200 mm above the picking target first, and then move it to the picking target from there.

1. From the **Basic Move** category in the Task library, drag the `relative_move` Task to the programming area. The Task is automatically named `relative_move_1`.
2. In the parameter panel, set the following fields:
  - **Relative To(move-type Task)**: select **Next**, which means that the target defined here is relative to the next target in the Workflow (i.e., the target of `visual_move_1`).
  - **Transition** tab: set the value of Z to -200.
3. Connect `visual_look_1` to `relative_move_1` and `relative_move_1` to `visual_look_1`.



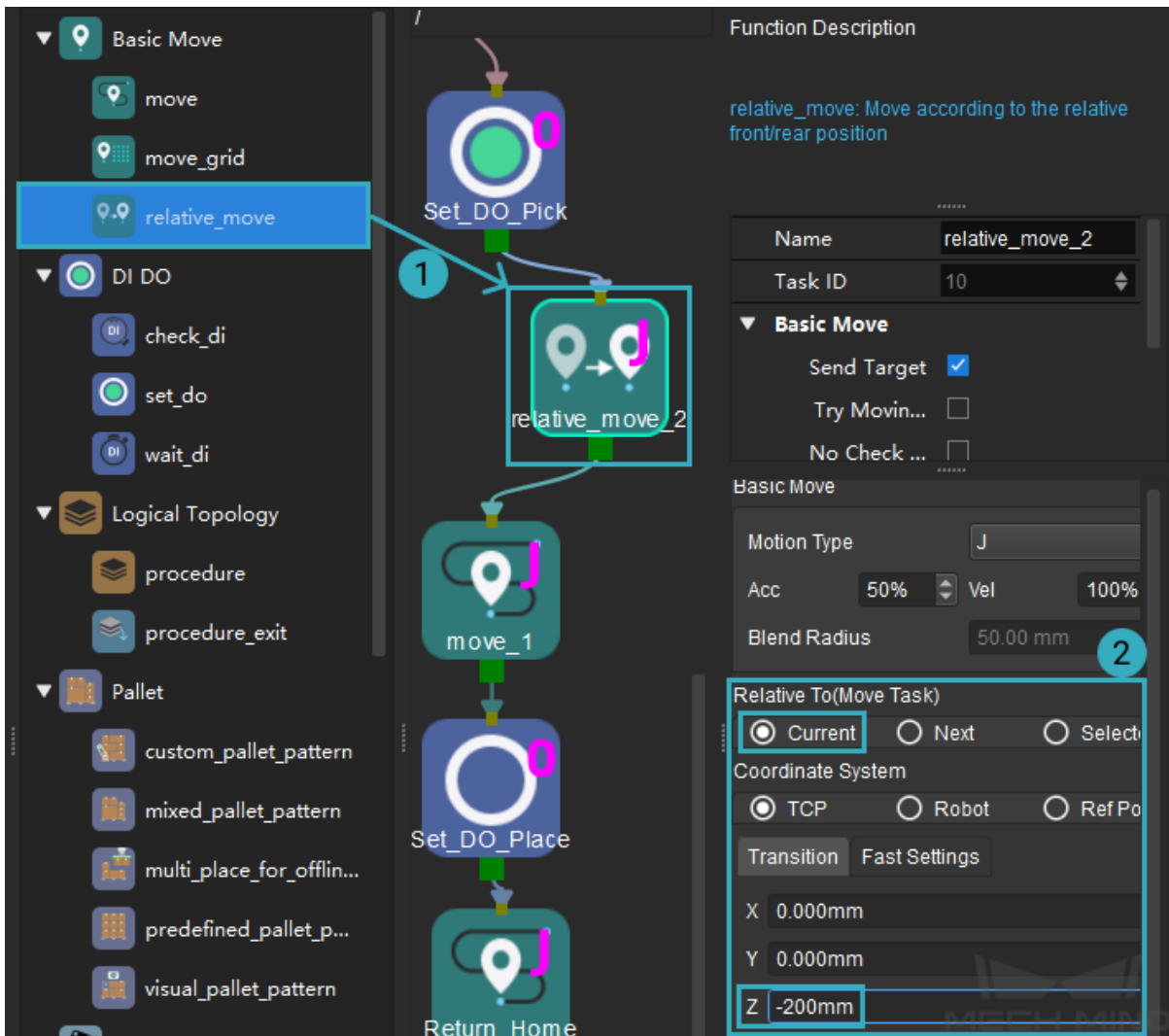


The screenshot displays the Mech-Mind software interface. On the left, a task library is visible with the 'Basic Move' category expanded, and 'relative\_move' selected. The main workspace shows a workflow diagram with three tasks: 'visual\_look\_1', 'relative\_move\_1', and 'visual\_move\_1'. A red circle '1' highlights the 'relative\_move' task in the library, and another red circle '2' highlights the 'Z' parameter in the 'Transition' tab of the 'relative\_move\_1' task's parameter panel. The parameter panel shows 'Motion Type' set to 'J', 'Acc' at '50%', 'Vel' at '100%', and 'Blend Radius' at '50.00 mm'. The 'Relative To(Move Task)' section has 'Next' selected, and the 'Coordinate System' is set to 'TCP'. The 'Transition' tab shows 'X' at '0.000mm', 'Y' at '0.000mm', and 'Z' at '-200mm'.

