



Dobot VX500 Smart Camera User Guide



Copyright © SHENZHEN DOBOT CORP LTD 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without the prior written consent of SHENZHEN DOBOT CORP LTD (hereinafter referred to as "Dobot").

Disclaimer

To the maximum extent permitted by applicable law, the products described (including its hardware, software and firmware, etc.) in this document are provided **AS IS**, which may have flaws, errors or faults. Dobot makes no warranties of any kind, express or implied, including but not limited to, merchantability, satisfaction of quality, fitness for a particular purpose and non-infringement of third party rights. In no event will Dobot be liable for any special, incidental, consequential or indirect damages resulting from the use of our products and documents.

Before using our product, please thoroughly read and understand the contents of this document and related technical documents that are published online, to ensure that the robot arm is used on the premise of fully understanding the robot arm and related knowledge. Please use this document with technical guidance from professionals. Even if follow this document or any other related instructions, damages or losses may happen in the using process. Dobot shall not be considered as a guarantee regarding all security information contained in this document.

The user has the responsibility to make sure following the relevant practical laws and regulations of the country, in order that there is no significant danger in the use of the robot arm.

SHENZHEN DOBOT CORP LTD

Address: Room 1003, Building 2, Chongwen Garden, Nanshan iPark, Liuxian Blvd,
Nanshan District, Shenzhen, Guangdong Province, China

Website: www.dobot-robots.com

Preface

Purpose

This document introduces the installation and usage of Dobot VX500 smart camera, which is convenient for users to understand and use the smart camera.

Intended audience

This document is intended for:

- Customer
- Sales Engineer
- Installation and Commissioning Engineer
- Technical Support Engineer

Compatible versions

Product	Version
VX500 plugin	V1.2.0
Calibration tool	V1.0.11 or above
DobotStudio Pro	V4.6.1.0-stable or above
SmartCamera firmware	V3.1.0_250326 or above
Controller CRA + CC262/CC263	V4.6.0.1-stable or above
Controller CRV + CC262V/CC263V	V4.5.4.1-stable or above
Controller CRV/CRA + CCBOX	V4.6.1.0-stable or above
Tool IO firmware	V6.5.0.2 or above

 NOTICE

Upgrading the camera firmware may cause compatibility issues. Older version solutions will not work with the new firmware version. Please proceed with caution and keep a record of any changes.

Related documents

Document	Description	Download link
DobotStudio Pro User Guide	Explains how to use the robot control software, DobotStudio Pro.	Dobot website
Dobot Robot Maintenance Tool User Guide	Guide to using the maintenance tool for Dobot robot firmware upgrades, controller file backup, and recovery.	Contact technical support for access.
Dobot CR A Series Hardware Guide	Introduces the functions, technical specifications, and installation instructions for Dobot CR A series collaborative robots.	Dobot website
Dobot CR V Series Hardware User Guide	Introduces the functions, technical specifications, and installation instructions for Dobot CR V series collaborative robots.	Contact technical support for access.

Revision history

Date	Version	Revised content
2025-04-08	V1.2.0	- Updated for VX500 plugin V1.2.0. - Added "Vision Application Cases" section.
2024-05-16	V1.1.1	- Added a note on prohibiting hot-plugging the camera. - Added a note on 2.5D positioning code size bound to the solution.
2024-03-29	V1.1.1	Updated for VX500 plugin V1.1.1.
2023-10-18	V1.0.1	Details optimization, updated for VX500 plugin V1.0.1.
2023-07-03	V1.0.0	The first release

Symbol conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
 DANGER	Indicates a hazard with a high level of risk which, if not avoided, could result in death or serious injury.
 WARNING	Indicates a hazard with a medium level or low level of risk which, if not avoided, could result in minor or moderate injury, robot arm damage.
 NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in robot arm damage, data loss, or unanticipated result.

Symbol	Description
 NOTE	Provides additional information to emphasize or supplement important points in the main text.

Contents

Preface	ii
1. Product Introduction	1
2. Hardware/Software Installation and Connection	3
2.1 Hardware installation and connection	3
2.2 Software installation and connection	5
3. Software Main Interface	8
4. Camera Settings and Parameters	9
4.1 Camera settings	9
4.1.1 Camera settings	10
4.1.2 Firmware upgrade	10
4.1.3 One-click restore	11
4.1.4 Camera direction	11
4.1.5 2.5D positioning code	12
4.2 Camera parameters.....	14
4.2.1 Photo trigger mode	14
4.2.2 Image parameter settings	15
4.2.3 Setting light source	17
4.2.4 Setting gamma	18
5. Solution	19
5.1 Creating a new solution	19
5.2 Running the solution	22
5.3 Managing solutions.....	23
6. Calibration and Coordinates	26
6.1 2.5D calibration and coordinate system.....	26
6.1.1 2.5D calibration.....	26
6.1.2 Common causes of calibration failure.....	39
6.1.3 Creating coordinate system based on 2.5D positioning code	40
6.2 2D calibration and coordinate system.....	43
6.2.1 2D calibration.....	43
6.2.2 Common causes of calibration failure.....	61
6.2.3 Creating coordinate system based on 2D template	61
6.3 Managing calibration file	66

6.4	Managing coordinate system	67
7.	Vision tool.....	68
7.1	Positioning tool	70
7.1.1	2.5D positioning	70
7.1.2	2D positioning.....	72
7.2	Recognition tool	73
7.2.1	Code recognition	73
7.2.2	Character recognition	75
7.2.3	Template recognition	77
7.2.4	Spot recognition.....	81
7.3	Measurement tool.....	84
7.3.1	Width measurement	84
7.3.2	Diameter measurement	86
7.3.3	Grayscale area	89
8.	Blockly/Script programming	92
8.1	Block description	92
8.2	Script commands	94
9.	Vision Application Cases	98
9.1	2.5D positioning case	98
9.1.1	Technical overview	98
9.1.2	Environmental requirements.....	99
9.1.3	2.5D positioning process	99
9.2	2D positioning case	113
9.2.1	Technical overview	113
9.2.2	Environmental requirements.....	114
9.2.3	2D positioning process	116
9.3	2.5D and 2D linkage positioning case.....	127
9.3.1	Technical overview	127
9.3.2	Linkage positioning Demo	128
Appendix A	Technical specifications	129

1. Product Introduction

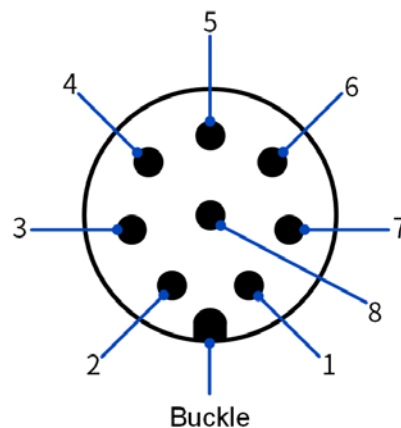
Dobot VX500 smart camera is compatible with all Dobot CR A / CR V series robots, featuring plug-and-play function. Equipped with an embedded high-performance vision system, it enables robots to recognize orientations, achieve precise positioning, and execute vision-based tasks, providing a universal Dobot vision solution.

Integrated with DobotStudio Pro, it simplifies on-site deployment, reduces debugging efforts, and delivers a more comprehensive and cost-effective option for vision inspection.

VX500 smart camera overview:



- Aviation plug:** Connects to the aviation socket at the robot's end effector for RS485 communication between the camera and the robot. The pins are defined as follows.



Pin	Definition	Pin	Definition
1	485A	5	24V output
2	485B	6	Digital output 2
3	Digital input 2	7	Digital output 1
4	Digital input 1	8	GND

- **Aviation socket:** When the end tool needs to communicate with the robot arm via an aviation plug, you can insert the aviation plug into this socket to establish communication. This socket type is the same as the aviation socket at the end of CRA series robots.

i NOTE

The camera communicates with the robot arm via RS485. It is recommended that tools connected to this socket use digital input/output (DI/DO) for communication with the robot arm. If the tool also uses RS485 for communication, data conflicts may occur, affecting performance.

- **Network interface:** Ethernet interface with a silicone protective cover. The camera needs to be connected to the computer via a network cable during the camera configuration, and can be used offline without a network connection when running a project.

i NOTE

For the connection between the robot arm and the computer, you need to use another network cable or WiFi, independent of this network interface.

- **Photo button and indicator:** When the "**Photo trigger mode**" is set to "**Photo button**", you can press the round button to capture an image. There are three indicators located near the button: PWR (power indicator), LNK (network indicator) and STS (status indicator). See the silkscreen markings on the camera for details.
- **Lens and light source:** Equipped with a camera lens, 14 LED lights, and 2 positioning indicators.
- **Result display indicator:** Displays the operation status (OK/NG).
- **Adapter flange:** Used to attach the camera to the robot arm's end.

2. Hardware/Software Installation and Connection

The VX500 smart camera follows an eye-in-hand installation method, meaning it is mounted at the end of the robot arm. The smart camera is pre-assembled with the adapter flange before delivery, so you only need to install it at the robot arm's end. The flange design conforms to ISO 9409-1, making it compatible with all Dobot CR A / CR V series robots.

2.1 Hardware installation and connection

1. Insert the aviation plug of the VX500 smart camera into the aviation socket at the robot arm's end.



NOTICE

Do not plug or unplug the camera while powered on. Before disconnecting or connecting the aviation plug, turn off the controller power and make sure that the indicators of the robot and camera are all off.

2. Secure the adapter flange to the robot arm's end using four M6 screws.
 - For 2D positioning, the camera must be installed in one of the following two orientations: Camera towards the robot base, Camera back towards the robot base.

Camera towards the robot base



Camera back towards the robot base



- When using the 2.5D positioning function only, the camera can be mounted at any angle. For example: Camera side-facing the robot base.

Camera side-facing the robot base

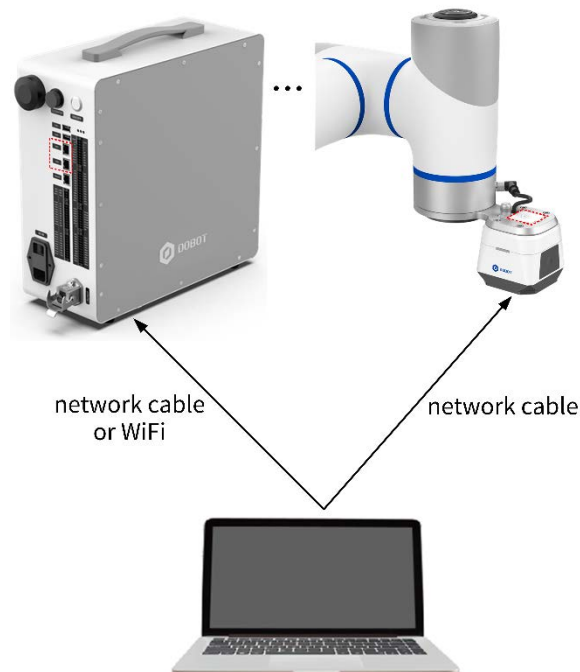
(The standard flange does not support side-facing installation. A custom flange is required.)



NOTICE

To connect the VX500 smart camera to CR20A robot, the following additional accessories are needed:

- 1) **M8-to-M12_8pin aviation adapter cable:** Used to connect the VX500 smart camera to CR20A robot.
- 2) **CRA & CR20A adapter flange:** Mounts onto the CR20A's end, adapting it to the standard CRA series flange.
3. Connect the VX500 smart camera to the computer using a network cable. The network cable is used to debug the camera and can be unplugged when you run the project.
4. Connect the robot controller to the computer using network cable or WiFi. The network cable or WiFi is used for the computer to control the robot arm through DobotStudio Pro.



2.2 Software installation and connection

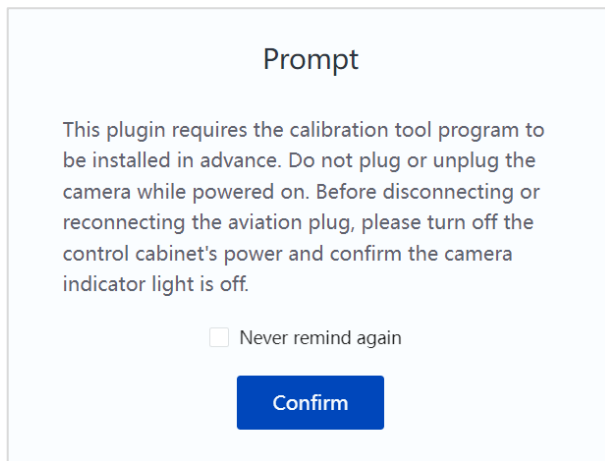
1. Please download the SmartCamera plugin package from [Dobot website](#) or contact Dobot technical support to obtain it, including the SmartCamera Plugin (VX500_v1-2-0), Calibration Program Installer (DobotCalibrateSetup.exe), and Camera Firmware Upgrade Package.
2. Double-click **DobotCalibrateSetup.exe** to install the calibration program (requires administrator permission). The installation path cannot be changed (default installation location: C drive). Once installed, **DobotCameraCalibrate** will automatically launch and run in the background when using the VX500 plugin.
3. Open DobotStudio Pro and connect to the robot. For specific procedure, see DobotStudio Pro user guide.
4. Open the Dobot+ page, **import** and **install** the VX500_v1-2-0 plugin.



NOTE

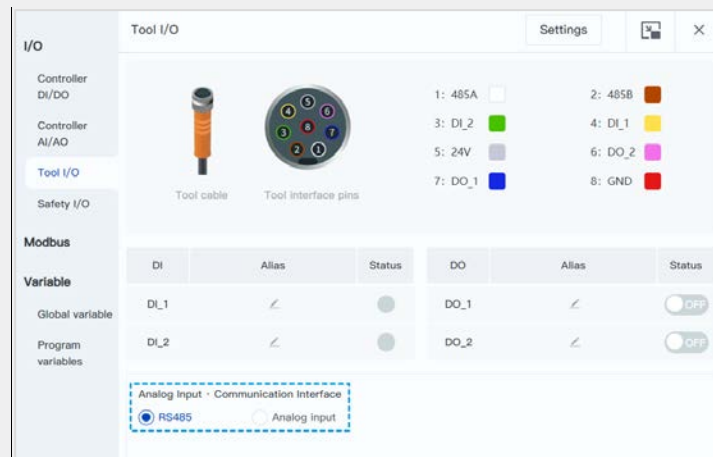
The VX500 plugin currently supports Simplified Chinese, Traditional Chinese, English, Japanese (日本語), Korean (한국어), German (Deutsch), Spanish (Español), and French (Français).

5. Open the VX500 plugin, the following pop-up window will appear. Click "**Confirm**" and wait for the plugin to detect the camera that has been connected to the computer via the network cable. Check "Never remind again" to prevent this prompt from appearing next time.

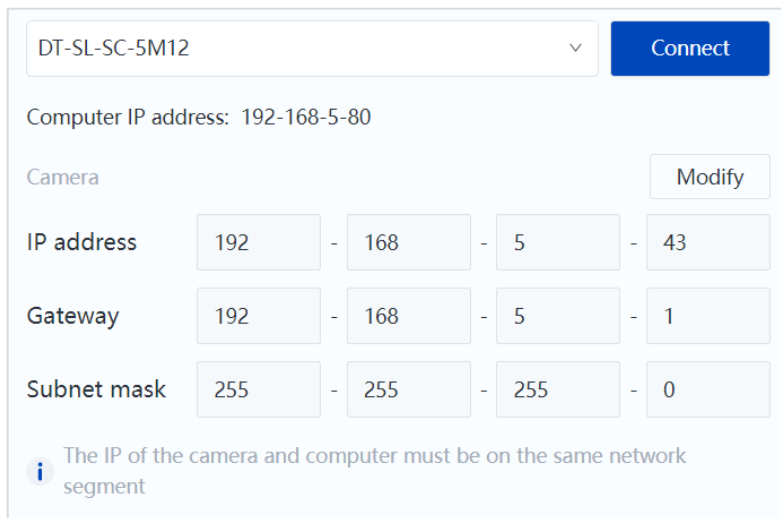


NOTE

- Disable the firewall on your computer before connecting the camera to avoid communication issues.
- Make sure that the robot's "**Analog Input · Communication Interface**" is set to "**RS485**" before connecting to the camera. You can verify and modify this setting in **Monitor > Tool I/O** in DobotStudio Pro.

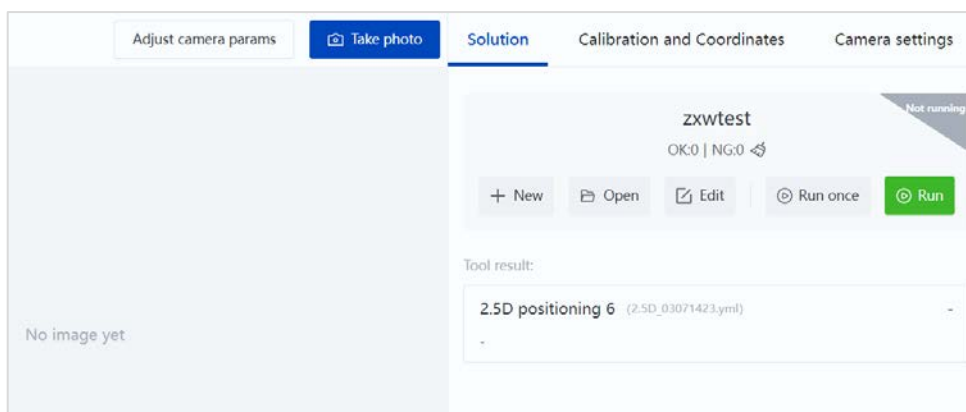


- Once the plugin detects the camera, both the computer's IP and the camera's IP will be displayed. Adjust either the camera IP or computer IP to ensure they are on the **same subnet** (same first three values, different last value). The camera may disconnect temporarily during this process, please be patient.



The screenshot shows an IP configuration interface. At the top, there's a dropdown menu showing 'DT-SL-SC-5M12' and a 'Connect' button. Below this, the 'Computer IP address' is '192-168-5-80'. Under the 'Camera' section, there's a 'Modify' button and three rows of IP configuration: 'IP address' (192-168-5-43), 'Gateway' (192-168-5-1), and 'Subnet mask' (255-255-255-0). At the bottom, a note states: 'The IP of the camera and computer must be on the same network segment'.

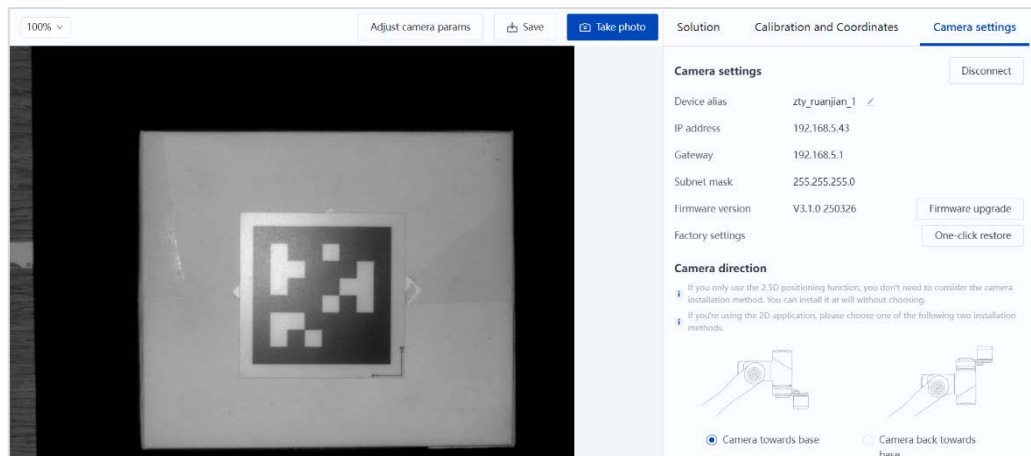
- Once the camera is detected again, click "**Connect**" and the plugin starts to connect to the smart camera. After a successful connection, the **Solution** page will display the current running solution of the camera.



 NOTE

For firmware upgrade procedures, refer to the [Firmware upgrade](#) section.

3. Software Main Interface



Left Panel: Camera Image Display and Controls

- The image area displays the last captured image from the camera upon connection.
- You can click "**Take photo**" on the top-right corner of the image area to take a photo (Ensure "**Photo trigger mode**" is set to "**Manual photography**", see [Camera parameters](#) for details).



NOTICE

If capturing fails, check if the firewall is disabled; or verify that the current running solution is compatible with the firmware version.

- When an image is displayed in the image area, a "**Save**" button will appear, allowing you to save the image locally.
- You can click "**Adjust camera params**" to open the "Camera parameters" page (see [Camera parameters](#) for details).
- You can adjust the image size by selecting the display scale from the drop-down list on the top-left corner of the page.