### Departing the Picking Target

Move the robot to 200 mm above the picking target after it picks up the workobject.

1. From the **Basic Move** category in the Task library, drag the `relative_move` Task to the programming area. The Task is automatically named `relative_move_2`.
2. In the parameter panel, set the following fields:
  - **Relative To(move-type Task)**: select **Current**, which means that the target defined here is relative to the previous target in the Workflow (i.e., the target of `visual_move_1`).
  - **Transition** tab: set the value of **Z** to **-200**.
3. Connect `Set_DO_Pick` to `relative_move_2` and `relative_move_2` to `move_1`.



The screenshot displays the Mech-Mind software interface. On the left is a task library with categories: Basic Move (move, move\_grid, relative\_move), DI DO (check\_di, set\_do, wait\_di), Logical Topology (procedure, procedure\_exit), and Pallet (custom\_pallet\_pattern, mixed\_pallet\_pattern, multi\_place\_for\_offlin..., predefined\_pallet\_p..., visual\_pallet\_pattern). The main workspace shows a task sequence: Set\_DO\_Pick, relative\_move\_2, move\_1, Set\_DO\_Place, and Return Home. A red circle '1' highlights the 'relative\_move' task in the library and its corresponding task in the sequence. A red circle '2' highlights the 'Blend Radius' field in the 'Basic Move' configuration panel, which is set to 50.00 mm. The 'Function Description' panel for 'relative\_move\_2' shows: Name: relative\_move\_2, Task ID: 10, Basic Move settings (Send Target checked, Try Movin... unchecked, No Check ... unchecked), Motion Type: J, Acc: 50%, Vel: 100%, Blend Radius: 50.00 mm. The 'Relative To(Move Task)' panel shows 'Current' selected. The 'Coordinate System' panel shows 'TCP' selected. The 'Transition' panel shows 'Fast Settings' selected. The 'X' and 'Y' coordinates are 0.000mm, and the 'Z' coordinate is -200mm.

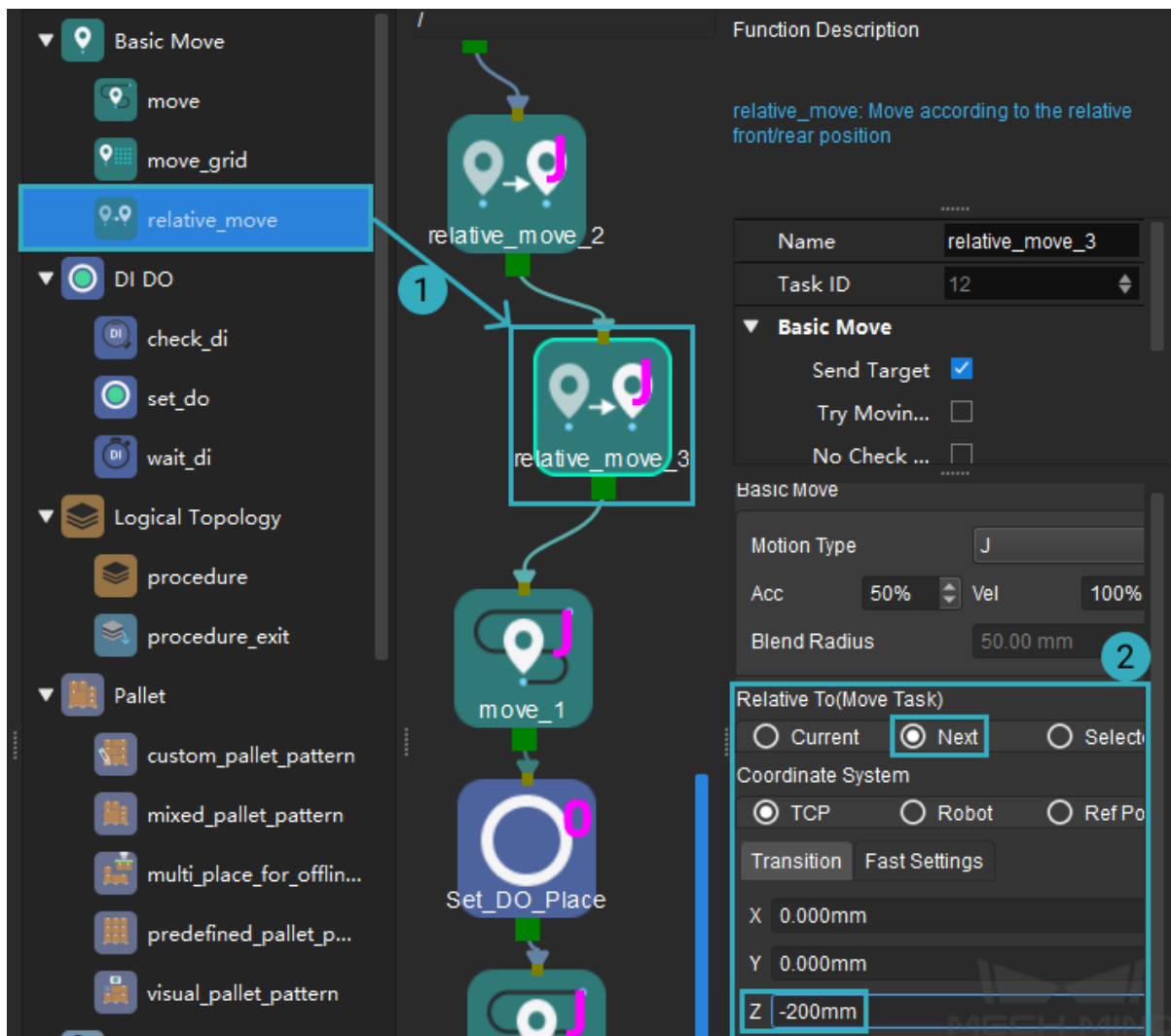
### Add Relative Targets for Placing

Similarly, the more practical placing motion should be: the tool moves vertically down to the placing target, and then moves vertically up after it drops off the workobject.

### Approaching the Placing Target

Move the robot to 200 mm above the placing target first, and then move it to the placing target from there.

1. From the **Basic Move** category in the Task library, drag the `relative_move` Task to the programming area. The Task is automatically named `relative_move_3`.
2. In the parameter panel, set the following fields:
  - **Relative To(move-type Task)**: select **Next**, which means that the target defined here is relative to the next target in the Workflow (i.e., the target of `move_1`).
  - **Transition** tab: set the value of **Z** to `-200`.
3. Connect `relative_move_2` to `relative_move_3` and `relative_move_3` to `move_1`.