Right Panel: Camera Configuration and Vision Functions

- Camera settings:** Displays and allows modifications to basic camera settings. See [Camera Settings and Parameters](#) for details.
- Calibration and Coordinates:** Allows to manage the calibration files and coordinate systems. See [Calibration and Coordinates](#) for details.
- Solution:** Allows to manage the camera's operating solutions. See [Solution](#) for details.

4. Camera Settings and Parameters

4.1 Camera settings

Solution
Calibration and Coordinates
Camera settings

Camera settings
Disconnect

Device alias
zty_ruanjian_1

IP address
192.168.5.43

Gateway
192.168.5.1

Subnet mask
255.255.255.0

Firmware version
V3.1.0 250326
Firmware upgrade

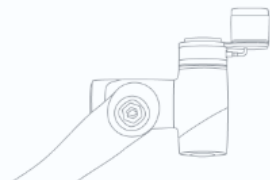
Factory settings
One-click restore

Camera direction

i If you only use the 2.5D positioning function, you don't need to consider the camera installation method. You can install it at will without choosing.

i If you're using the 2D application, please choose one of the following two installation methods.


☒ Camera towards base


☐ Camera back towards base

2.5D positioning code
i This value is used for calculations when using the 2.5D positioning code in the future

50mm x 50mm (Standard)

4.1.1 Camera settings

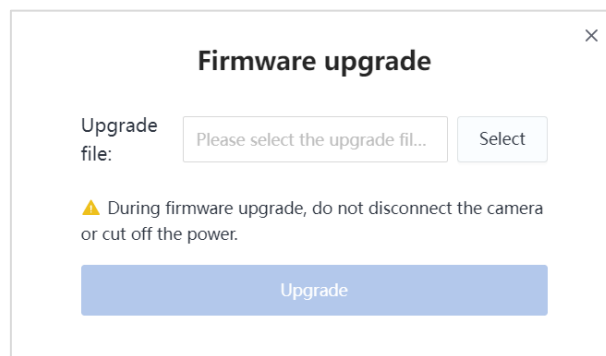
Device alias: Editable. You can set an alias to distinguish between different camera devices.

IP address/Gateway/Subnet mask: Displays the current network configuration of the camera.

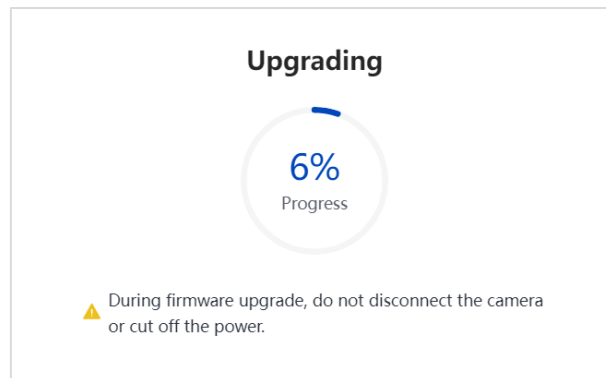
4.1.2 Firmware upgrade

Please check the firmware version of the VX500 smart camera. The firmware version must match the camera plugin version. See [Compatible versions](#) for details.

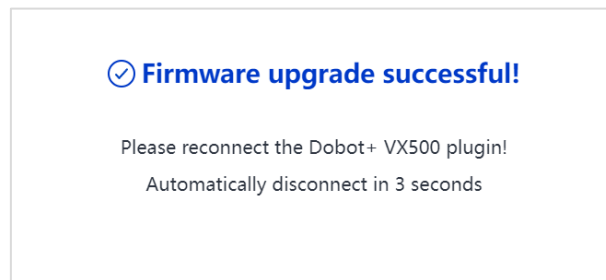
If the firmware version is lower than **V3.1.0_250326**, click the "**Firmware upgrade**" button, then select the upgrade file (with a ".dav" or ".bin" extension) in the pop-up window to proceed with the upgrade.



During the upgrade process, the progress will be displayed.



Once the upgrade is completed, a success message will appear, and the connection will automatically disconnect after 3 seconds.

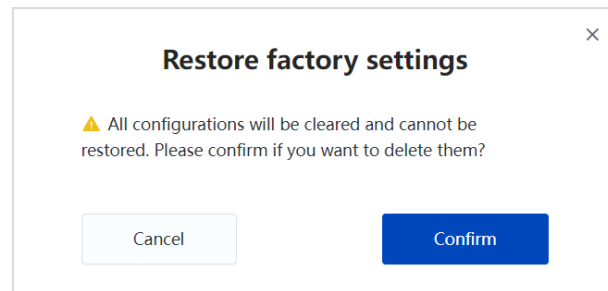


**NOTICE**

- The upgrade file path only supports Chinese and English.
- Do not disconnect the camera or power off the device during the upgrade process. If the upgrade fails due to network issues or human error, the camera may disconnect.
- If the target firmware version differs from the first two digits of the current version, previous firmware-dependent solutions may become incompatible. You may need to recreate solutions after the upgrade.
- To upgrade from V3.0.1 230625 (the initial version first-shipped cameras) to V3.1.0_250326, you must first upgrade to V3.0.6 240118, and then upgrade from V3.0.6 240118 to V3.1.0_250326.
- If your firmware version is V3.1.0 but an earlier date (e.g., V3.1.0 240116), you can upgrade directly to V3.1.0_250326.

4.1.3 One-click restore

After performing a firmware upgrade, you can use the **One-click restore** function to clear all configurations and prevent potential problems. This action is irreversible, so proceed with caution.

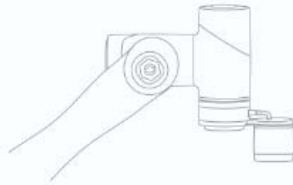


4.1.4 Camera direction

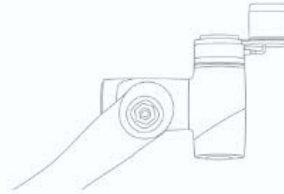
Select the appropriate camera direction based on its actual installation.

Camera direction

- i If you only use the 2.5D positioning function, you don't need to consider the camera installation method. You can install it at will without choosing.
- i If you're using the 2D application, please choose one of the following two installation methods.



☒ Camera towards base



☐ Camera back towards base

i NOTE

- To use the 2.5D positioning function, the camera must maintain the selected direction during calibration, but there are no restrictions when using the positioning function.
- To use the 2D positioning function, the camera must maintain the selected direction both during calibration and positioning.

4.1.5 2.5D positioning code

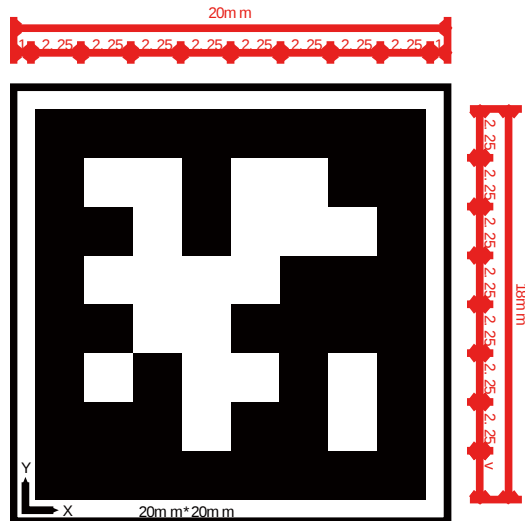
Dobot positioning codes are supplied with the following 3 sizes as standard. Measure the outer frame of the 2.5D positioning code in use and select the appropriate size in the "Camera settings" page.

- 1) 60mm x 60mm (Standard)
 - 2) 50mm x 50mm (Standard)
 - 3) 20mm x 20mm (Standard)
- The standard positioning codes are included as accessories with the VX500 smart camera, with sizes measured including the outer frame.

2.5D positioning code

i This value is used for calculations when using the 2.5D positioning code in the future

60mm x 60mm (Standard) ▾



- To use a custom 2.5D positioning code, select **"Custom"** and enter the dimensions (range: 10mm – 300mm).

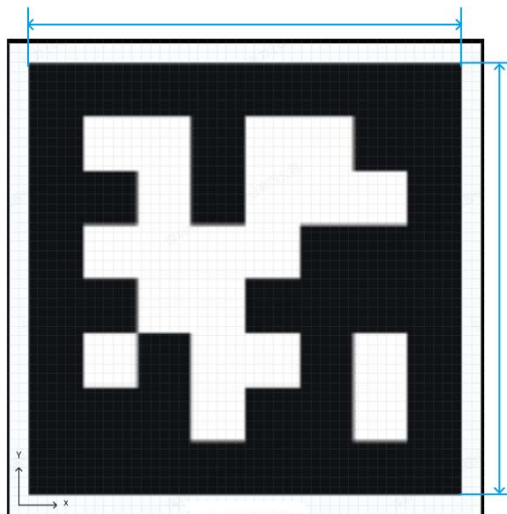
2.5D positioning code

This value is used for calculations when using the 2.5D positioning code in the future

Custom

50 mm x 50 mm

The size of the custom positioning code refers to the inner black area of the code, as shown in the figure below.



NOTE

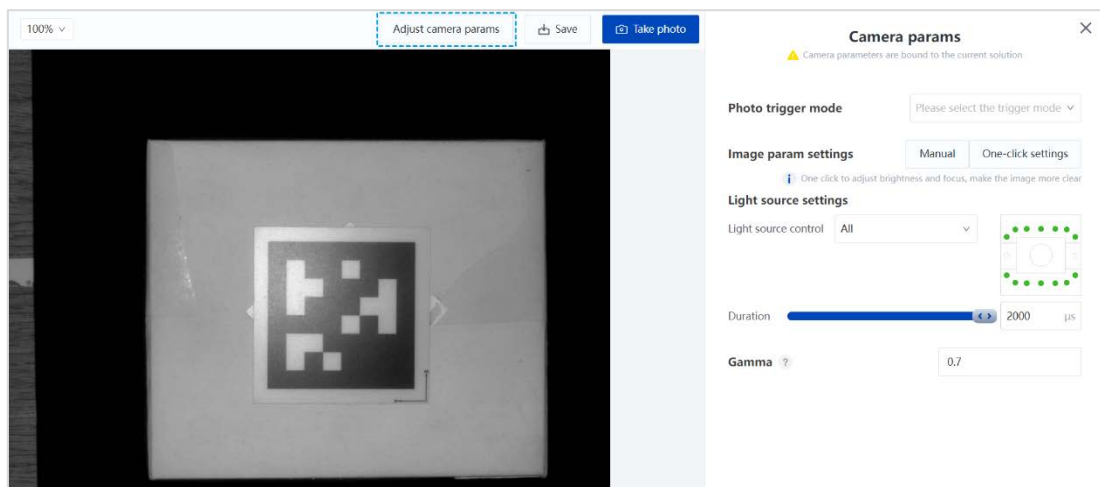
- To print the 2.5D positioning codes on your own, contact Dobot technical support to obtain the design files.
- The positioning code size is a global setting, meaning different-sized positioning codes cannot be mixed.
- Positioning codes of the same size may have multiple styles, each with a unique serial number (see [2.5D coordinate system](#) for details). Select the appropriate code according to your needs.

4.2 Camera parameters

You can click "**Adjust camera params**" above the image to open the "Camera parameters" page.

NOTE

Camera parameters are bound to the current solution. Any modifications will only apply to the current solution. (A new camera comes with a blank solution by default.)



4.2.1 Photo trigger mode

The smart camera supports the following trigger modes. Regardless of the selected mode, the auto-triggered capture function remains active when adjusting camera parameters.

- **Auto photography:** The camera automatically takes photos at regular intervals (non-real-time imaging).
- **Manual photography:** Click the "**Take photo**" button in the plugin to trigger image capture.
- **Camera button:** Press the physical button on the side of the camera to trigger image capture.

The trigger mode used when the smart camera communicates with the robot does not belong to any of the above modes, so this option will be left blank after automatic calibration or running a project. You can reconfigure it if needed.

4.2.2 Image parameter settings

One-click settings

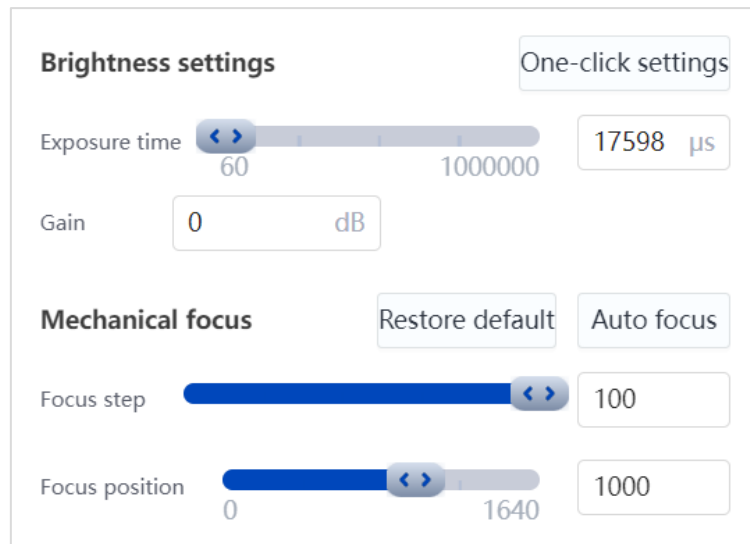
Click "**One-click settings**", and the camera will automatically adjust the camera parameters (brightness and focus) based on the surrounding environment.

The automatic adjustment process takes time, during which the camera cannot be operated. Please wait patiently.

If the image quality does not meet your requirements after one-click adjustment, you can manually fine-tune the settings.

Manual settings

Click "**Manual**" to open the manual settings page.



NOTE

Before clicking "**Manual**", you must first select a "**Photo trigger mode**", or a "Please select a trigger mode" message will appear.

Regardless of the selected trigger mode, after clicking "**Manual**", the camera will automatically switch to "**Auto photography**" (to provide real-time image feedback during manual adjustments). Once adjustments are complete, the camera will revert to the previously set trigger mode.

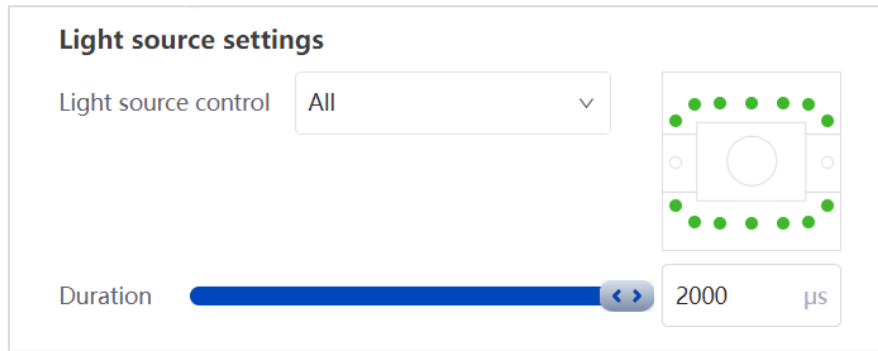
Brightness settings

- Click "**One-click settings**" to automatically adjust the brightness.
- If "**One-click settings**" does not meet your requirements, you can manually adjust "**Exposure time**" and "**Gain**" to modify the image brightness.

Mechanical focus

- Click "**Auto focus**", and the camera will automatically adjust the focus position in fixed focus steps, and ultimately select the focus position with the sharpest overall image.
- Click "**Restore default**" to reset the focus position to default settings.
- If the image lacks clear contrast, the **Auto focus** may fail. In such cases, please adjust the focus manually.
- **Focus step:** Set the step size for focus adjustments.
- **Focus position:** If **Auto focus** does not meet your requirements, you can modify the focus position manually.

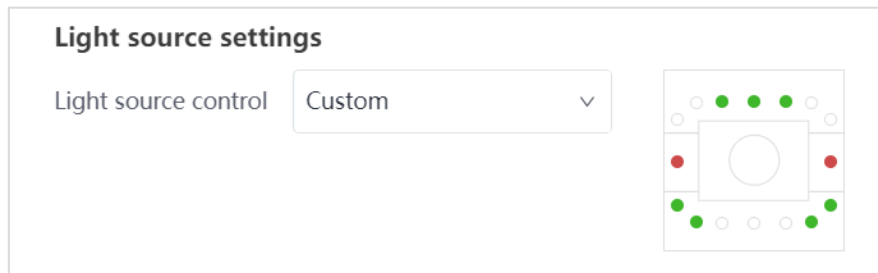
4.2.3 Setting light source



- **Light source control:** Select the mode for light source control, including **All**, **Custom** and **None**.
 - **All:** Turns on all LEDs. The **Duration** setting applies to all LEDs.
 - **Custom:** Allows independent control of the LEDs in each zone (click the corresponding area in the diagram on the right, Green = On, White = Off) and the duration.

When switched to "**Custom**" mode, all LEDs are "**Off**" by default.

The LEDs on both sides of the lens serve as "device aimer" to indicate the camera's field of view. You can click the corresponding LEDs to turn the device aimer on or off (Red = On, White = Off).



- **None:** Turns off all LEDs, and the **Duration** setting becomes invalid.
- **Duration:** Configures how long the LEDs remain on after being triggered. It can be configured only when the "**Light source control**" is "**All**", but also applies to the LEDs turned on in "**Custom**" mode. If all LEDs are turned off and saved, reopening the camera settings will reset the duration to 100 μs (minimum value).

4.2.4 Setting gamma

You can manually adjust the **Gamma** value to fine-tune image contrast.

Gamma between 0.5 – 1.0: Enhances brightness in darker areas.

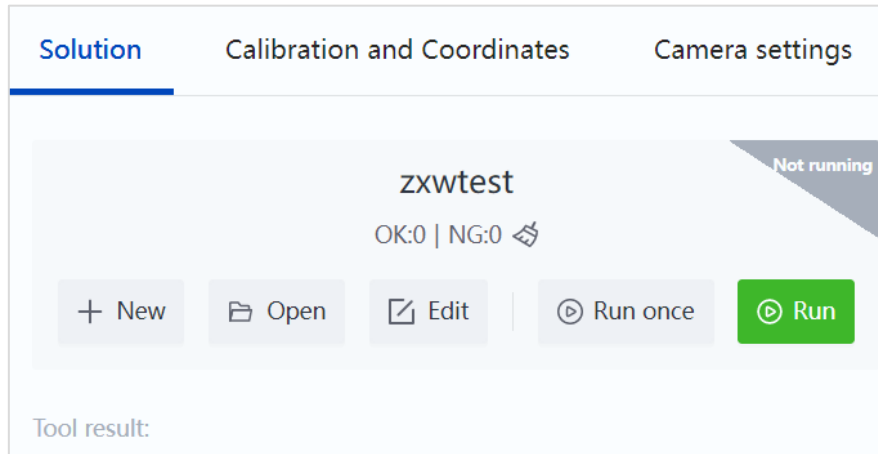
Gamma between 1.0 – 4.0: Reduces brightness in darker areas. Factory default: 0.7.

Gamma ?

0.7

5. Solution

A solution is a collection of vision tools, and multiple tools can be added to a single solution. The VX500 smart camera comes with a default blank solution that contains no tools or configurations. You need to create a new solution based on your actual needs.

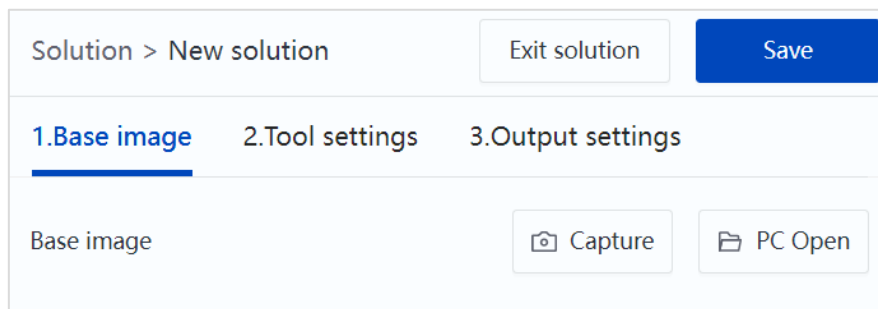


NOTICE

If the current solution running on the camera was created before a firmware upgrade, please exit and delete it, then create a new solution.

5.1 Creating a new solution

1. Click "**New**" to enter the new solution creation page.
2. Set the **base image**. All tools except the positioning tool require an ROI (Region of Interest), which is mainly used for defining detection and template areas in the VX500 smart camera plugin.