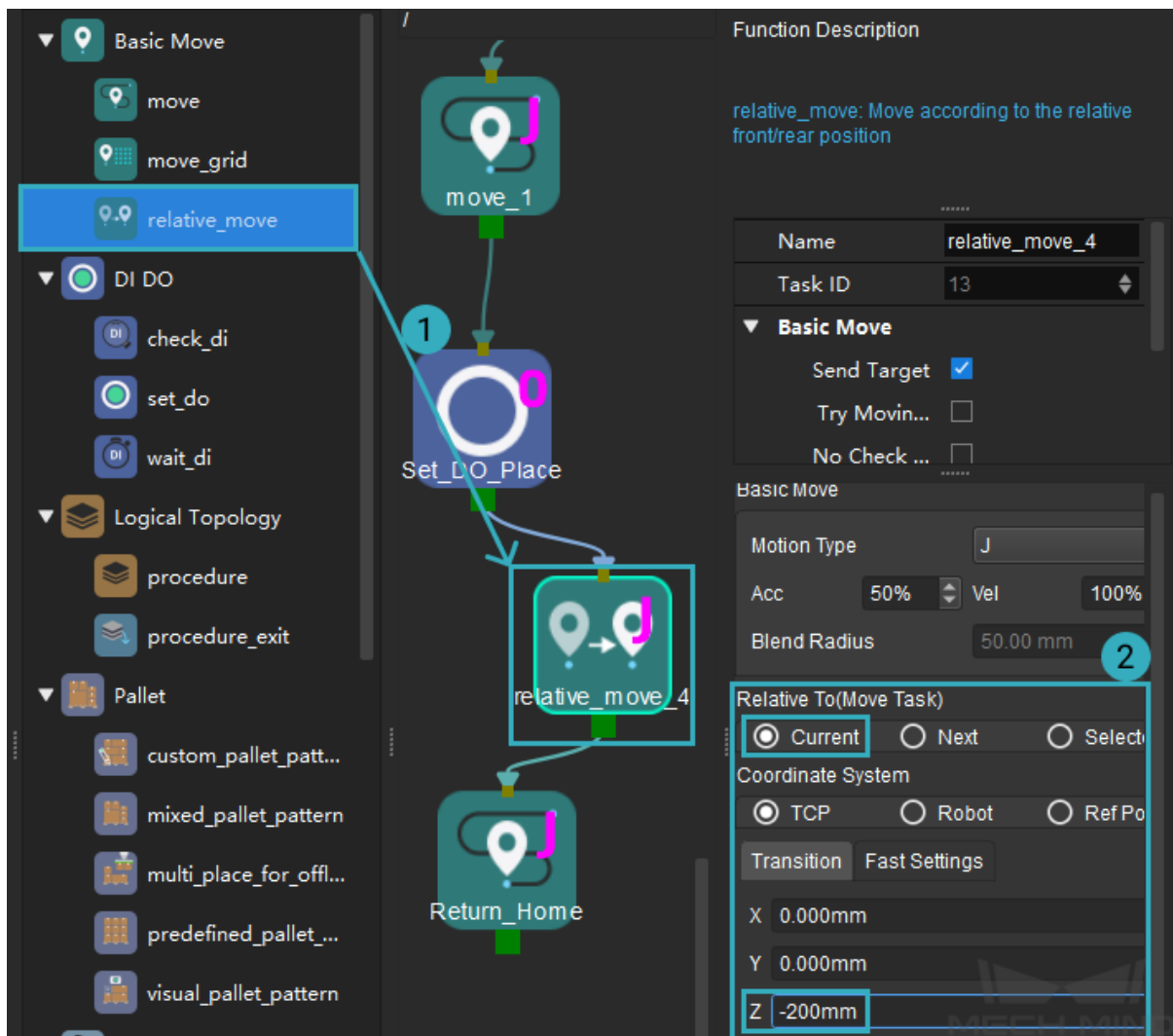


The screenshot displays the Mech-Mind software interface. On the left, a task library is visible with categories like 'Basic Move', 'DI DO', 'Logical Topology', and 'Pallet'. The 'relative\_move' task is highlighted in the 'Basic Move' category. In the center, a workflow diagram shows a sequence of tasks: 'relative\_move\_2' (highlighted with a red box and a '1' callout), 'relative\_move\_3' (highlighted with a red box and a '2' callout), 'move\_1', and 'Set\_DO\_Place'. On the right, the 'Function Description' and 'Basic Move' configuration panels are shown. The 'Basic Move' panel includes settings for 'Motion Type' (J), 'Acc' (50%), 'Vel' (100%), and 'Blend Radius' (50.00 mm). The 'Relative To(Move Task)' section has 'Next' selected. The 'Transition' tab is active, and the 'Z' coordinate is set to '-200mm'.

### Departing the Placing Target

Move the robot to 200 mm above the placing target after it drops off the workobject.

1. From the **Basic Move** category in the Task library, drag the `relative_move` Task to the programming area. The Task is automatically named `relative_move_4`.
2. In the parameter panel, set the following fields:
  - **Relative To(move-type Task):** select **Current**, which means that the target defined here is relative to the previous target in the Workflow (i.e., the target of `move_1`).
  - **Transition tab:** set the value of **Z** to `-200`.
3. Connect `Set_DO_Place` to `relative_move_4` and `relative_move_4` to `Return_Home`.

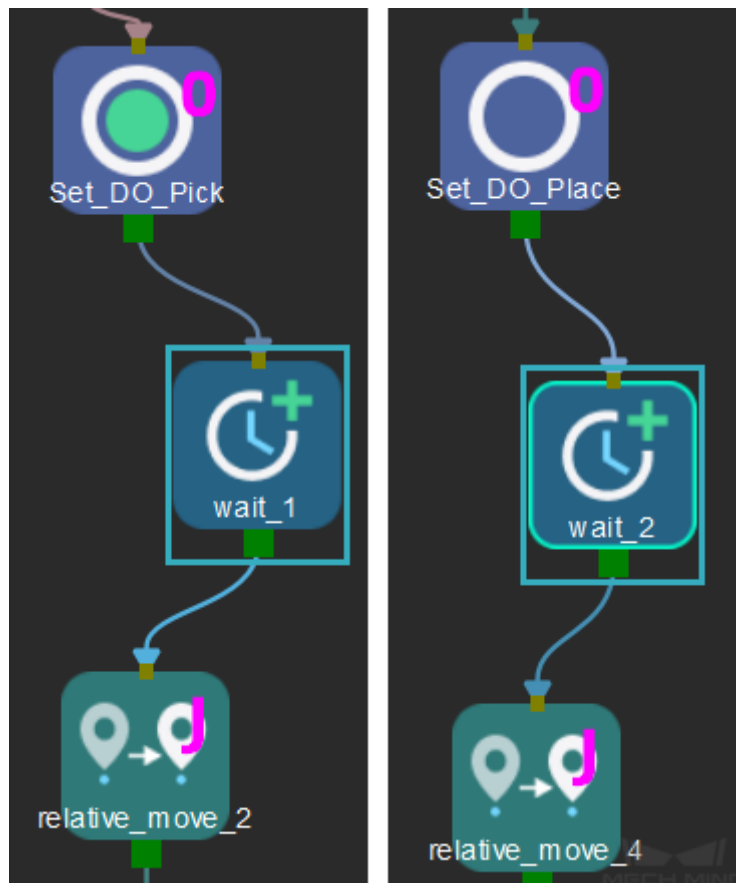


The screenshot displays the Mech-Mind software interface. On the left, a task library is visible with categories like 'Basic Move', 'DI DO', 'Logical Topology', and 'Pallet'. The 'relative\_move' task is highlighted in blue. In the center, a workflow diagram shows a sequence of tasks: 'move\_1' (a location pin icon), 'Set\_DO\_Place' (a circular icon with a '0'), 'relative\_move\_4' (a location pin icon with a right-pointing arrow), and 'Return\_Home' (a location pin icon). A blue circle with the number '1' is placed over the 'Set\_DO\_Place' task, and a blue circle with the number '2' is placed over the 'relative\_move\_4' task. On the right, the 'Function Description' and configuration panels for 'relative\_move\_4' are shown. The 'Basic Move' section includes 'Send Target' (checked), 'Try Movin...' (unchecked), and 'No Check ...' (unchecked). The 'Basic Move' section also shows 'Motion Type' set to 'J', 'Acc' at '50%', and 'Vel' at '100%'. The 'Blend Radius' is set to '50.00 mm'. The 'Relative To(Move Task)' section has 'Current' selected. The 'Coordinate System' section has 'TCP' selected. The 'Transition' tab is active, and the 'Z' coordinate is set to '-200mm'.

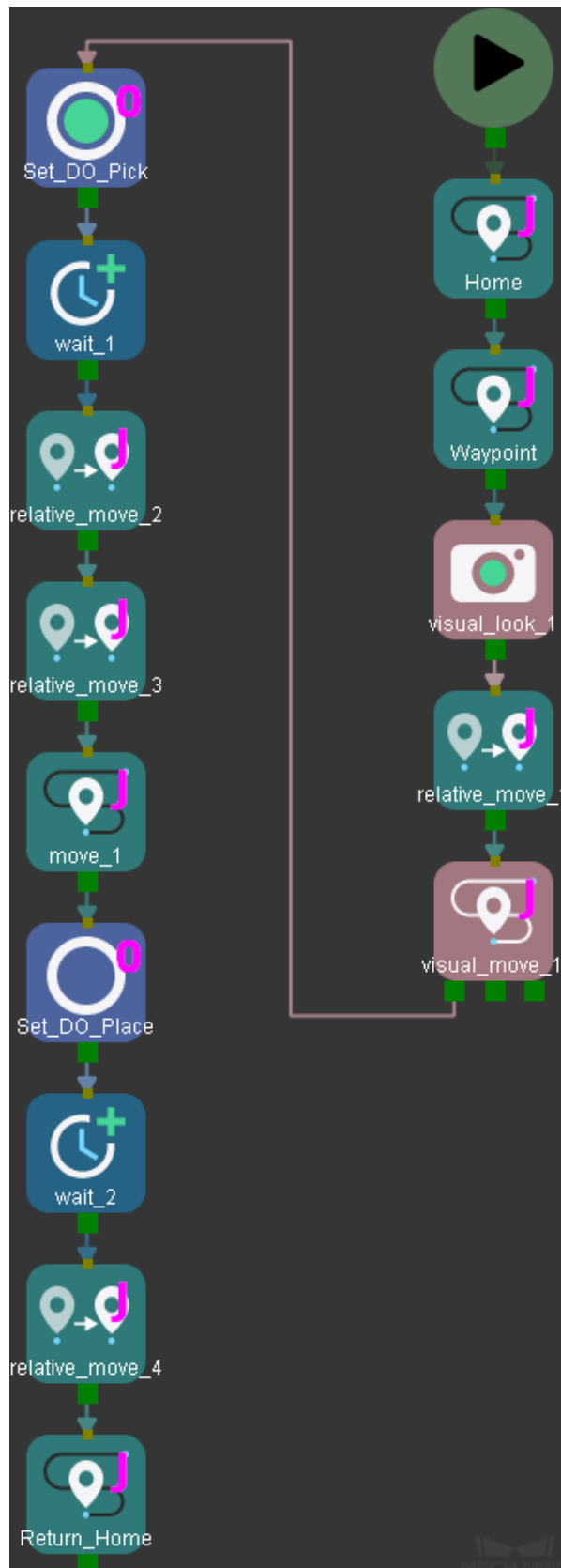
### Wait after Setting DO Signals

Make the robot wait a little while after the DO signal is set to ensure that the picking and releasing actions are fully completed. This can be accomplished with the wait Task.

1. From the **Tools** category in the Task library, drag the wait Task to the programming area. The Task is automatically named **wait\_1**.
2. Connect **Set\_DO\_Pick** to **wait\_1** and **wait\_1** to **relative\_move\_2**.
3. From the **Tools** category in the Task library, drag the wait Task to the programming area. The Task is automatically named **wait\_2**.
4. Connect **Set\_DO\_Place** to **wait\_2** and **wait\_2** to **relative\_move\_4**.



The complete workflow should look like this:



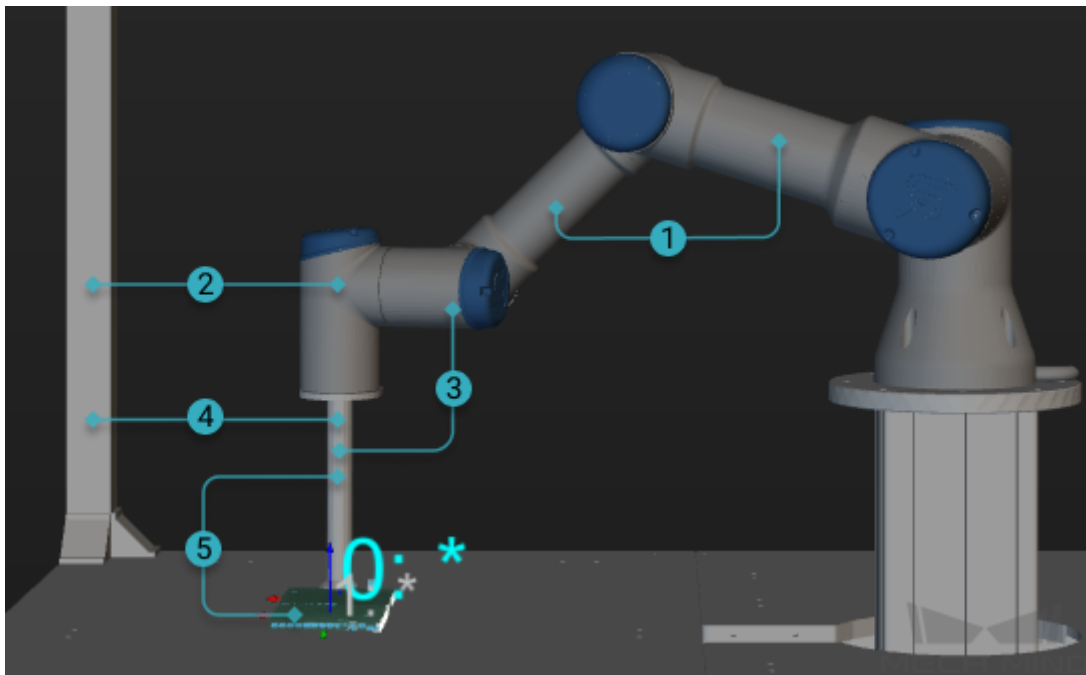
Click on *Simulate* in the toolbar, the simulated robot will move according to the workflow.

You can see that the robot path is more smooth and safe now.

### 5.2.4 Configure Collision Detection

To prevent the robot from colliding with other objects while it performs its tasks, collision detection must be configured.

In the **Collisions** tab, click on *Collision Detection Configuration* to set between which two objects you wish to check for collision.

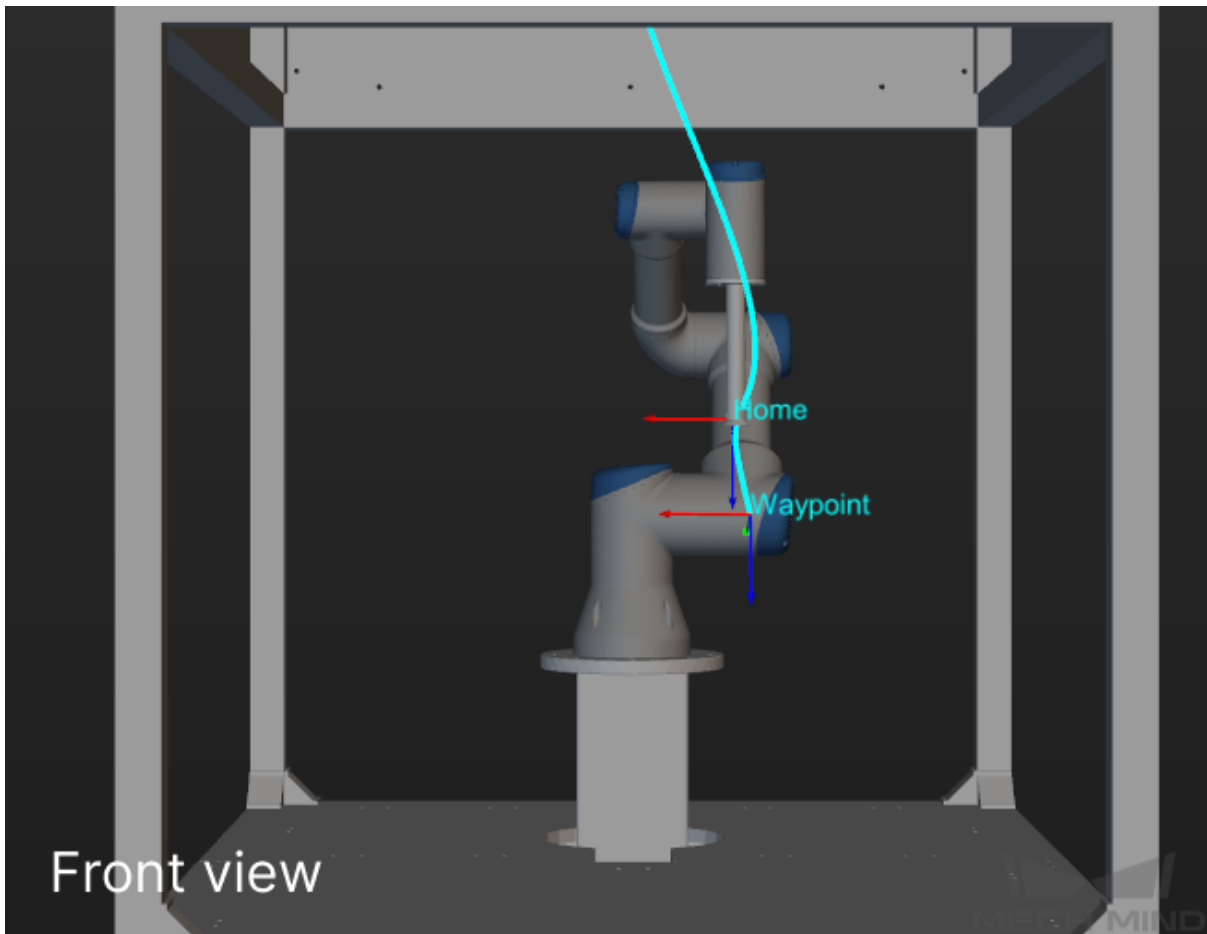


The following object pairs are checked for collision by default:

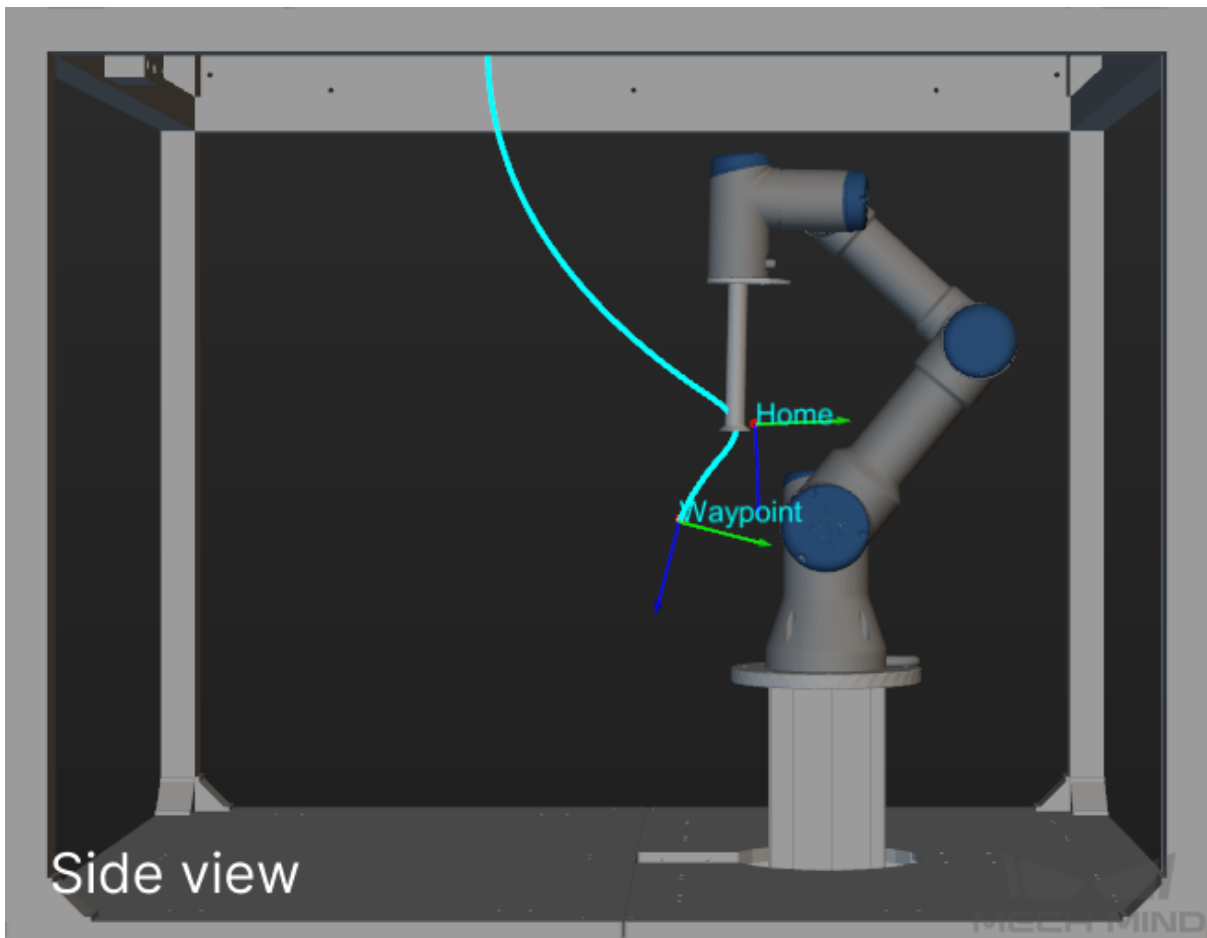
1. Robot links and robot links
2. Robot links and scene objects
3. Robot links and robot tool (end effector)
4. Scene objects and robot tool
5. If you turn on **Detect collision between point cloud and others**, the collision between point cloud and robot tool is also detected.

### 5.2.5 Simulate the Project

After finishing all the configurations in the previous chapters, click on *Simulate* to simulate the project. The following are different views of the simulated robot's motion.







So far, you have simulated the robot motion path successfully. Now you can move the real robot to complete the box palletizing task.

---

**Note:** The settings in each of the above steps are described with the [example Mech-Viz project](#) .

---

### 5.3 Use Mech-Center to Control a Real Robot to Pick

So far, you have successfully obtained the box poses and successfully simulated the picking and placing of a box. Now you can try running the real robot to physically implement the picking and placing for the first time.