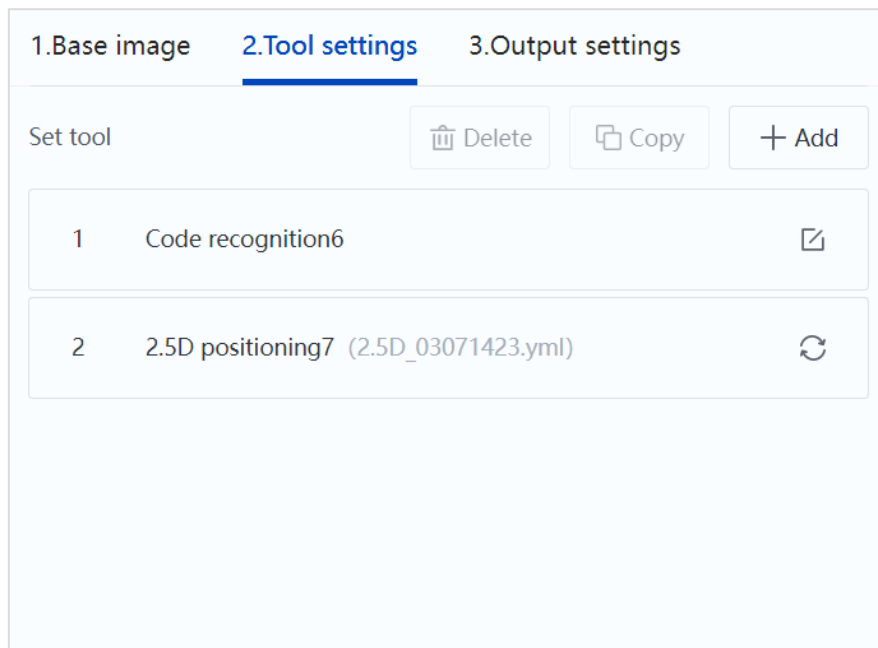
- If no base image has been set, click "**Capture**" to take a new photo as the base image. If a base image is already set, the "**Capture**" button changes to "**Replace**", you can click it to update the base image.
- Click "**PC Open**" to import an image from the computer as the base image.

NOTE

Make sure that the base image imported from the PC is in JPG, BMP, or PNG format and matches the camera's current resolution.

It is recommended to import an image taken by the smart camera and saved locally.

- Go to the **"Tool settings"** tab and click **"Add"** to add a new tool. For detailed descriptions of each tool, refer to the [Vision Tools](#) section. To use a positioning tool, please complete the hand-eye calibration and coordinate system setup first, as described in the [Calibration and Coordinates](#) section.



After selecting the added tool:

- You can click **"Delete"** to delete the tool.
- You can click **"Copy"** to duplicate the tool.

The maximum number of tools per solution is 40, but each tool has its own limit. For example, 2.5D positioning and code recognition modules allow only one tool per solution. If the camera memory reaches the limit, additional tools cannot be added even if the total number is below 40.

- You can click  next to the 2.5D positioning tool to modify the calibration file bound to the tool.
- 2D positioning tool cannot be edited.
- You can click  next to other tools to modify their configurations.

- Go to the **"Output settings"** tab, enter a solution name, and select a solution status. Solution status:
 - All Tools OK:** The solution result is OK only when the module status of all tools in the solution is OK.
 - Any Tool OK:** The solution result is OK if the module status of at least one tool in the solution is OK.

1.Base image
2.Tool settings
3.Output settings

Solution name

Solution status
All Tools OK

Tool output result

Tool	Output result	Operation
No result yet, please firstConfigure		

- Click "**Configure**" to open the tool output settings page. Select the output results for the added tools (multiple selections allowed), click "**Confirm**", and the selected output results will be output, which can be viewed in the interface or called via programming.

Positioning tool

☒ 2.5D positioning 7

Recognition tool

☐ Code recognition 6

☒ Module status
☒ Coordinate system

- Once you have finished creating the solution, click "**Save**" in the upper-right corner to save and return to the "Solution" page. Click "**Exit solution**", and a confirmation prompt will appear. Click "**Confirm exit**" to return to the "Solution" page.

⚠ All current edits will be temporarily saved and can be restored to their original state by changing the solution or restarting the camera.

Cancel

Confirm exit

5.2 Running the solution

Solution
Calibration and Coordinates
Camera settings

0402
OK

OK:1 | NG:0

+ New
Open
Edit
Run once
Run

Tool result:

2.5D positioning 7 (2.5D_03071423.yml) exist is ok | OK

Module status: OK

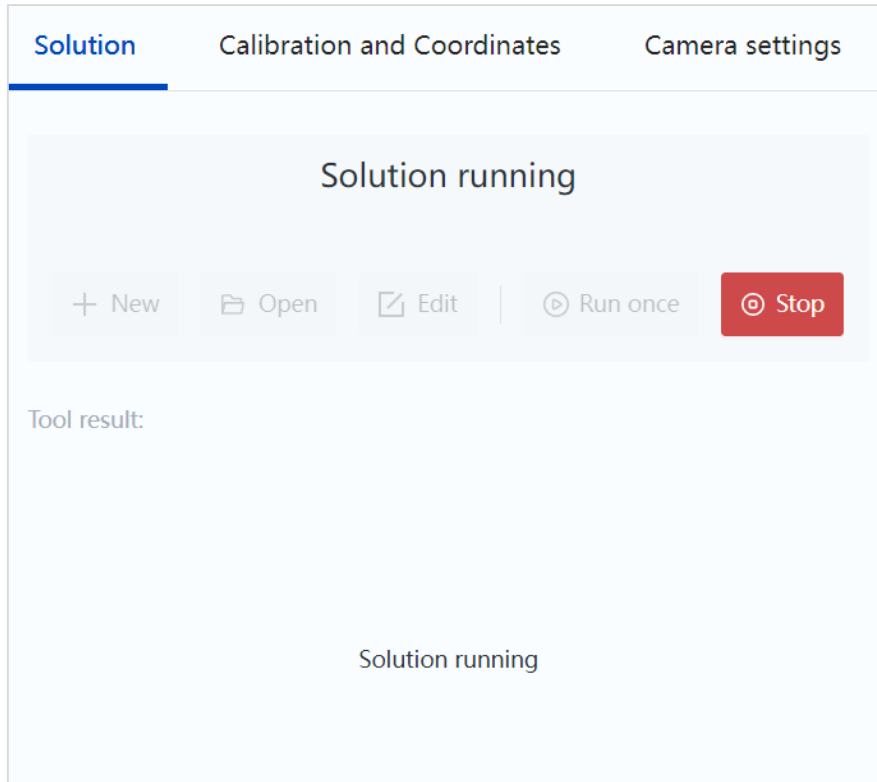
Coordinate system: (179.3553772, -630.7700195, -107.0478516, 2.1647179, -1.0753...

Code recognition 6 Not Exist is OK | OK

Module status: OK

Code count: 0

- Click **"Run once"** on the solution page to execute the current solution once (take a photo and process it). The result will be displayed.
 - A status icon (OK/NG) next to the solution name will indicate the result of this run.
 - Below the solution name, the counter displays the number of the current solution's running results. You can click  to clear the count.
 - "Tool result"** displays the individual results of each tool within the solution for this run.
- Click **"Run"** to continuously execute the solution. While running continuously, results will not be displayed.

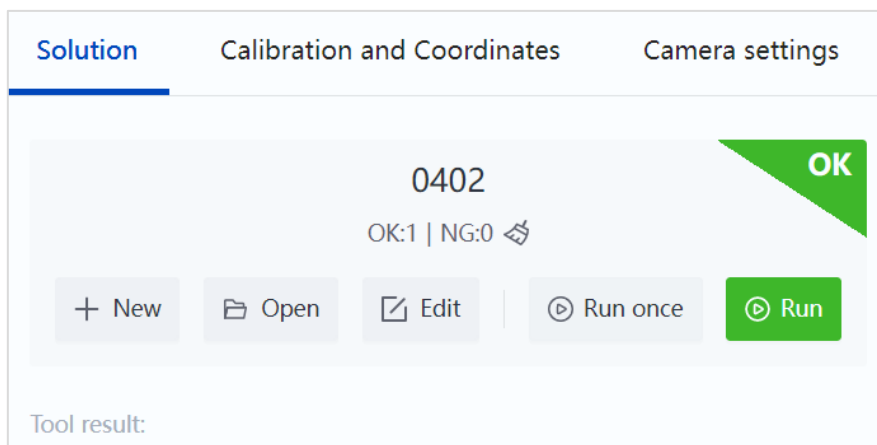


After running the solution, you can refer to [Blockly/Script programming](#) and [Vision Application Cases](#) to develop your program.

NOTE

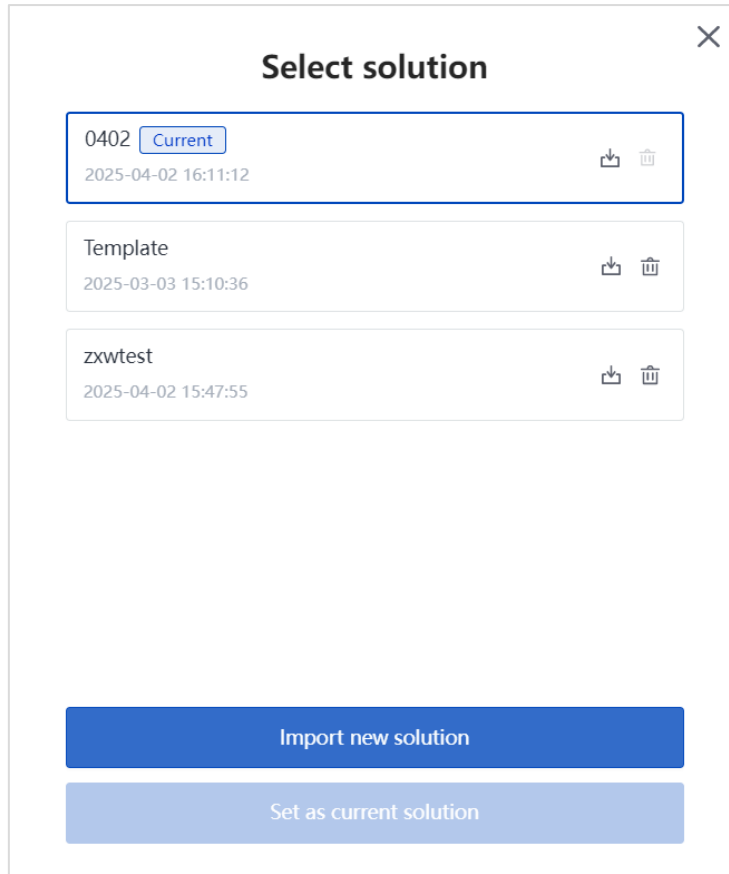
The solution must be executed at least once to ensure that it functions properly within the program.

5.3 Managing solutions

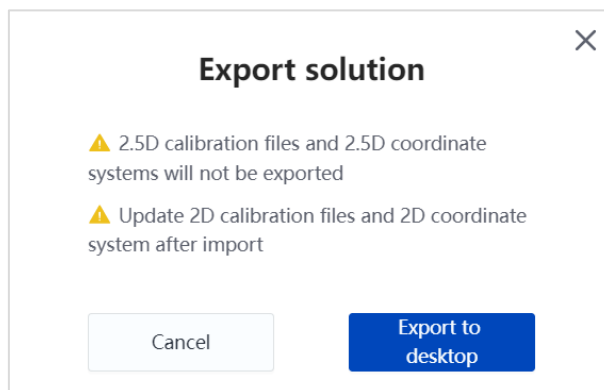


Click "**Edit**" to modify the current solution.

Click "**Open**" to display the solution list, as shown below:



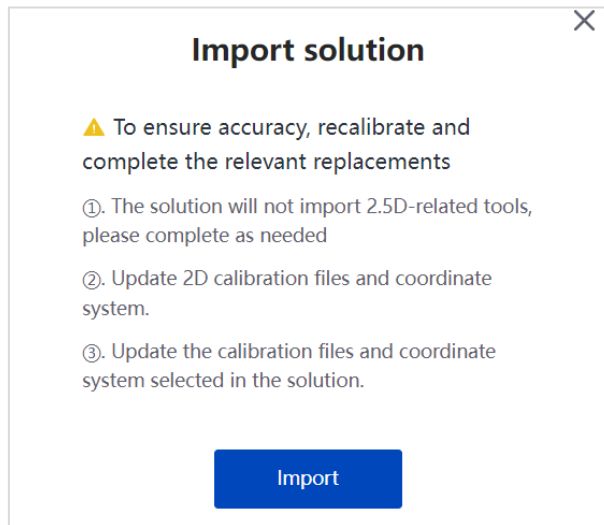
- Select a solution and click "**Set as current solution**" to apply it as the active solution.
- Click  next to a solution to remove it. The current solution cannot be deleted.
- Click  next to a solution will pop up the "**Export solution**" prompt. Click "**Export to desktop**" to save the selected solution to the desktop with the filename format "Solution_Time.VX500".



NOTE

2.5D calibration files and 2.5D coordinate systems will not be exported.

- Select **"Import new solution"** will pop up the following prompt, click **"Import"** to select the file path to save the solution.

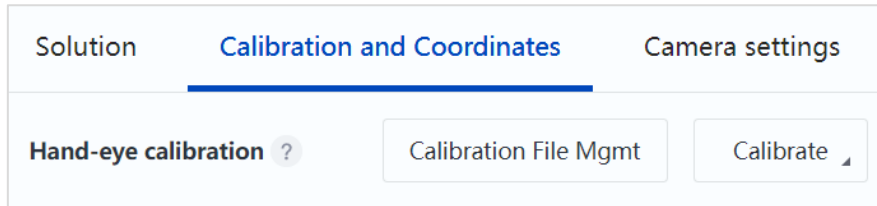


i NOTE

1. 2.5D-related tools will not be imported when importing a solution, please configure as needed.
2. After importing, update the 2D calibration files and 2D coordinate systems (invalid 2D coordinate systems will be marked in red and labeled "Failure_User").
3. Imported solution files must have the ".VX500" extension.

6. Calibration and Coordinates

You can perform hand-eye calibration on "**Calibration and Coordinates**" page.



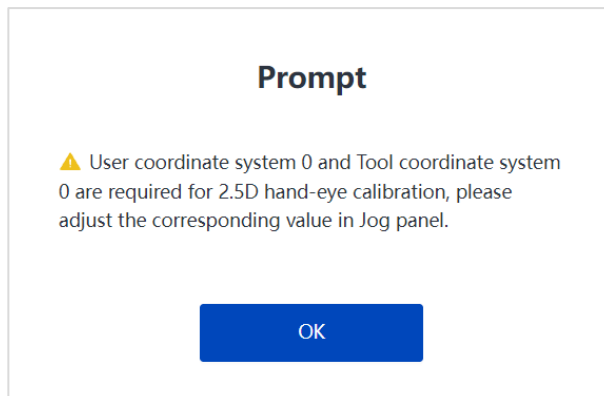
Hand-eye calibration determines the transformation between the **physical coordinate system** of the robot's end-effector (hand) and the **image coordinate system** of the camera (eye). This allows image-based coordinates to be converted into physical coordinates for robot motion.

6.1 2.5D calibration and coordinate system

6.1.1 2.5D calibration

To ensure accuracy, the following conditions must be met during 2.5D calibration:

1. The **user coordinate system** and **tool coordinate system** on the Jog panel must be set to 0, otherwise the following popup will appear:



2. Keep the joint angle of J3 positive when setting the calibration-related points.

2.5D auto calibration

2.5D automatic calibration: After you teach the center point of the calibration board and the initial photo point, the robot automatically generates 25 calibration points, moves to each point, captures images, and then generates a calibration file based on the captured images.

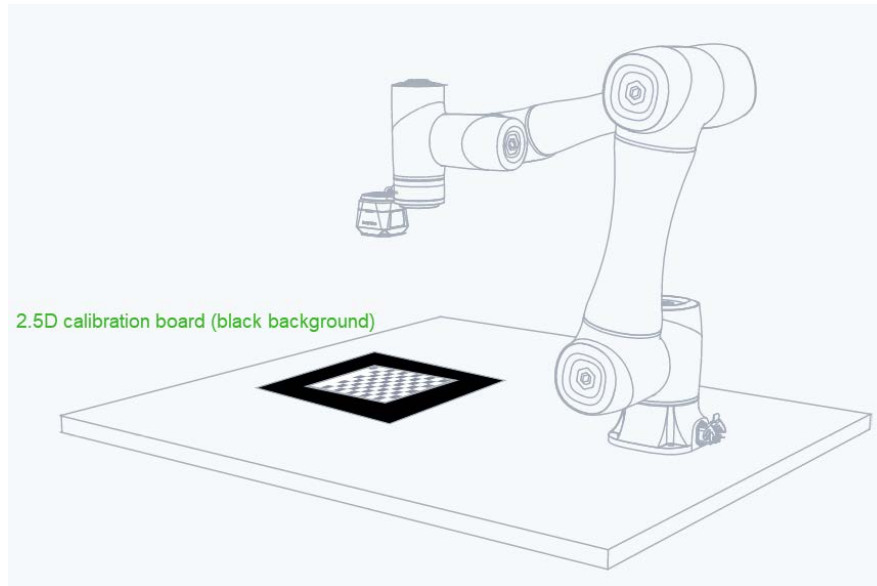
Ensure that there are no obstacles within the robot's motion range when using this calibration method.

Steps for 2.5D automatic calibration:

Step 1: Place the calibration board

Place the 2.5D calibration board directly below the camera, aligning the long edges of the board with the long edges of the camera's field of view and the short edges of the

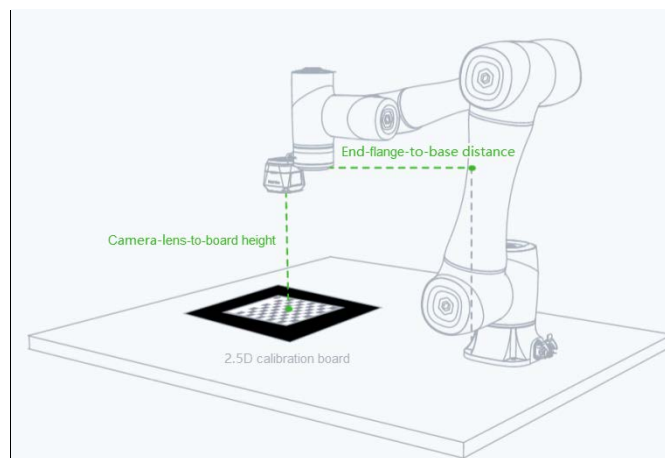
board with the short edges of the camera's field of view. A black or white background is recommended.



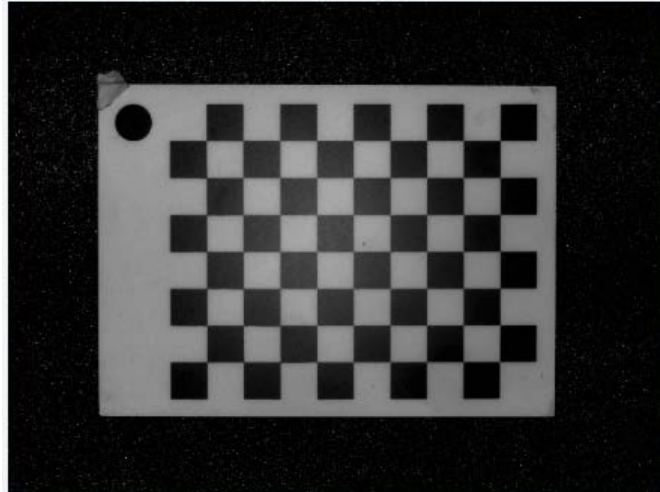
Step 2: Adjust the robot's posture

Adjust the robot's posture so that the camera lens is parallel to the calibration board. Ensure the calibration board is centered in the camera's field of view and occupies approximately 70% of the view. The recommended camera height from calibration board is 300mm, with recommended angles of $RX = \pm 180^\circ$, $RY = 0^\circ$.

Diagram



Camera field of view

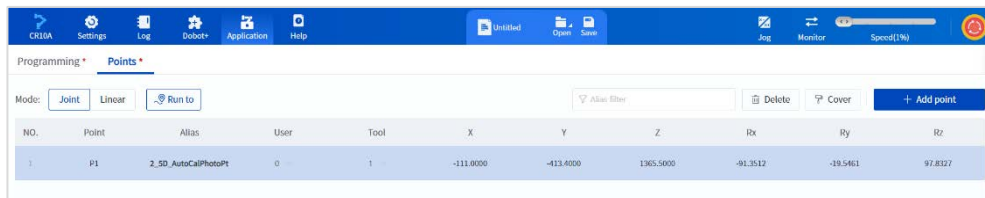


NOTE

For CR10A, the distance from the end of the arm to the robot base should be 500mm. For CR5A, this distance should be 350mm.

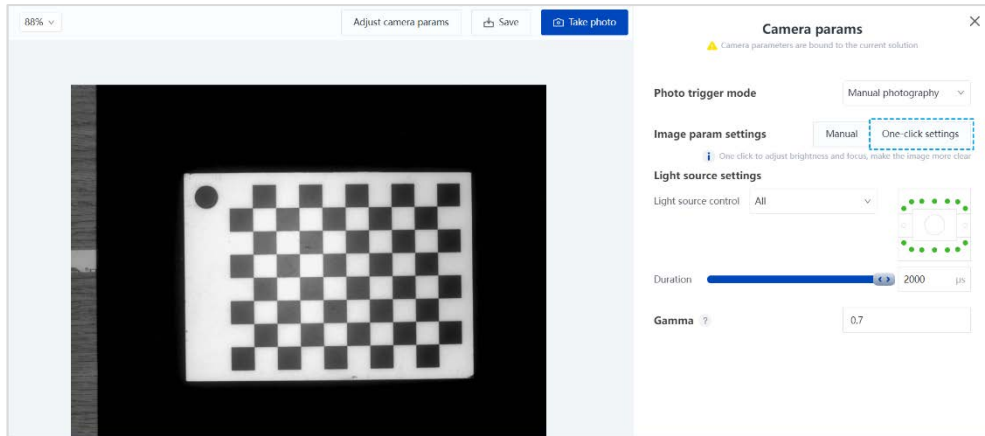
Step 3: Save photo point

- ① Open "DobotStudio Pro > Application > Blockly programming", then create a new project (self-define the name).
- ② Go to the "Points" page, click "Add point" to save the current points (recommended alias: "2_5D_AutoCalPhotoPt").



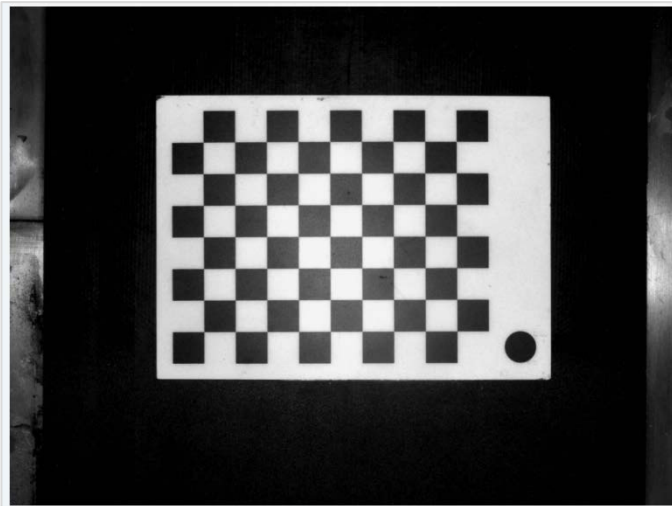
Step 4: Adjust camera parameters

- ① Go back to Dobot+ plugin page and click "Adjust camera params" to enter the [Camera parameters](#) setting interface.
- ② Click the "One-click settings" button to auto-focus. Please wait for the process to complete.



Calibration board image quality:

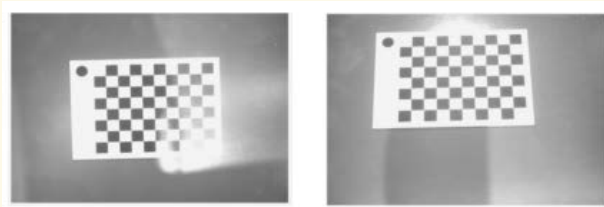
The chessboard edges must be sharp, with strong contrast between black and white. Overexposure must be avoided, as it can cause corner points to separate and affect detection.



NOTICE

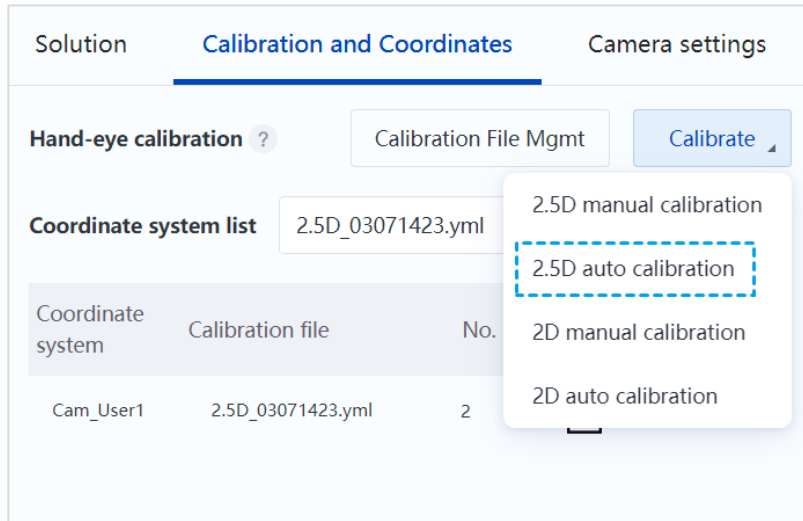
Avoid overexposure!

Overexposure occurs when the exposure time is too long, making the image excessively bright and causing it to appear washed out.

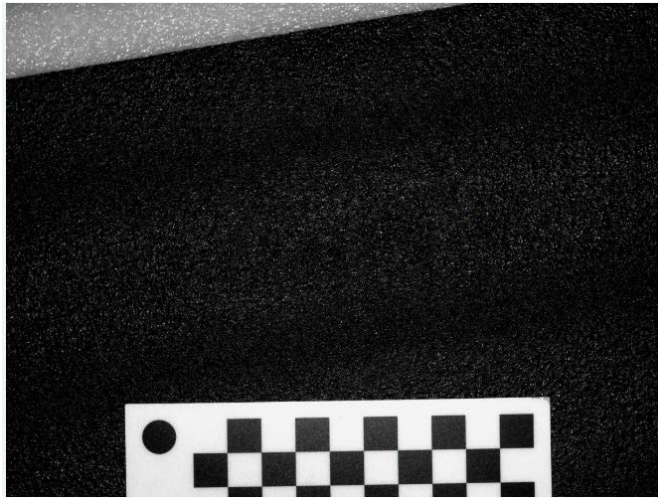


Step 5: Set center point

- ① Go to "Calibration and Coordinates > Calibrate > 2.5D auto calibration".



- ② Adjust the robot arm so that the flange center aligns with the center of the 2.5D calibration board, ensuring an error of less than 10mm (no height requirement). This can be verified by capturing an image of the calibration board. For example: At 300mm camera-to-board distance, the captured image should display two rows of grid squares. As shown in the figure below.

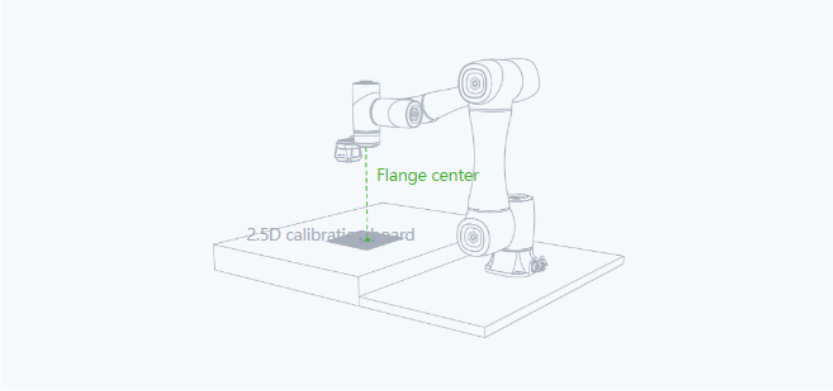


- ③ Click "Obtain center point" on the "2.5D auto calibration" page.

2.5D auto calibration

① **Set center point**
② **Set initial photo point**
③ **Collect calibration point**

Place the 2.5D calibration board horizontally (align the long side of the board with the long side o...[View more](#))



Set center point
● Not obtained

Obtain center point

Cancel

Next

- ④ Click "Next" to proceed to the "Set initial photo point" page.

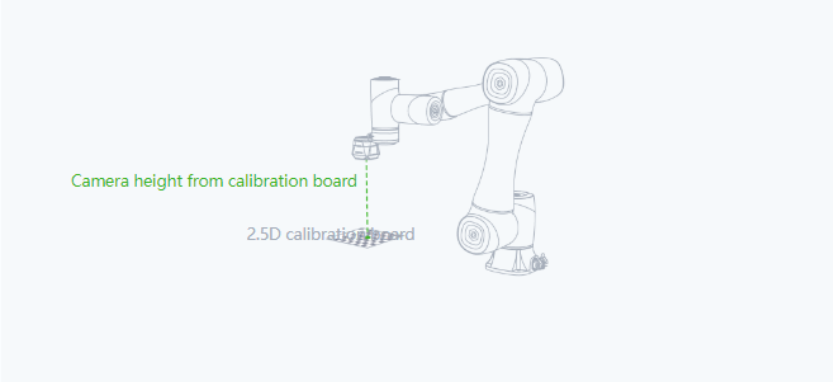
Step 6: Set initial photo point

- ① Go to the "Points" page, select the saved "2_5D_AutoCalPhotoPt", and long-press "Run to" to move the robot to the corresponding point.
- ② Go to the "Set initial photo point" page in the Dobot+ plugin.

2.5D auto calibration

① Set center point
② Set initial photo point
③ Collect calibration point

Adjust the robot arm's posture to keep the camera lens parallel to the 2.5D calibration board. Ens... [View more](#)



Camera height from calibration board(±50mm error)

mm

Initial photo point ● Not obtained

📍 Obtain coordinates

Cancel
Last step
Next

- ③ Measure the height of the camera lens from the calibration board and enter it into the input field, then click "Obtain coordinates". Once the coordinates have been successfully obtained, click "Next" to proceed to the "Collect calibration point" page.

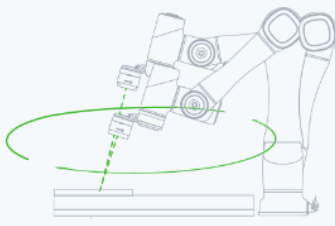
Step 7: Collect calibration point

Ensure that there are no obstacles within the robot's motion range, then click "Start collecting". The robot will automatically move to 25 points for image capture. Please wait for the process to complete.

2.5D auto calibration

① Set center point
② Set initial photo point
③ Collect calibration point

① Please ensure there are no obstacles within the motion range of robot arm.
 ② After starting the collection, the robot arm will automatically move to the 25 points generated around the calibration board. Take back ^

No obstacles around


Collect calibration point
● (0/25)
Start collecting

Last step
Save calibration file
Cancel

Step 8: Calibration result

After calibration, the result will be displayed. You need to determine whether recalibration is necessary based on the calibration result and error data.

2.5D auto calibration

① Set center point
② Set initial photo point
③ Collect calibration point

① Please ensure there are no obstacles within the motion range of robot arm.
 ② After starting the collection, the robot arm will automatically move to the 25 points generated around the calibration board. Take
back ^

Collect calibration point
● (25/25)
Re-collect

Result
● Excellent

Average translation error (mm): X: 0.200; Y: 0.105; Z: 0.197;
 Max translation error (mm): X: 0.522; Y: 0.347; Z: 0.406;
 Average rotation angle error (°): Rx: 0.105; Ry: 0.175; Rz: 0.015;
 Max rotation angle error (°): Rx: 0.205; Ry: 0.356; Rz: 0.039;

File name

.yaml

Last step
Save calibration file
Cancel

- If the calibration result is **Excellent**, enter a filename (supports letters, numbers, and underscores) and save the calibration file.
- If the calibration result is **Good**, review the error data and decide whether to recalibrate or save the calibration file based on the required calibration accuracy.
- If the calibration result is **Failed**, adjust the settings and repeat the calibration process.

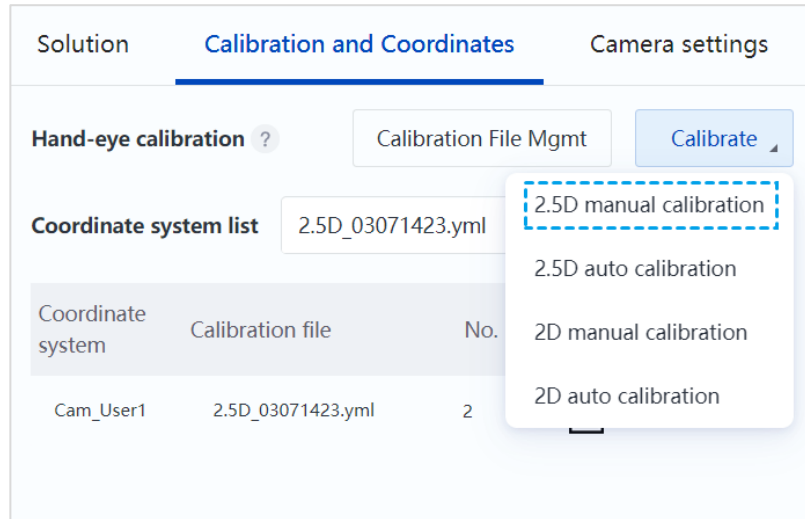
Once the calibration file is saved, the system will automatically return to the "Calibration and Coordinates" page. 2.5D calibration file is bound to the robot. If the robot is replaced, recalibration is required.

2.5D manual calibration

If automatic calibration cannot be performed due to workspace constraints (e.g., obstacles within the robot's motion range), 2.5D manual calibration can be used. 2.5D manual calibration allows users to manually define 20 calibration points, ensuring the camera captures a clear and complete 2.5D calibration board from different distances and angles. The user must manually move the robot arm to each calibration point and take a photo. The system then generates a calibration file based on the captured images.

Step 1:

Click "**Calibrate > 2.5D manual calibration**" to enter the "2.5D manual calibration" page.

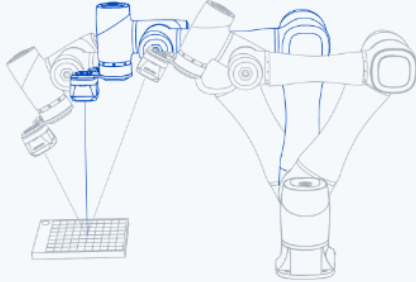


Step 2: Collect calibration point

2.5D manual calibration

1 Collect calibration point
2 Hand-eye calibration

Initial calibration point: Place the 2.5D calibration board horizontally (align the long side of the ... [View more](#))



Collect calibration point
● (0/20)

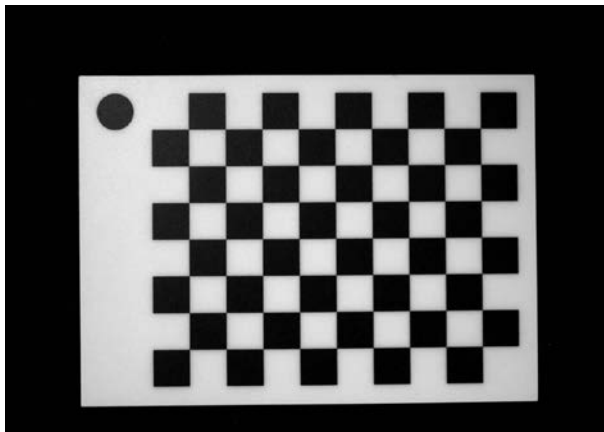
Collect (0/20)

Cancel

Next

Initial calibration point

- ① Place the 2.5D calibration board horizontally, aligning the long edges of the board with the long edges of the camera's field of view. A black or white background is recommended, as shown in the figure below.



- ② Adjust the robot's posture so that the camera lens is parallel to the calibration

board. Ensure the calibration board is centered in the camera's field of view and occupies 70% of the view. The recommended camera height from calibration board is 250mm – 350mm, with recommended angles of $RY = 0^\circ$, $RX = \pm 180^\circ$ (camera towards base) or 0° (camera back towards base).

- ③ [Adjust the camera parameters](#) for a clear image, then click "Collect" to record the initial calibration point.

Additional calibration points

- ① Adjust the robot's posture so that it significantly changes: Any X/Y/Z value changes by more than 20mm. Any RX/RX/RZ angle changes by more than 3° .
- ② Click "Collect" to record the calibration point. Repeat the above steps until all 20 calibration points have been collected. Then click "Next" to proceed to the "Hand-eye calibration" page.



NOTE

Once the initial calibration point has been collected, camera parameters cannot be adjusted. Ensure the calibration board remains clear and within the field of view by adjusting the robot's posture. The robot must remain stationary while collecting (taking photos).

Step 3: Hand-eye calibration

You need to determine whether recalibration is necessary based on the calibration result and error data.

2.5D manual calibration

① Collect calibration point
2 Hand-eye calibration

Result
● Good
Recalibrate

Average translation error (mm): X: 0.609; Y: 0.417; Z: 0.321;

Max translation error (mm): X: 1.872; Y: 1.798; Z: 1.110;

Average rotation angle error (°): Rx: 0.066; Ry: 0.069; Rz: 0.024;

Max rotation angle error (°): Rx: 0.181; Ry: 0.262; Rz: 0.122;

File name

2.5D_
.yaml

Cancel

Last step

Save calibration file

6.1.2 Common causes of calibration failure

If calibration fails due to excessive error (greater than 2mm), follow these steps to troubleshoot:

1. Check that the camera and robot arm are securely installed.
2. Check that the calibration board remains fixed relative to the robot base.

Example: When performing calibration with a robot arm mounted on an AGV, ensure the AGV remains stable and does not shake or move.

3. Check that the robot arm is calibrated using a positive posture ($J3 > 0$); otherwise, significant errors may occur.
4. If calibration still fails, it may be due to poor accuracy of the robot arm itself. Consider replacing the robot arm or recalibrating it.

If the system fails to collect calibration points, check whether the calibration board is out of the camera's field of view or if camera parameters are incorrectly set. Try turning off the light source and adjusting the exposure time to ensure a clear image.

6.1.3 Creating coordinate system based on 2.5D positioning code

Solution	Calibration and Coordinates	Camera settings
Hand-eye calibration ? Calibration File Mgmt Calibrate ▾		
Coordinate system list 2.5D_03071423.yml ▾ Create coordinate		

Step 1: Select calibration file

Select a previously saved 2.5D calibration file from the "Coordinate system list" dropdown menu, then click "Create coordinate".

Step 2: Move to photo point

Create coordinate

1 Move to photo point

2 Set user coordinate system



How does camera take a clear photo ?

Set photo point

☐ Not obtained

Obtain photo point

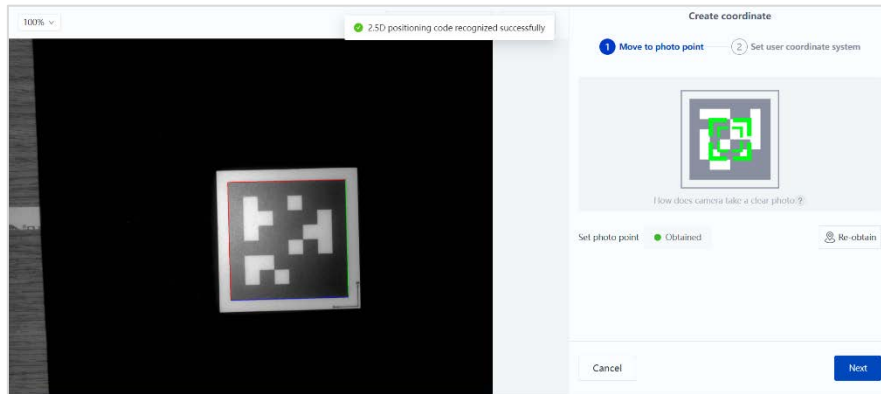
Cancel

Next

- ① Secure the 2.5D positioning code on a flat surface.
- ② Move the robot arm so that the camera lens is parallel to the 2.5D positioning

code, ensuring that the code is centered in the image. The recommended height from the front of the camera lens to the calibration board is 190mm.

- ③ Refocus the camera automatically. Once the image is sharp and properly exposed, click "Take photo" to detect the 2.5D positioning code. If "2.5D positioning code recognized successfully" pops up, a detection box will appear around the code.
- ④ Click "Next" to proceed to the "Set user coordinate system" page.



NOTE

1. Before taking photos, the **User coordinate system** and **Tool coordinate system** on the Jog panel must be set to "0".
2. It is recommended to save this point for later use in programming.

Step 3: Set user coordinate system

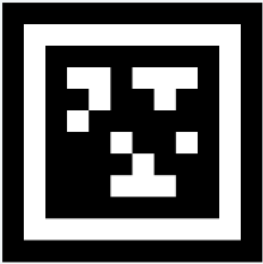
Based on the captured image, the system will recognize the serial number of the 2.5D positioning code. Select the corresponding user coordinate system name, then click "Finish" to save.

If coordinate system recognition fails, manually adjust the exposure time to optimize image brightness.

Create coordinate

1 Move to photo point

2 Set user coordinate system



Positioning code

Positioning code S/N

2

Set as user coordinate system:

Cam_User2

Cancel

Last step

Finish

Once the coordinate system is saved, you will return to the "Calibration and Coordinates" page, where the newly created coordinate system and associated calibration board details will be displayed. You can click the calibration board diagram to view a larger image.

6.2 2D calibration and coordinate system

6.2.1 2D calibration

To ensure accuracy, when performing 2D calibration, it is recommended to set the calibration-related points while keeping the J3 joint angle positive.