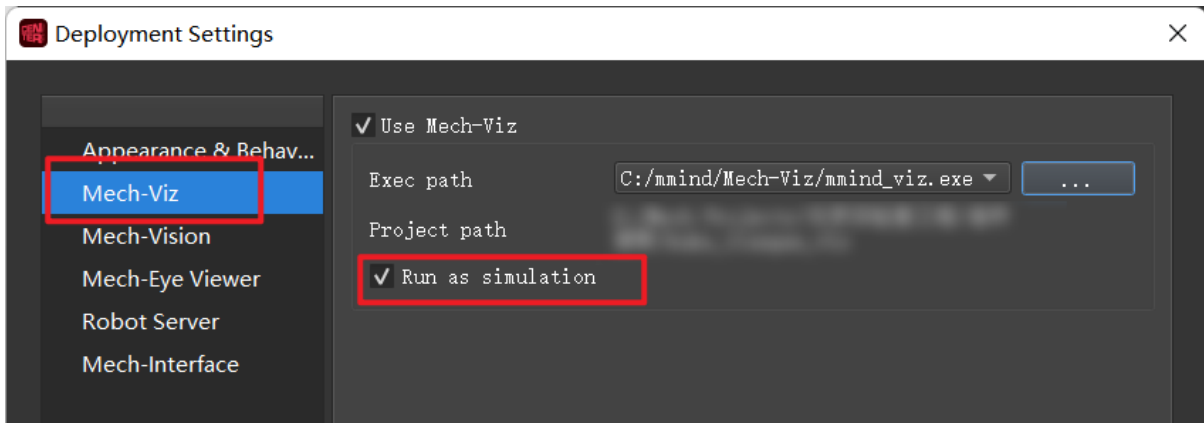
1. Adjust the robot's speed.

In the toolbar, adjust the **Robot Speed** to 5%.

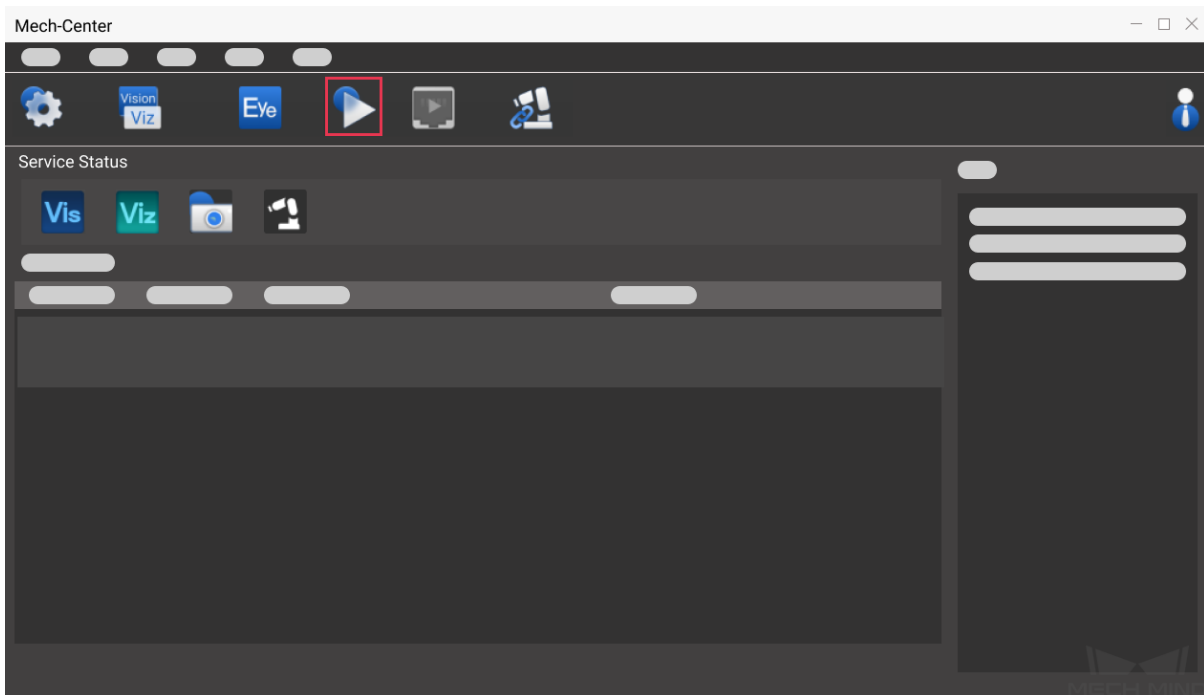
**Attention:** Be sure to turn down the robot movement speed!

2. Uncheck Run as simulation.

Mech-Center → Deployment Settings → Mech-Viz



3. Run



If the robot moves according to the planned trajectory and picks up and places the box, then you have successfully completed the first pick and place.

**Attention:** When running the robot, please ensure personnel safety. When an emergency occurs, please press the emergency stop button on the teach pendant!