2D auto calibration

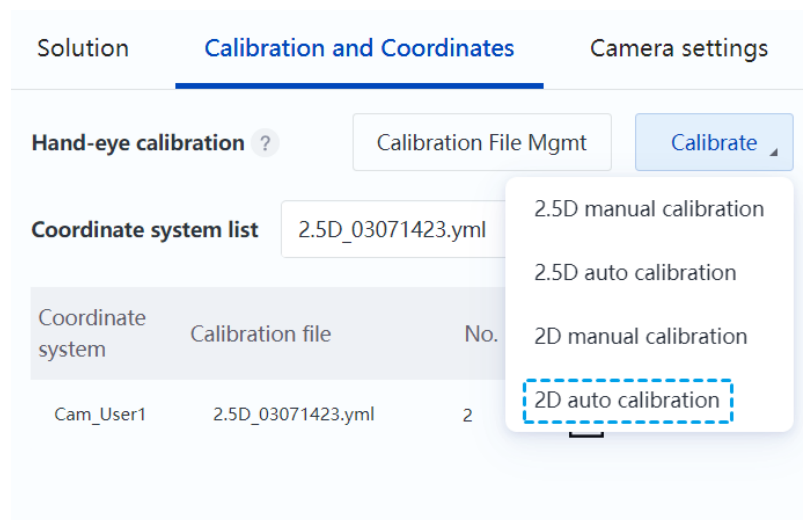
2D automatic calibration: After setting the photo point, template, and offset, the robot automatically generates 14 calibration points and moves to each point sequentially to capture both image coordinates and physical coordinates. A calibration file is then generated.

Ensure that there are no obstacles within the robot's motion range when using this calibration method.

Steps for 2D automatic calibration:

Step 1:

Click "**Calibrate > 2D auto calibration**" to enter the "2D auto calibration" page.



Step 2: Set photo point

2D auto calibration

① Set photo point — ② Set template — ③ Set offset — ④ Collect calibration point

Please create a new project in "Application" page and create a new photo point in "Points" page. ... [View more](#)



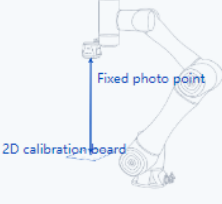
OK
Clear and complete

Fig 1



NG
Unclear or incomplete

Fig 2



Fixed photo point
2D calibration board

Fig 2

Set photo point
☐ Not obtained

Obtain photo point

Cancel

Next

Adjust photo point

- ① Create a new project in the **"Application"** page (Blockly programming/Script programming), and create a new point in the **"Points"** page (recommended alias: "2D_PhotoPoint"). Manually set RX to 180° / -180° and RY to 0°, then move the robot to this point, ensuring that the camera lens is parallel to the workplane (the surface where objects will be detected).
- ② Adjust the robot's X, Y, Z, and RZ so that the detection area occupies about 90% of the camera's field of view (as shown in the figure below).



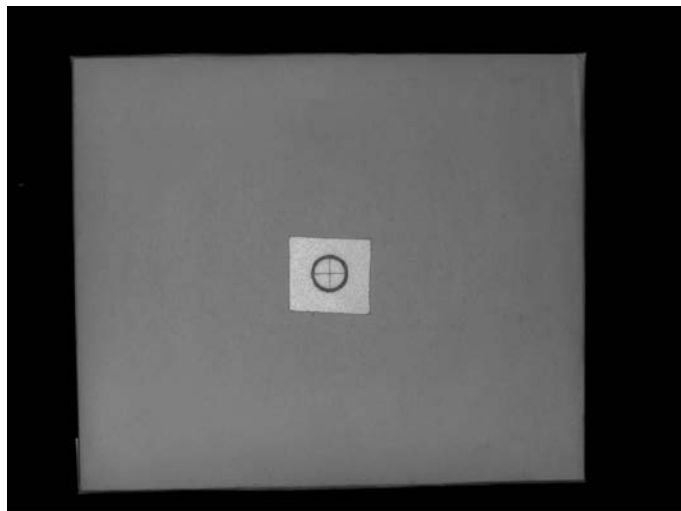
- ③ You can create a user coordinate system based on the workplane and store the "2D photo point" within this coordinate system, but ensure uniform user coordinate settings for subsequent points.
- ④ Overwrite the current point to "2D photo point".

NOTE

2D calibration only supports taking photos at fixed point. When using this calibration file later, it is essential to ensure that the robot arm is always at the specified photo point for every image.

Place 2D calibration board

- ① Place the 2D calibration board on the workplane. Cover it with white paper, leaving only one circle visible, and ensure that the circle is centered in the camera's field of view.



- ② [Adjust the camera parameters](#) for a clear image, then click "Obtain photo point". Then click "Next" to proceed to the "Set template" page.

NOTE

If the captured image is blurry during 2D calibration, stop the calibration process. Adjust the camera parameters to obtain a clear image, then restart the calibration. **Once calibration is completed, do not adjust the focus parameters.**

Step 3: Set template

2D auto calibration

1 Set photo point

2 Set template

3 Set offset

4 Collect calibration point

Select the template area and ensure the green lines accurately outline the template contour. If th...[View more](#)

A

B

C

Detection range

Detection area

Mask area

Edit

Search settings ?

Template area

Template mask area

Template matching point

Template sensitivity

Edit

Edit

Manual

Auto

Low

Middle

High

Extended params

Min score

Angle range

Match polarity

Algorithm timeout

50

-45

45

Consider polarity

0

ms

Set template

Make sure the calibration board is centered in the camera's field of view, then perform the following operations:

- ① Click the "Edit" button next to the "Template area" to display the editing tools



. Then use the rectangle tool to select the central circle of the calibration board, ensuring the green outline accurately fits the template shape.



NOTE

If the template is not detected accurately, switch the "Template sensitivity" to "Manual" mode and adjust the "Scale" and "Grayscale threshold" based on the image conditions.

- ② Click  next to the "Template matching point", then select the crosshair intersection at the center of the circle.



- ③ It is recommended to set the "Min score" (minimum score) to 70 to prevent detection failures due to excessive rotation during actual detection. Adjust the angle range to -45° 45° .
- ④ Click "Set template" to complete the setup.
- ⑤ Click "Next" to proceed to the "Set offset" page.

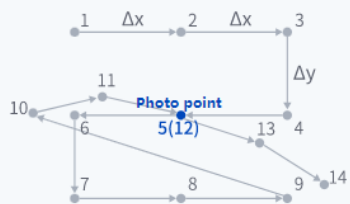
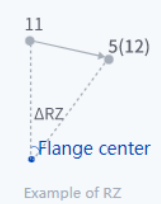
Step 4: Set offset

2D auto calibration generates 14 points automatically based on the offset you set, with the photo point as the center. The first 9 points are translation points (along the X and Y axes), and the remaining 5 points are rotation points (rotated around the RZ axis, distributed differently depending on the rotation angle). Point 5 and Point 12 share the same coordinates. The 14 points should cover 1/2 to 2/3 of the camera's field of view.

2D auto calibration

① Set photo point —
 ② Set template —
 ③ Set offset —
 ④ Collect calibration point

Take the photo point as the center, the robot arm will automatically generate 14 points based on the offset. It is recommended that the distribution of the 14 points occupies 1/2 to 2/3 of the vision field.

X-direction offset

60
mm

Y-direction offset

40
mm

RZ angle offset (Current range: -20°-20°)

15
°

i When the camera is at a height of 280mm from the calibration board, the long side of the image should be parallel to the X direction of the robot arm. The recommended value for ΔX is 60mm, ΔY is 40mm, and ΔRZ is 15°.

Cancel

Last step

Next

i NOTE

The template's angle range affects the RZ angle offset range:

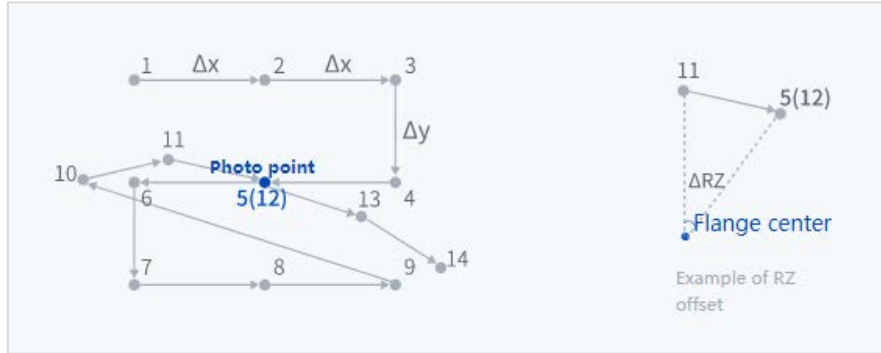
$RZ \text{ angle offset} < (\text{Template angle range} - 5^\circ) / 2$.

Since the template angle range is $-45^\circ - 45^\circ$, the RZ offset range is $-20^\circ - 20^\circ$.

• Recommended values

Please make sure that all 14 template points are within the camera's field of view and that there are no collisions with obstacles or the robot body during motion.

When the camera is 280mm above the calibration board, the recommended **X-direction offset** is 60mm, **Y-direction offset** is 40mm, and **RZ angle offset** is 15°.



• Operation tips

Move the robot arm along the XY plane at the photo point and click "Take photo". When the calibration object appears in the upper-left corner of the view, the absolute difference between the new XY values and the initial photo point XY values is the XY offset. Starting from the initial position, rotate until the calibration object reaches the edge of the field of view, compare the new RZ value with the initial RZ value, and take 1/3 of the absolute difference as the RZ angle offset.

Once setup is completed, click "Next" to proceed to the "Collect calibration point" page.

Step 5: Collect calibration point

Ensure that there are no obstacles within the robot's motion range, then click "Start collecting". The robot will automatically move to 14 points. Please wait for the process to complete.

2D auto calibration

① Set photo point
② Set template
③ Set offset

4 **Collect calibration point**

Ensure there are no obstacles within the motion range of robot arm. After starting the collection, the robot arm will automatically move to the 14 points generated by the system to complete the collection.



Collecting...

Please ensure there are no obstacles within the motion range of robot arm.

Collect calibration point

(0/14)

Pause collecting

Cancel

Last step

Save calibration file

Step 6:

After collection is completed, review the calibration results and on-site accuracy requirements to determine whether to finalize the calibration.

2D auto calibration

① Set photo point
② Set template
③ Set offset

4 Collect calibration point

Ensure there are no obstacles within the motion range of robot arm. After starting the collection, the robot arm will automatically move to the 14 points generated by the system to complete the collection.

Collect calibration point

● (14/14)

Result

● Good

Translation error (px):0.957926

Rotation error (px):2.2730179

File name

2D_user0_ .yaml

Cancel

Save calibration file

- If the calibration result is **Excellent**, enter a filename (supports letters, numbers, and underscores) and save the calibration file. The calibration file name follows the format "2D_Reference coordinate system when calibration is completed_Custom name", e.g. "2D_User1_CalFile1".
- If the calibration result is **Good**, review the error data and decide whether to recalibrate or save the calibration file based on the required calibration accuracy.
- If the calibration result is **Failed**, adjust the settings and repeat the calibration process.

Once the calibration file is saved, the system will automatically return to the "Calibration and Coordinates" page. 2D calibration file is bound to the working height. If the working height changes, recalibration is required.

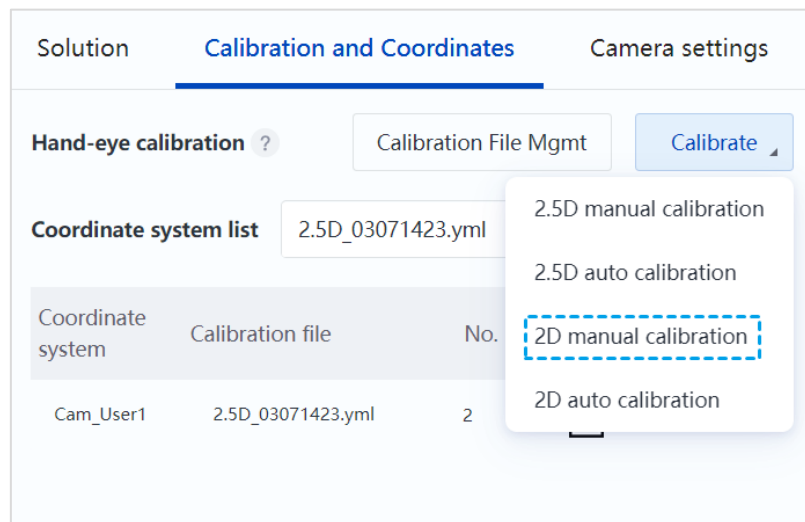
2D manual calibration

If automatic calibration cannot be performed due to workspace constraints (e.g., obstacles within the robot's motion range), 2D manual calibration can be used. Additionally, when using 2D positioning, if the target's angle varies significantly, 2D manual calibration may achieve higher accuracy (provided that the template matching point is precisely aligned using a needle tip during manual calibration).

2D manual calibration involves collecting 9 calibration points on a captured image using template matching, then guiding the robot arm to each point to record the coordinates, ultimately generating a calibration file.

Step 1:

Click "**Calibrate > 2D manual calibration**" to enter the "2D manual calibration" page.



Step 2: Set photo point

2D manual calibration

① Set photo point
② Set template

③ Collect calibration point
④ Obtain point coordinates

⑤ Hand-eye calibration

Please create a new project in "Application" page and create a new photo point in "Points" page. ... [View more](#)



OK
Clear and complete

Fig 1



NG
Unclear or incomplete

Fig 2

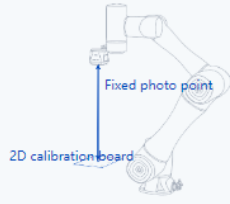


Fig 2

Set photo point
☒ Not obtained

Obtain photo point

Cancel

Next

Adjust photo point

- ① Create a new project in the **"Application"** page (Blockly programming/Script programming), and create a new point in the **"Points"** page (recommended alias: "2D_PhotoPoint"). Manually set RX to 180° / -180° and RY to 0°, then move the robot to this point, ensuring that the camera lens is parallel to the workplane (the surface where objects will be detected).
- ② Adjust the robot's X, Y, Z, and RZ so that the detection area occupies about 90% of the camera's field of view (as shown in the figure below).



- ③ You can create a user coordinate system based on the workplane and store the "2D photo point" within this coordinate system, but ensure uniform user coordinate settings for subsequent points.
- ④ Overwrite the current point to "2D photo point".

 NOTE

2D calibration only supports taking photos at fixed point. When using this calibration file later, it is essential to ensure that the robot arm is always at the specified photo point for every image.

Place 2D calibration board

- ① Place the 2D calibration board on the workplane, ensuring that it is centered in the camera's field of view (if you previously performed 2D automatic calibration and covered the outer 8 circles, remove the coverings before proceeding).
- ② [Adjust the camera parameters](#) for a clear image, then click "Obtain photo point". Then click "Next" to proceed to the "Set template" page.

 NOTE

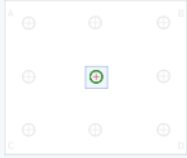
If the captured image is blurry during 2D calibration, stop the calibration process. Adjust the camera parameters to obtain a clear image, then restart the calibration. Once calibration is completed, do not adjust the focus parameters.

Step 3: Set template

2D manual calibration

- ① Set photo point —
- ② Set template** —
- ③ Collect calibration point —
- ④ Obtain point coordinates —
- ⑤ Hand-eye calibration

Select the template area and ensure the green lines accurately outline the template contour. If th...[View more](#)



Search settings ?

Template area

Template mask area

Template matching point

Template sensitivity

Edit

Edit



Manual

Auto

Low

Middle

High

Extended params

Min score

Angle range

Match polarity

Algorithm timeout

—



ms

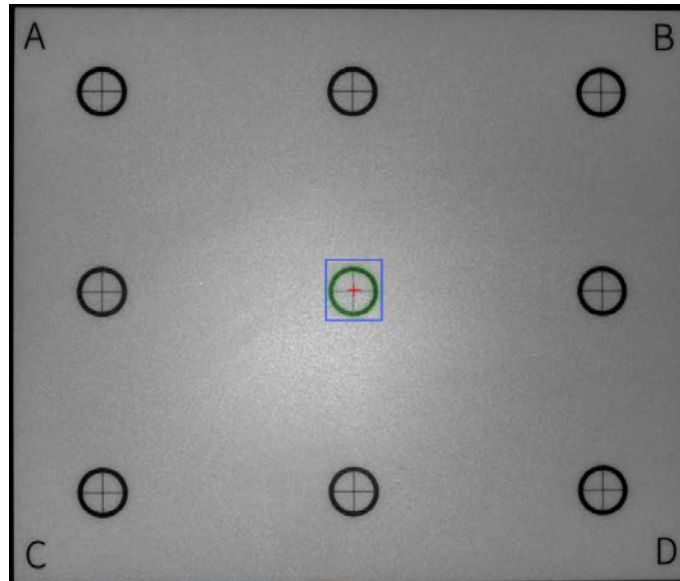
Set template

Make sure the calibration board is centered in the camera's field of view, then perform the following operations:

- ① Click the "Edit" button next to the "Template area" to display the editing tools

  . Then use the rectangle tool   to select any circle on the calibration board, ensuring the green outline accurately fits the template shape.

- ② Click  next to the "Template matching point", then select the crosshair intersection at the center of the circle. For more details on advanced template settings, refer to the [Template matching](#) section.
- ③ Click "Set template" to complete the setup.
- ④ Click "Next" to proceed to the "Collect calibration point" page.



Step 4: Collect calibration point

2D manual calibration

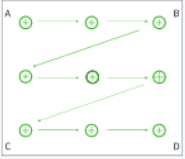
① Set photo point
② Set template

③ Collect calibration point

④ Obtain point coordinates

⑤ Hand-eye calibration

Frame the calibration point on the left base image using the rectangle or polygon tool, and click "Collect" af...

Framing order diagram (any order)

Detection area

Collect calibration point 0/9

Edit

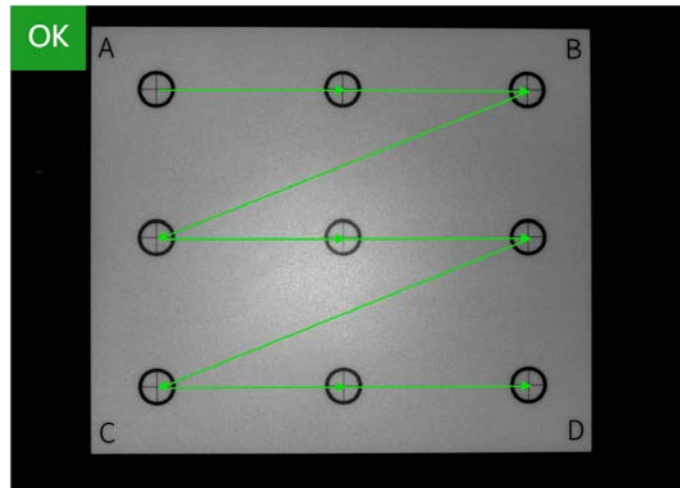
Collect calibration point

Cancel

Last step

Next

- ① Click "Edit" next to the "Detection area" and draw a rectangular selection around a single circle in the base image.
- ② Click "Collect calibration point", and the system will detect the point using template recognition.
- ③ Repeat the above steps until all 9 calibration points have been collected.
- ④ Click "Next" to proceed to the "Obtain point coordinates" page.



i NOTE

When collecting the 9 calibration points, make sure that the current user coordinate system on the Jog panel is consistent with the user coordinate system at the photo point. If inconsistent, a pop-up warning will appear: "The entire 2D hand-eye calibration process requires a unified user and tool coordinate system. Please adjust the corresponding values in the Jog panel".

Prompt

⚠ For the entire 2D hand-eye calibration process, ensure the user and tool coordinates are consistent. Please adjust the corresponding values in the Jog panel.

OK

Step 5: Obtain point coordinates

2D manual calibration

① Set photo point

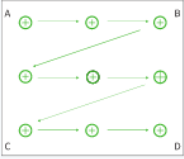
② Set template

③ Collect calibration point

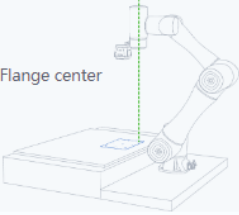
④ Obtain point coordinates

⑤ Hand-eye calibration

Keep the position of calibration board unchanged, move the robot arm to align the center of the ... [View more](#)



Calibration order diagram (same as the calibration point's collection order)



Flange center

Point coordinates

● 1 / 9

Re-obtain

Obtain

Cancel

Last step

Next

- ① Make sure the calibration board remains in the same position as in the base image, move the robot arm so that the flange center aligns with the first calibration point (for precise alignment, a calibration pin or similar tool may be used), then click "Obtain".
- ② Obtain the coordinates of all 9 calibration points sequentially in the order they were collected.
- ③ Click "Next" to proceed to the "Hand-eye calibration" page.

Step 6: Hand-eye calibration

Determine whether to finalize the calibration based on the calibration results and on-site accuracy requirements.

2D manual calibration

① Set photo point

② Set template

③ Collect calibration point

④ Obtain point coordinates

⑤ Hand-eye calibration

Result

● Need to recalibrate

The current calibration error is greater than 3 (pixels)

Please complete the calibration according to the operating instructions. If the calibration still fails, it may be due to poor accuracy of the robot arm body, and the robot arm needs to be replaced or recalibrated

Cancel

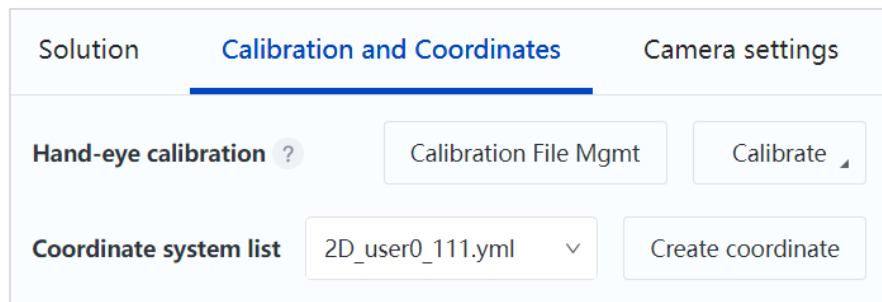
Recalibrate

6.2.2 Common causes of calibration failure

If calibration fails due to excessive error (greater than 3 pixels), follow these steps to troubleshoot:

1. Check that the camera and robot arm are securely installed.
2. Check that the calibration board remains fixed relative to the robot base.
3. Check that the robot arm is calibrated using a positive posture ($J3 > 0$); otherwise, significant errors may occur.
4. Check that the calibration was performed according to the operation guidelines (e.g., the calibration board must be placed on a horizontal surface; the camera lens must remain parallel to the workplane; the captured image must be clear and the calibration board must be centered in the field of view; poor template creation during calibration may result in low recognition accuracy; and the needle tip at the robot end must be accurately aligned with the calibration points during manual calibration).
5. If calibration still fails, it may be due to poor accuracy of the robot arm itself. Consider replacing the robot arm or recalibrating it.

6.2.3 Creating coordinate system based on 2D template

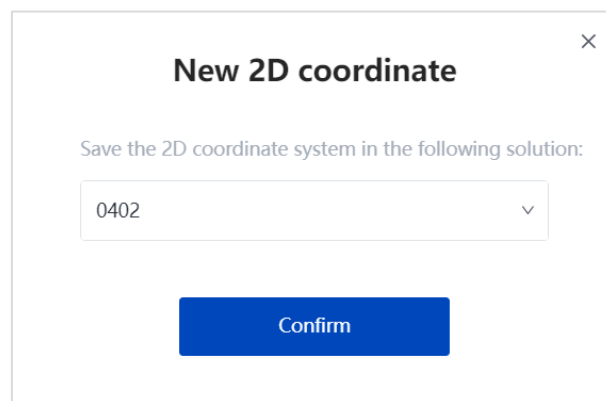


Step 1: Select calibration file

Select a previously saved 2D calibration file from the "Coordinate system list" dropdown menu, then click "Create coordinate".

Step 2: Select storage solution

Select the solution to which the newly created 2D coordinate system belongs. The 2D positioning tool can only be used in solutions associated with a 2D coordinate system. Click "Confirm" to proceed to the "Move to photo point" page.



Step 3: Move to photo point

Create coordinate

1 **Move to photo point**

2 **Set template**

3 **Set user coordinate system**

Select tools

Please choose which tool will be used to configure the user coordinate system.

When using the Cam_user application, users do not need to set the tool coordinate system and can directly teach, reducing errors.

Select tools

Template ▼

Move to photo point

① Adjust the posture of the robot arm to the photo point at the current calibration

② Place a target to the center of your vision field

Collect image from photo point

☒ Not obtained

Cancel

Obtain photo point

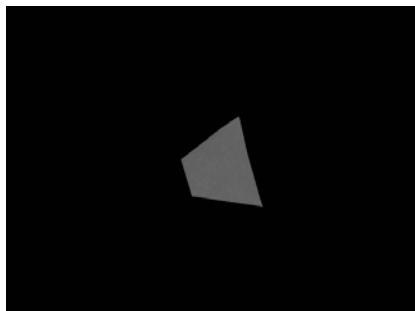
- **Select tool**

Choose whether to configure the user coordinate system using the "Template" or "Spot" tool.

When using Cam_user, the tool coordinate system does not need to be set, allowing direct teaching and reducing errors.

- **Move to photo point**

- ① Move the robot arm to the photo point used during 2D calibration.
- ② Place the target to be recognized at the center of the camera's field of view. Click "Take photo" to confirm that the target appears centered.



- ③ Click "Obtain photo point" to get the photo point's position data.
- ④ Click "Next" to proceed to the "Set template" page.

Step 4: Set template

Create coordinate

① Move to photo point
2 Set template
③ Set user coordinate system

Range settings

Detection area

☒
☐
☐

Mask area
Edit

Search settings

Template area
Edit

Template mask area
Edit

Template sensitivity

Manual
Auto

Sensitivity

Low
Middle
High

50

Recognition Num

−

10

+

1
50

Result evaluation

Count range

Extended params

Min score

50

Match polarity

Consider polarity v

Angle range

-45

-

45

Overlap rate

50

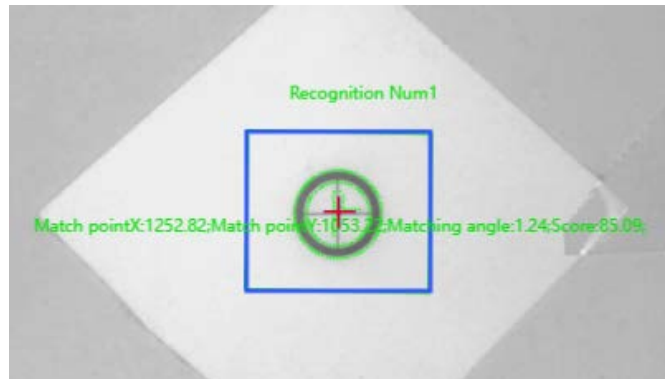
Algorithm timeout

0

- ① In the **Template area**, select the ROI (Region of Interest) shape to frame the template area. The operation is the same as setting the template in 2D

automatic calibration, ensuring that the target's contours are clearly defined.

- ② Set the **Template matching point**, which can be defined as a corner of the target.



- ③ To reduce false detections, increase the **Min score**, recommended at 75.

Extended params

Min score	75
Match polarity	Consider polarity v
Angle range	-145 - 145
Overlap rate	50
Algorithm timeout	0

- ④ Adjust the rotation angle based on the actual application to ensure the target can be correctly recognized.
- ⑤ Once the template is set, click "Next" to proceed to the "Set user coordinate system" page.

Step 5: Set user coordinate system

Create coordinate

1 Move to photo point

2 Set template

3 Set user coordinate system

Set result

● Recognition successful

Set as user coordinate system

user0WorkSurface_...
▼

user0WorkSurface_Cam_User2

Cancel

Last step

Finish

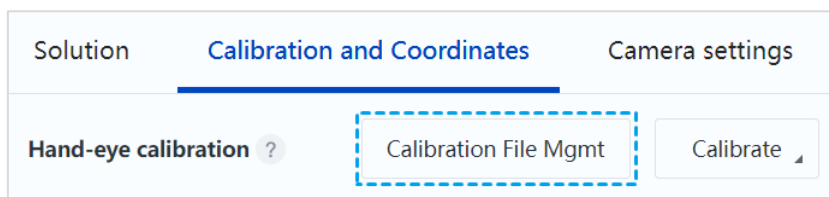
Once recognition is successful, select the desired user coordinate system name (automatically generated by the system and cannot be customized), then click "Finish" to save.

If the coordinate system recognition fails, check if the image is clear and whether the template has been correctly set.

Once the coordinate system is saved, you will return to the "Calibration and Coordinates" page, where the newly created coordinate system will be displayed.

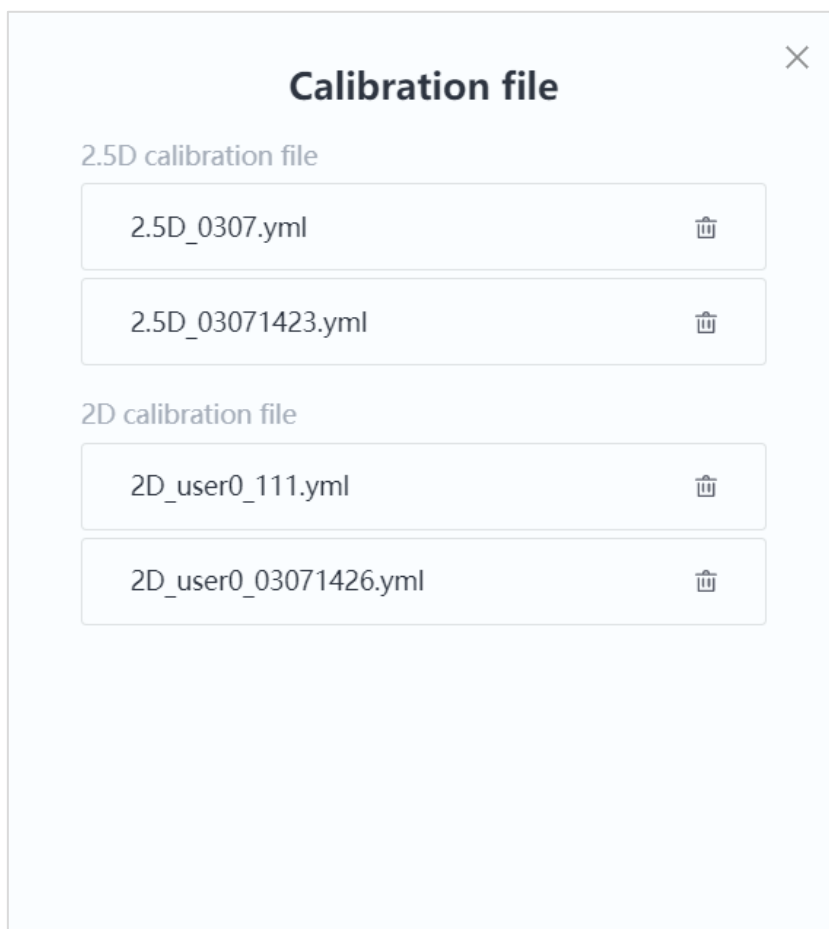
6.3 Managing calibration file

On the "Calibration and Coordinates" page, click "Calibration File Mgmt" (Calibration File Management) to view all saved calibration files.



Click  to delete a specific calibration file.

Calibration file names cannot be modified, as this may cause the SmartCamera plugin to malfunction.



6.4 Managing coordinate system

Solution
Calibration and Coordinates
Camera settings

Hand-eye calibration ?
Calibration File Mgmt
Calibrate

Coordinate system list
2D_user0_111.yml
Create coordinate

Coordinate system	Calibration file	No.	Position code	Operation
Cam_User1	2.5D_03071423.yml	2		Delete
user0WorkSur...	2D_user0_111.yml			Update Delete

- When "2.5D calibration file" is selected in the "Coordinate system list", all 2.5D coordinate systems in the list will display an "Update" button.
- When "2D calibration file" is selected in the "Coordinate system list", all 2D coordinate systems in the list will display an "Update" button.
- If the following two conditions are met, you can adjust the robot position and camera parameters as needed and click "Update" on this interface to directly update the coordinate system.
 - The coordinate system to be updated was created within the current solution. Otherwise, an error message appears: "This coordinate system was not created under the current solution and cannot be updated".
 - The template or spot used when creating the coordinate system is still recognizable within the camera's field of view.
- Click "Delete" in the "Operation" column to remove the corresponding coordinate system.



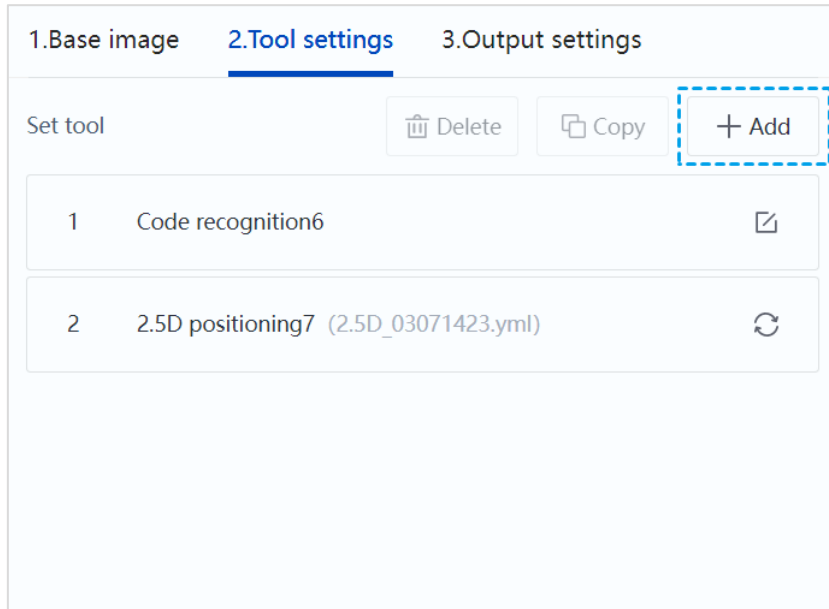
NOTICE

Do not modify or clear the camera coordinate system (Cam_User) in the **User coordinate system** page of DobotStudio Pro, as this may cause the camera project to run abnormally.

7. Vision tool

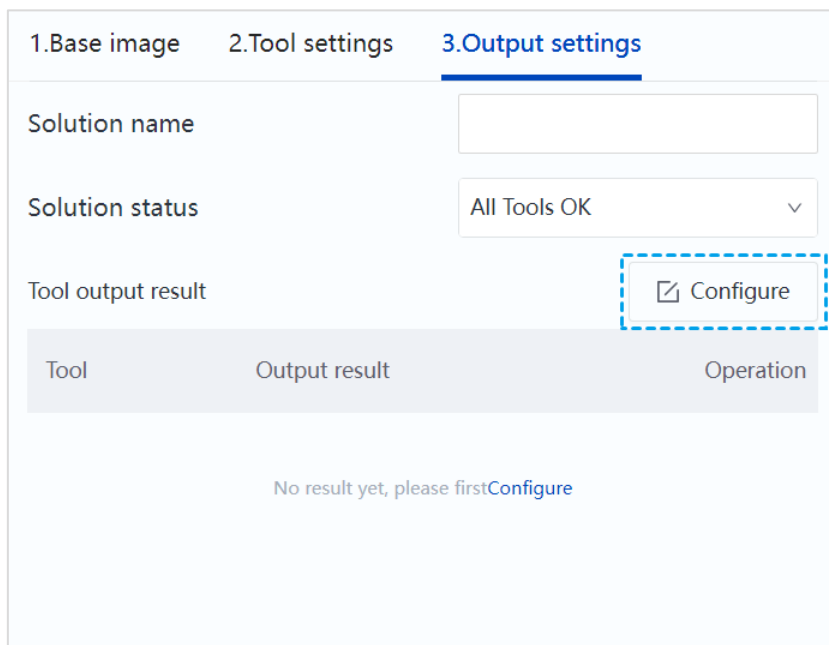
- Add tool**

When creating a new solution, switch to the "Tool settings" page. Click the "Add" button to open the tool selection window, where you can add [positioning](#), [recognition](#), and [measurement](#) tools based on your actual needs.



- Configure tool output result**

When creating a new solution, switch to the "Output settings" tab. Click the "Configure" button to open the "Tool output result" configuration page.



Select the output results for the added tools (multiple selections allowed), click "**Confirm**", and the selected output results will be output, which can be viewed in the interface or called via programming.

Positioning tool

☒ 2.5D positioning 7

Recognition tool

☐ Code recognition 6

Configure Tool

☒ Module status

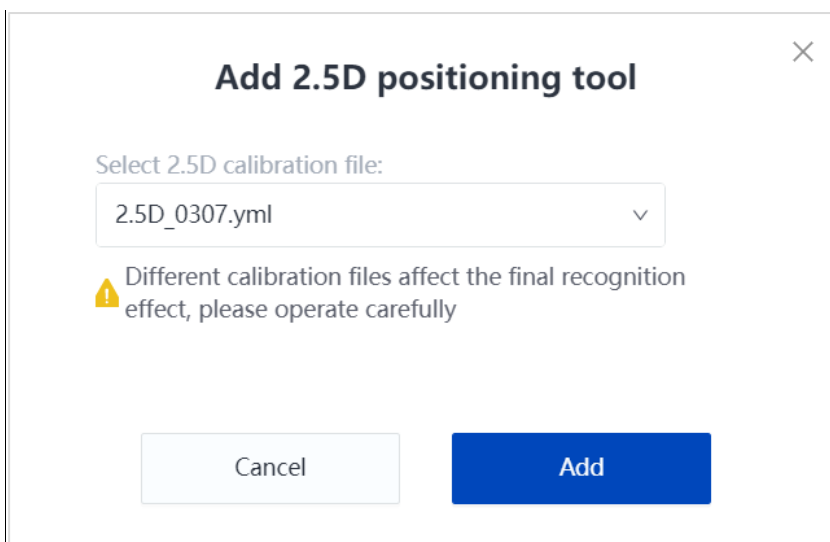
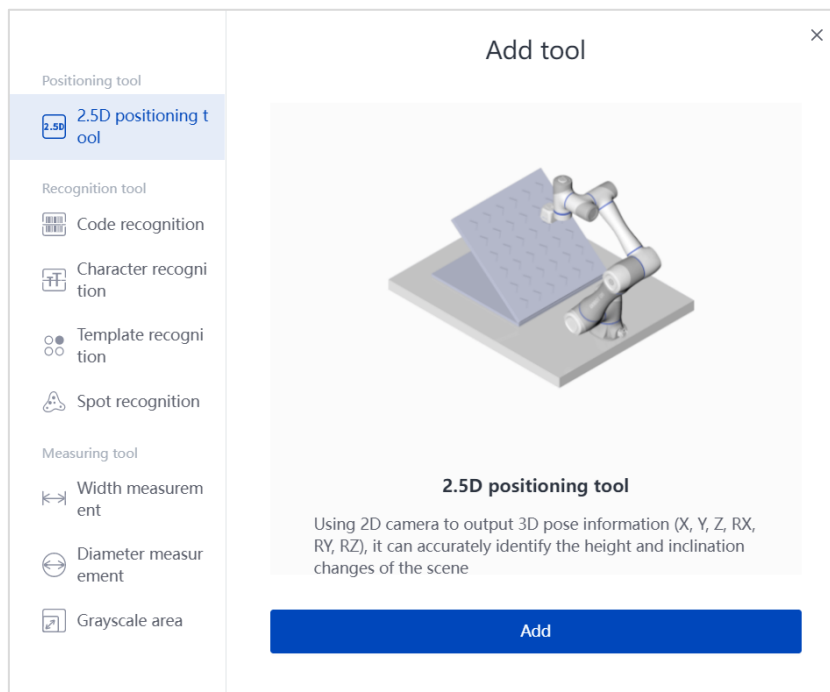
☒ Coordinate system

Confirm

7.1 Positioning tool

7.1.1 2.5D positioning

When a 2.5D calibration file has been selected and a 2.5D coordinate system has been created, this tool can be used in combination with programming to update the 2.5D user coordinate system.



Select the 2.5D calibration file for positioning and click "Add" to complete the addition of the 2.5D positioning tool.



NOTICE

The "2.5D positioning tool" option will appear in the left navigation window only if a 2.5D calibration file exists. Only one 2.5D positioning tool can be added.

2.5D positioning output results: Module status (OK/NG), Coordinate system.

Positioning tool

☐ 2.5D positioning 6

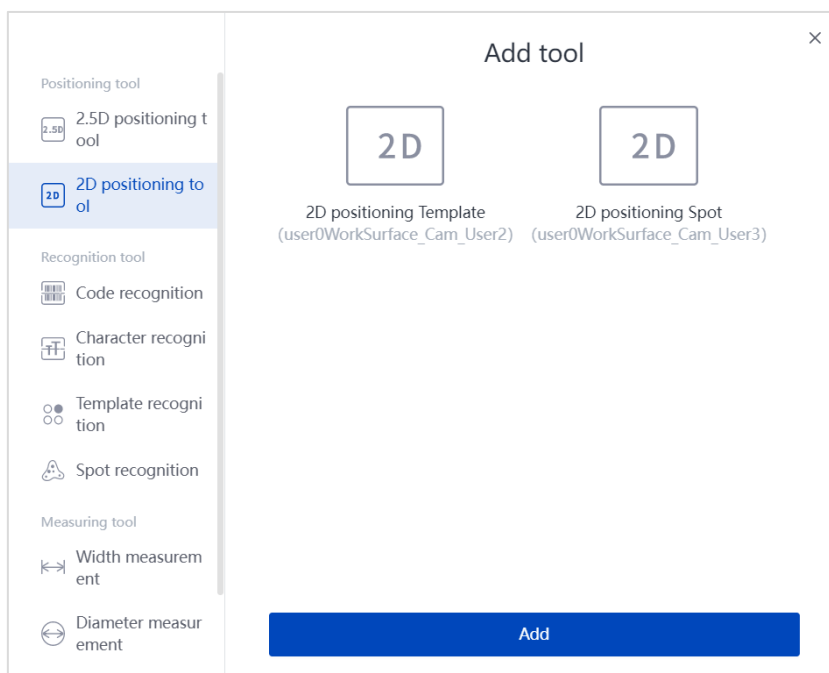
Configure Tool

☐ Module status ☐ Coordinate system

Confirm

7.1.2 2D positioning

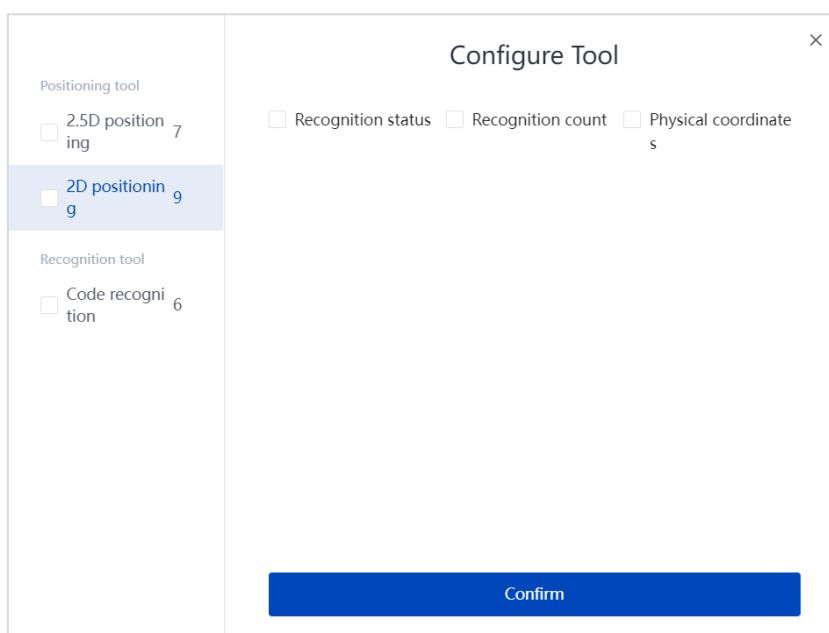
When a 2D calibration file has been selected and a 2D coordinate system has been created, this tool can be used in combination with programming to update the 2D user coordinate system.



This tool is uniquely bound to a 2D coordinate system. Select "Positioning tool > 2D positioning tool" from the left navigation window, click "Add", and it will be ready for use without additional parameter configuration.

2D positioning output results:

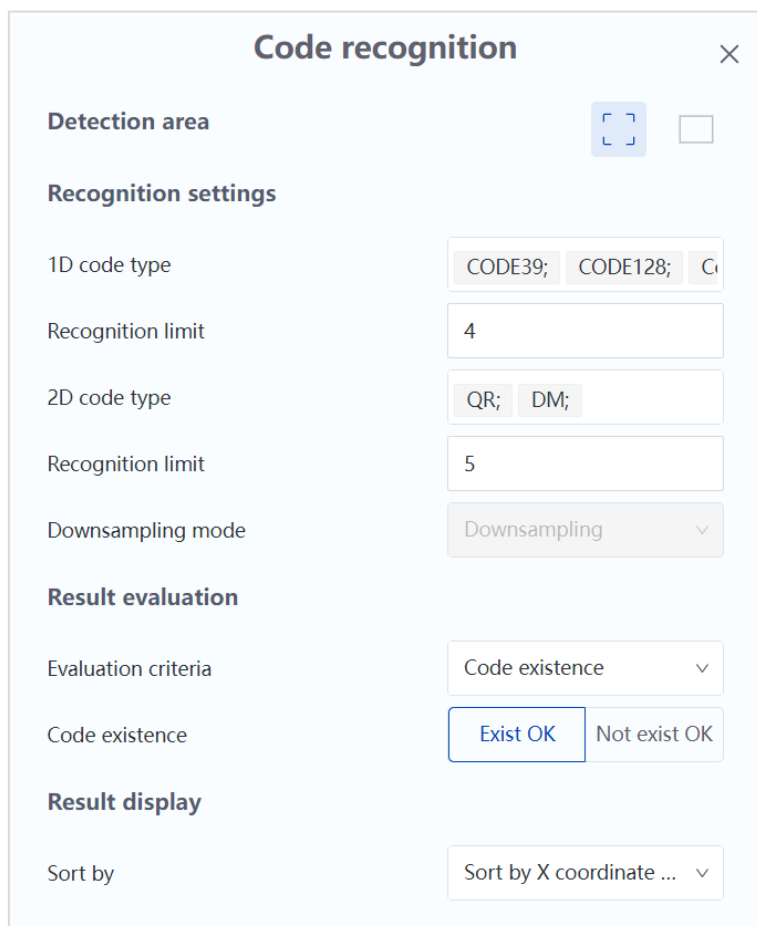
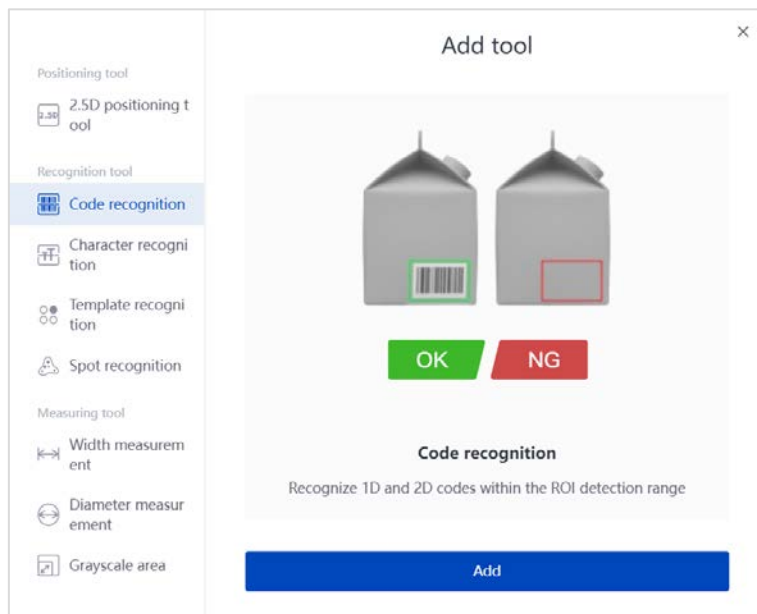
Module status (OK/NG), Recognition count, Physical coordinates.



7.2 Recognition tool

7.2.1 Code recognition

The **Code recognition** tool can recognize 1D and 2D codes within the detection area.



1. Detection area

- Set the **Detection area** based on your actual needs. By default, it covers the full screen, but you can also use the rectangle tool to define a specific area.

2. Recognition settings

- Set the code type to recognize (multiple selections allowed) and the maximum number of codes to recognize each time (range: 1 – 200).
 - 1D code types: CODE39, CODE128, Codabar, EAN8, EAN13, UPCA, UPCE, ITF25, and CODE93 (all selected by default).
 - 2D code types: QR and DM (all selected by default).

3. Result judgment

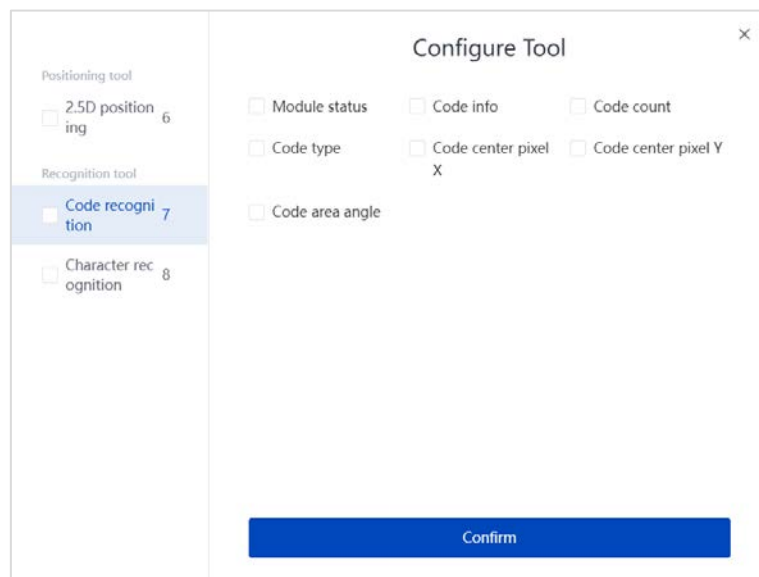
- Set the criteria for determining tool results.
 - When set to **Code existence**, **Exist OK** means that if at least one code is recognized, the result is OK. **Not exist OK** means that if no code is recognized, the result is OK.
 - When set to **Min code score**, you need to set a minimum score. If the actual score is higher than the minimum score, the result is OK; otherwise, it is NG.
 - When set to **Code count**, you need to set a range for the number of recognized codes. If the recognized count falls within this range, the result is OK; otherwise, it is NG.

4. Result display

- Set the sorting method for the output results.

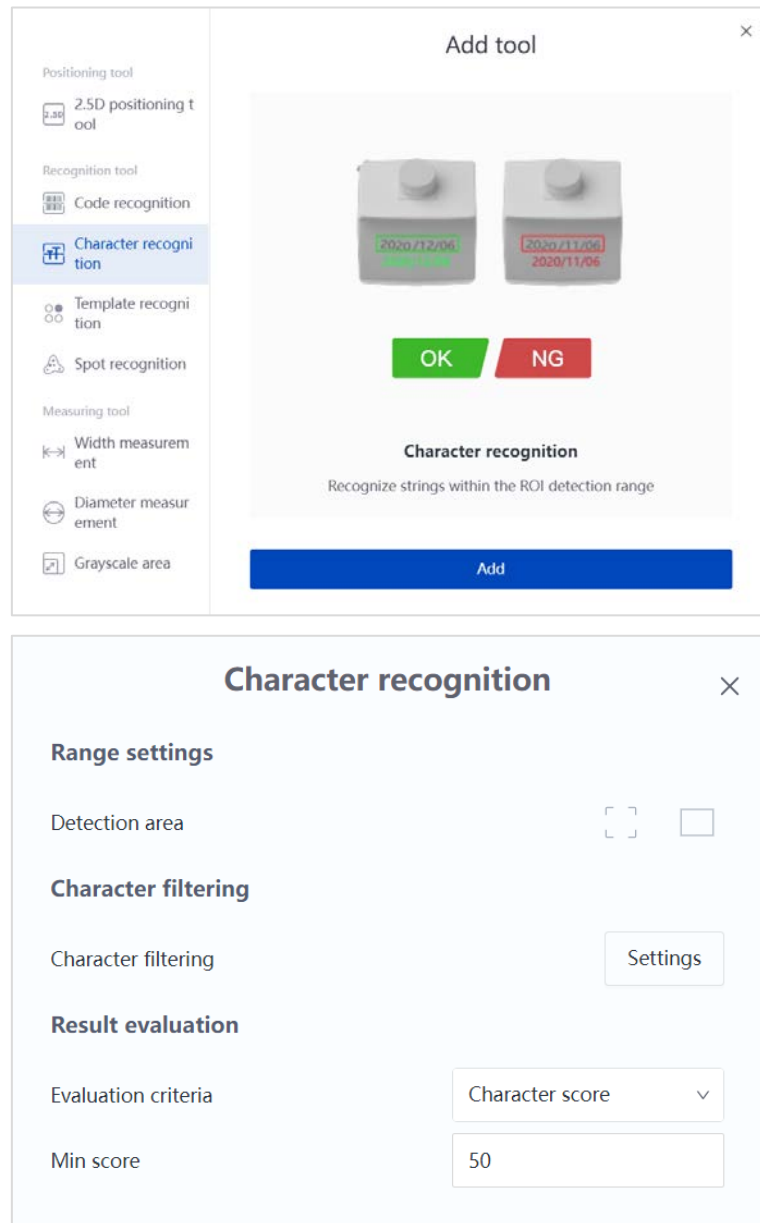
Code recognition output results:

Module status (OK/NG), Code information, Code count, Code type, Code center pixel X, Code center pixel Y, Code area angle.



7.2.2 Character recognition

The **Character recognition** tool can recognize strings within the detection area.



1. Range settings

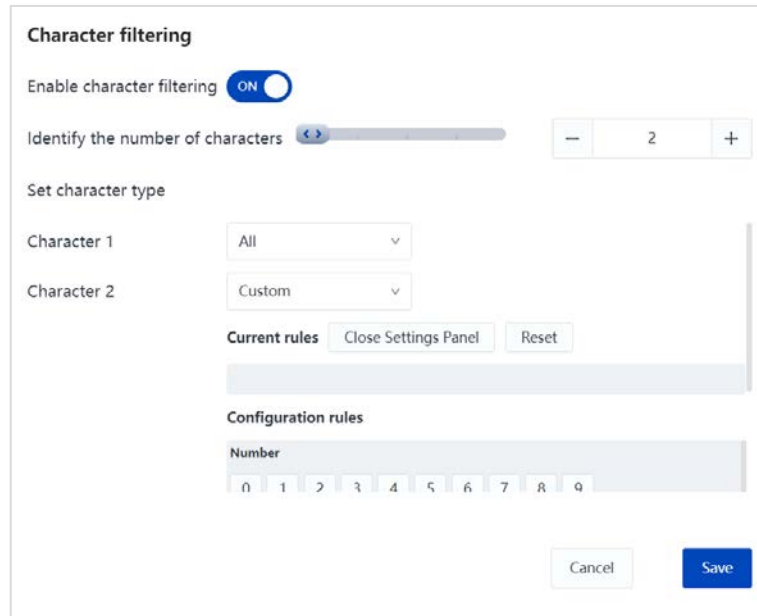
- Set the **Detection area** based on your actual needs. By default, it covers the full screen, but you can also use the rectangle tool to define a specific area.

2. Character filtering

- Click the "Settings" button next to "Character filtering" to open the configuration page. Once character filtering is enabled, you can specify the number of recognized characters and set the character type.
 - Enable character filtering: Disabled by default.
 - Recognized character count: Range: 0 – 100 (default: 0).
 - Set character type: Options include "All", "Number", "Uppercase

letters", "Lowercase letters", "Special characters", "Space", and "Custom" (default: All).

- Custom rule panel: When "Custom" is selected, the **Custom rule panel** appears (default: disabled). After enabling the panel, you can select specific numbers, uppercase letters, lowercase letters, or special characters. Click to highlight a selection and click again to deselect it.

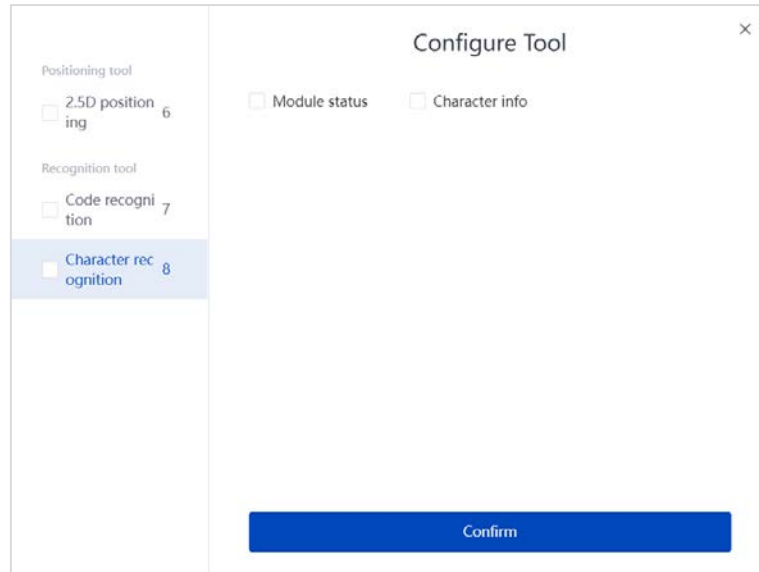


The image shows a 'Character filtering' settings panel. At the top, there is a toggle switch for 'Enable character filtering' which is currently turned 'ON'. Below this is a slider for 'Identify the number of characters' set to 2, with minus and plus buttons for adjustment. Under the heading 'Set character type', there are two dropdown menus: 'Character 1' is set to 'All' and 'Character 2' is set to 'Custom'. Below these are buttons for 'Current rules', 'Close Settings Panel', and 'Reset'. A section titled 'Configuration rules' contains a 'Number' row with buttons for digits 0 through 9. At the bottom right are 'Cancel' and 'Save' buttons.

3. Result judgment

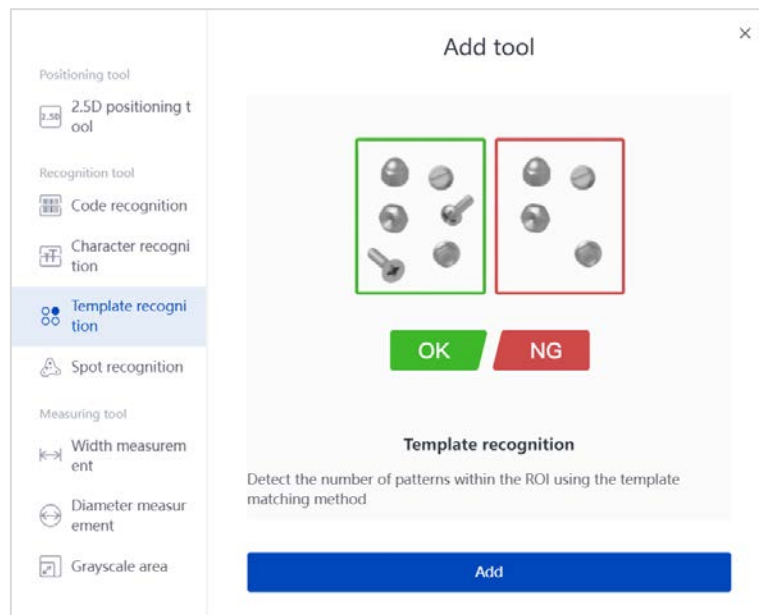
- Set the criteria for determining tool results.
 - When set to **Character score**, you need to set a minimum score. If the actual score is higher than the minimum score, the result is OK; otherwise, it is NG.
 - When set to **Character count**, you need to set a range for the number of recognized codes. If the recognized count falls within this range, the result is OK; otherwise, it is NG.
 - When set to **Base character**, you need to set a predefined character as a reference. If the recognized text matches the reference, the result is OK; otherwise, it is NG.

Character recognition output results: Module status (OK/NG), Character information.



7.2.3 Template recognition

The template recognition tool can recognize the number of patterns within the detection area using template matching.



×

Template recognition

Range settings

Detection area

+
+
+
+

Mask area
Edit

Search settings

Template area
Edit

Template mask area
Edit

Template sensitivity

Manual
Auto

Sensitivity

< >

Low Middle High

50

Recognition Num

−

10

+

< >

1 50

Result evaluation

Count range

< >

Extended params

Min score

50

Match polarity

Consider polarity ▼

Angle range

-45

-

45

Overlap rate

50

Algorithm timeout

0

1. Range settings

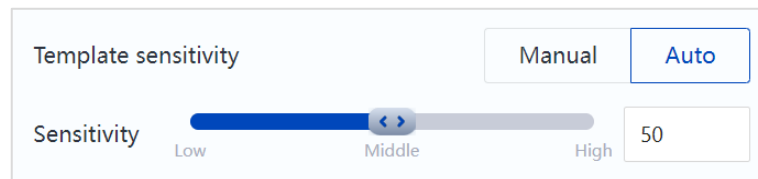
- Set the **Detection area** based on your actual needs. By default, it covers the full screen, but you can also use the rectangle or circle tool to define a specific area. Only one detection area is supported.
- If parts of the detection area need to be masked, click the "Edit" button to select the polygon tool, and draw the masking area within the detection area. Double-click to complete the drawing. Up to 8 masking areas can be created.

2. Search settings

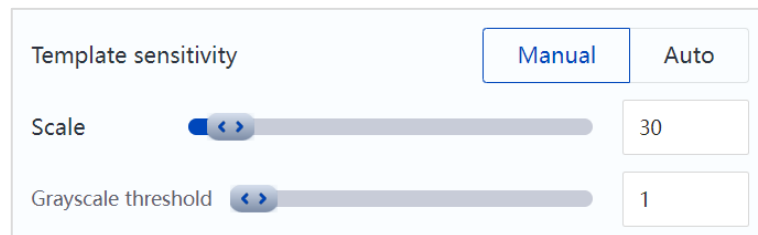
- Set the **Template area** based on your actual needs. Click the "Edit" button and use the rectangle or polygon tool to manually draw the template area.

Up to 8 template areas can be created.

- If parts of the template area need to be masked, click the "Edit" button to select the rectangle or polygon tool, then draw the template area within the detection area. Double-click to complete the drawing. Up to 8 template masking areas can be created.
- Adjust the **Template sensitivity** based on your actual needs, using either **Automatic** or **Manual** mode.
 - Automatic mode: Set the sensitivity level (range: 0 – 100, step: 25, default: 50). Higher sensitivity increases the number of detected feature points in the template area, improving matching accuracy. However, excessive sensitivity may cause noise to be misidentified as feature points. Adjust as needed.



- Manual mode: Set the **Scale** (range: 10 – 200, default: 30) and **Grayscale threshold** (range: 1 – 255, default: 1).



The scale value determines the downsampling factor (scale value ÷ 10). For example, if the scale is set to 50, the downsampling factor is 5, meaning 1 pixel is sampled every 5 pixels in each row and column of the original image to form a new image. Smaller values increase the detection time, while larger values reduce the sharpness of the image contours, which may affect detection accuracy. Adjust this setting based on your actual needs.

The grayscale threshold defines the range of grayscale values in the detection area. Only areas within the specified range will be recognized.

- Recognition count: Set the maximum number of recognized areas that meet the criteria (range: 1 – 50, default: 10, step: 1). If the number of recognized areas exceeds the set limit, only the most optimal areas will be output.

3. Result judgment

- **Count range:** 0 – 50, default: 0 – 10. If the recognition count falls within the set range, the result is OK; otherwise, it is NG. The maximum value of **Count range** is determined by the **Recognition count** parameter.

4. Extended parameters

Extended params

Min score

50

Match polarity

Consider polarity ▾

Angle range

-45 - 45

Overlap rate

50

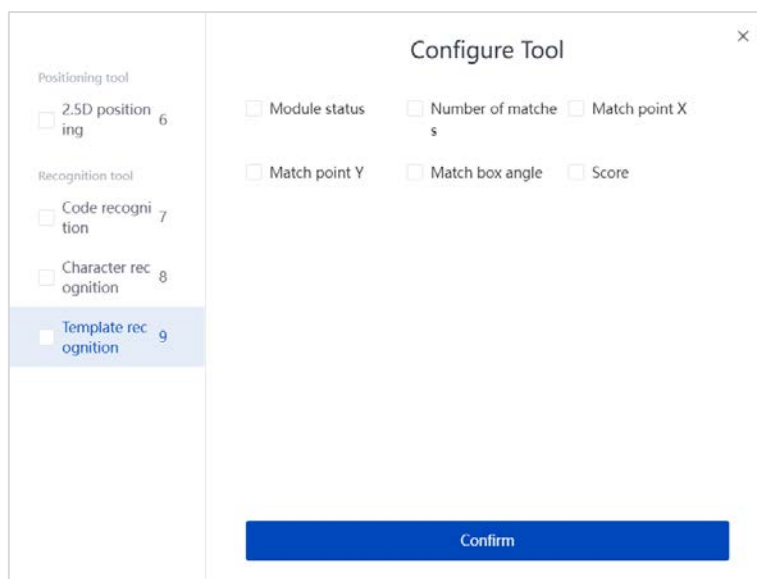
Algorithm timeout

0

- Minimum score:** Defines the similarity threshold between the feature template and the target in the image. Range: 0 – 100, default: 50, with 100 indicating a perfect match. Only targets with a similarity score equal to or greater than this threshold will be detected.
- Match polarity:** Specifies whether to consider polarity in template matching. Polarity refers to the contrast transition between the feature shape and the background. If the target's edge polarity differs from the feature template's polarity, set this option to **Ignore polarity** to ensure the target can be detected. If there is no need to ensure the target can be detected, set this option to **Consider polarity** (default setting), which reduces the detection time.
- Angle range:** Defines the allowable angle variation for template matching. If the target's angle variation stays within this range, it can still be detected; otherwise, it cannot be recognized. Range: $[-180^{\circ} - 180^{\circ}]$ and $a \leq b$, default: $-45^{\circ} - 45^{\circ}$.
- Overlap rate:** When searching for multiple targets, this parameter defines the maximum allowable overlap between the matching boxes of two detected targets. A higher value allows more overlap between two targets. Range: 0 – 100%, default: 50%.
- Algorithm timeout:** If the algorithm detection time exceeds the set value, detection will automatically stop once the timeout is reached, and output NG. If this parameter is set to 0, the algorithm timeout function is disabled, and the algorithm detection time is unrestricted. Range: 0 – 10000, default: 0.

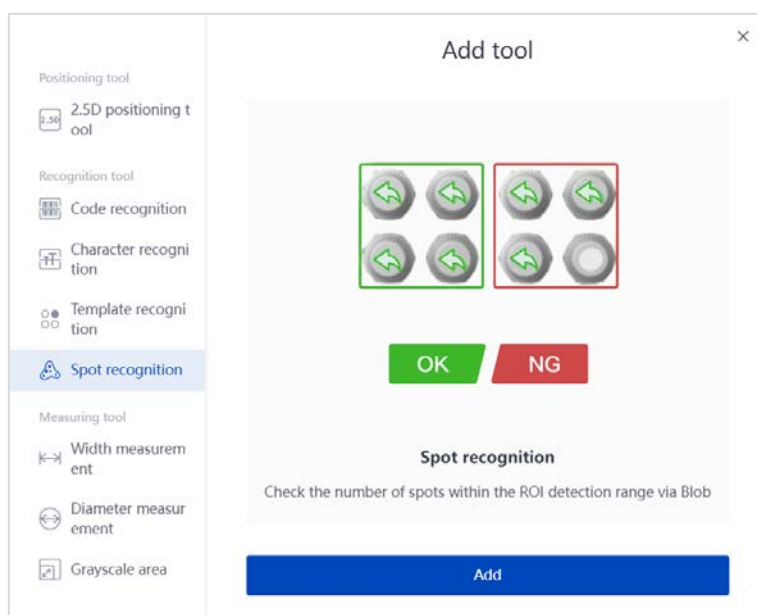
Template recognition output results:

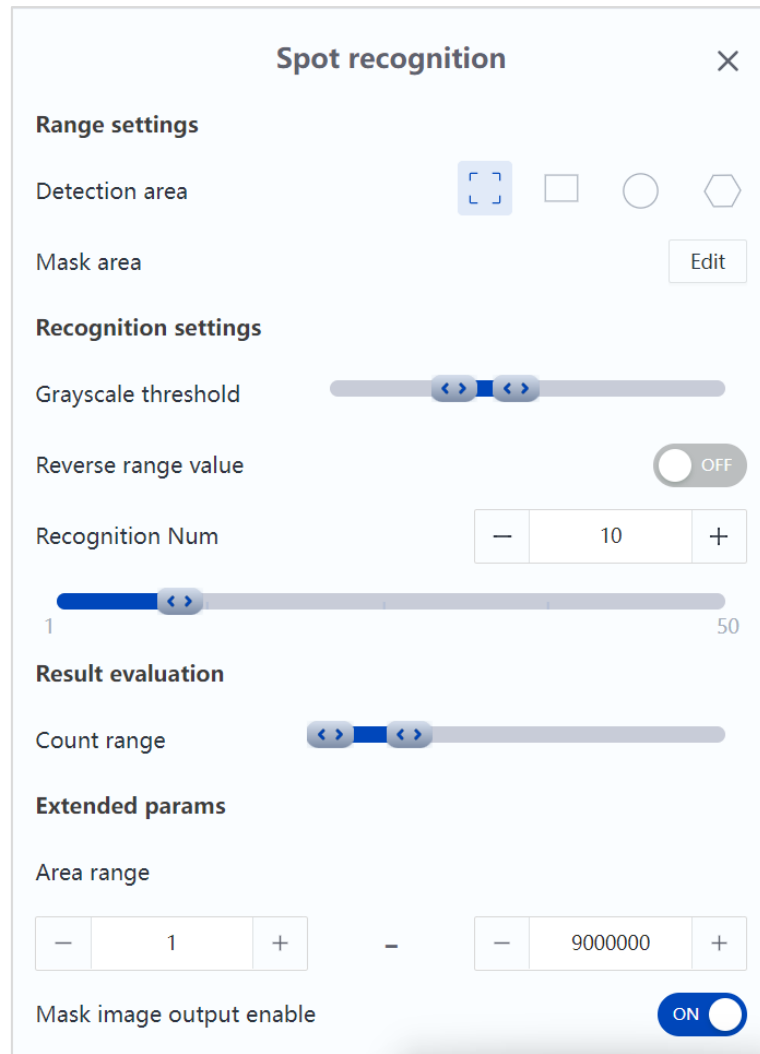
Module status (OK/NG), Number of matches, Match point X, Match point Y, Match box angle, Score.



7.2.4 Spot recognition

The spot recognition tool can recognize the number of spots within the detection area using the Blob method. Blob analysis is the process of detecting, locating, or analyzing targets within an image region of limited grayscale range. It enables the extraction of various object features, such as presence, quantity, position, shape, orientation, and topological relationships between blobs.





1. Range settings

- Set the **Detection area** based on your actual needs. By default, it covers the full screen, but you can also use the rectangle, circle, or polygon tool to define a specific area. Up to 8 detection areas can be set.
- If parts of the detection area need to be masked, click the "Edit" button to select the polygon tool, and draw the masking area within the detection area. Double-click to complete the drawing. Up to 8 masking areas can be created.

2. Recognition settings

- **Grayscale threshold:** Adjust the grayscale threshold based on actual detection requirements. Pixels within the selected threshold range will be identified and marked in the image. Range: 0 – 255, default: 80 – 120. System automatically selects a middle-range value.
- **Invert range:** Disabled by default. Enable the **Invert range** option to detect values at both ends of the **Grayscale threshold**.
- **Recognition count:** Set the maximum number of recognized areas that meet the criteria (range: 1 – 50, default: 10, step: 1). If the number of recognized areas exceeds the set limit, only the most optimal areas will be output.

3. Result judgment

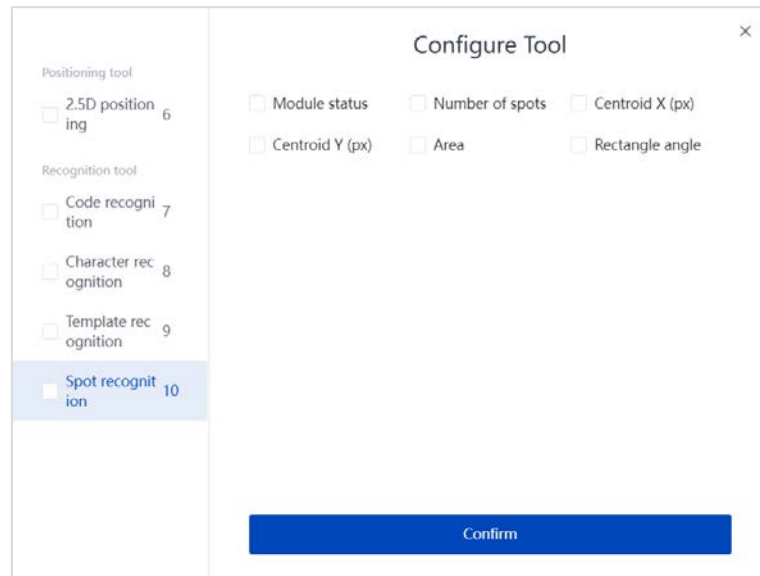
- **Count range:** 0 – 50, default: 0 – 10. If the recognition count falls within the set range, the result is OK; otherwise, it is NG. The maximum value of **Count range** is determined by the **Recognition count** parameter.

4. Extended parameters

- **Area range:** If the recognition results are not ideal, you can adjust the area range under the extended parameters to define the size of the recognized area. The unit is pixels. Range: 1 – 9000000, default: 1 – 9000000.
- **Mask image output enable:** When enabled, the mask output hides irrelevant details, reducing interference and improving visibility of targets. Disabled by default.

Spot recognition output results:

Module status (OK/NG), Number of spots, Centroid X (pixel), Centroid Y (pixel), Area, Rectangle angle.



Positioning tool

☐ 2.5D positioning 6

Recognition tool

☐ Code recognition 7

☐ Character recognition 8

☐ Template recognition 9

☒ Spot recognition 10

Configure Tool

☐ Module status
☐ Number of spots
☐ Centroid X (px)

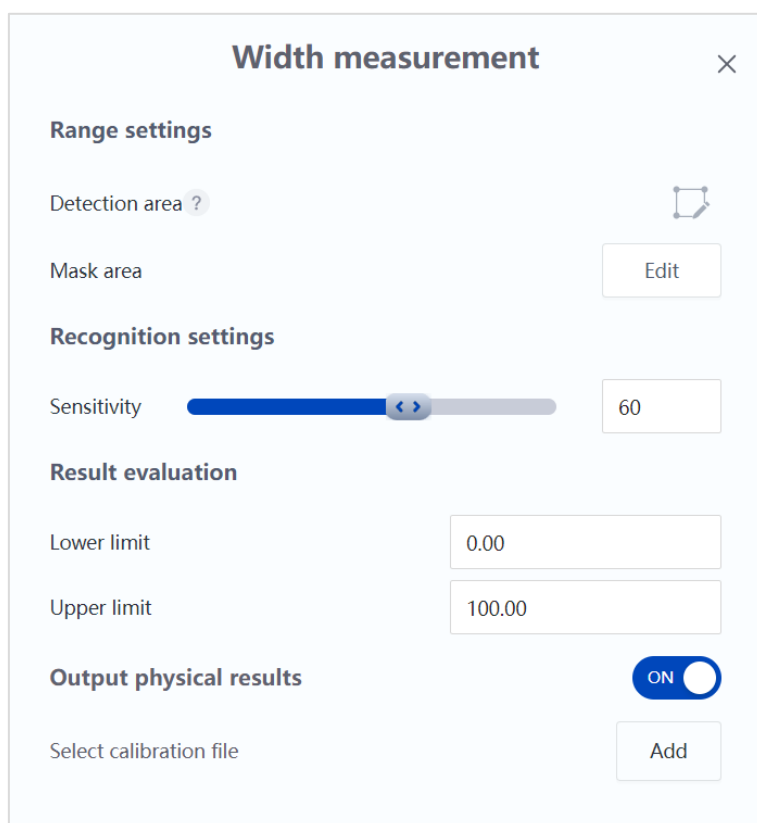
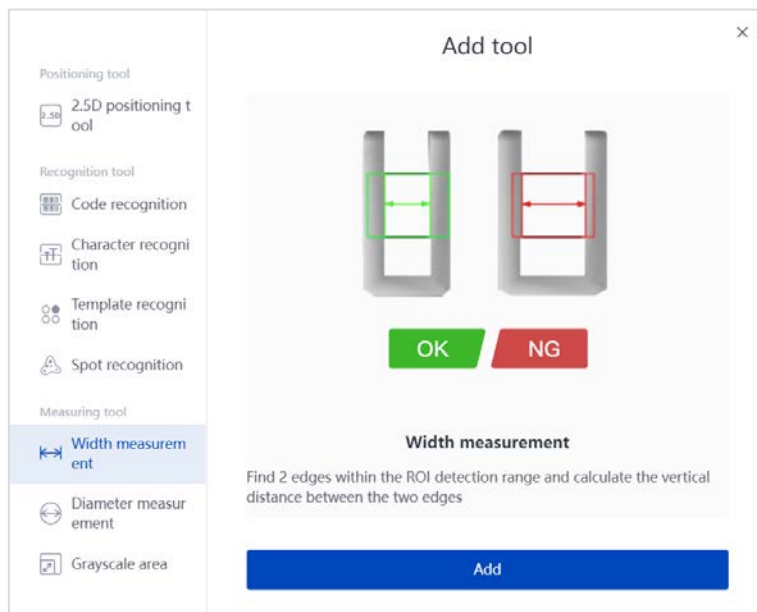
☐ Centroid Y (px)
☐ Area
☐ Rectangle angle

Confirm

7.3 Measurement tool

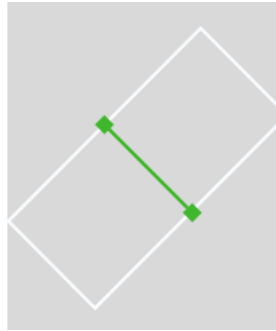
7.3.1 Width measurement

The width measurement tool detects two edges within the selected detection area and calculates the vertical pixel distance between them.



1. Range settings

- Click  to select two endpoints and define the detection area for width measurement.



- If parts of the detection area need to be masked, click the "Edit" button to select the polygon tool to draw a masking area.

2. Recognition settings

- Set the sensitivity of the width measurement according to your actual needs.

3. Result judgment

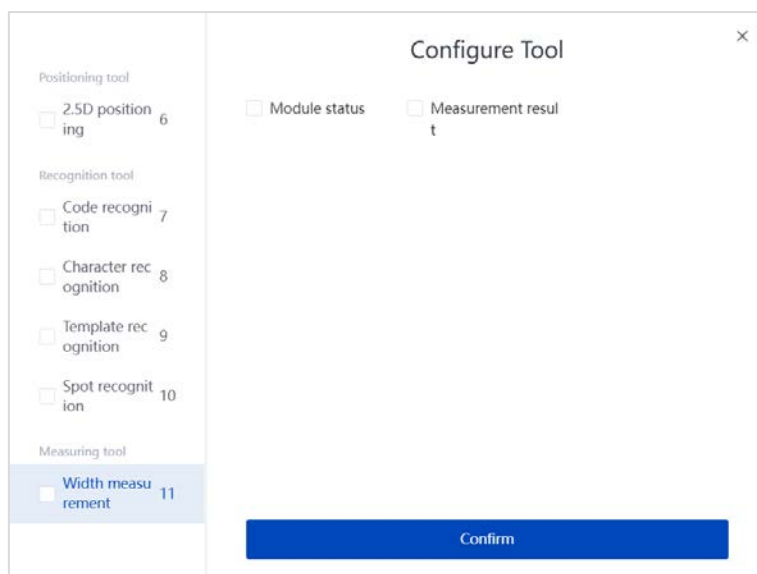
- Set the OK range for the measurement result. If the measured pixel width falls within this range, the result is OK; otherwise, it is NG.

4. Output physical results

- When the **Output physical results** option is OFF, the tool outputs the width as a pixel result. When the option is ON, the tool outputs the width as a physical result.
- **Select calibration file:** This option appears only when **Output physical results** is ON.
 - Click the "Add" button to open a pop-up window for selecting a calibration file. Select the desired calibration file and click "Confirm" to add it. If no 2D calibration files are available, a message will appear: "No calibration file can be added yet, please calibrate first".
 - The first added calibration file is automatically set as the current calibration file. Up to 4 calibration files can be added.

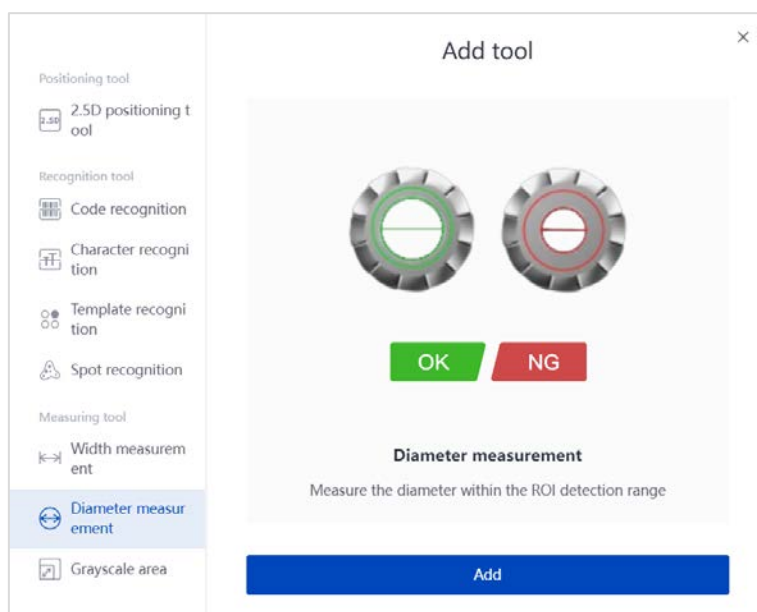
Width measurement output results:

Module status (OK/NG), Measurement result (Pixel value).



7.3.2 Diameter measurement

The diameter measurement tool detects circles within the selected detection area and calculates the pixel diameter of the detected circle.



Diameter measurement
✕

Range settings

Detection area


Mask area

Edit

Recognition settings

Sensitivity

<

>

60

Result evaluation

Lower limit

0.00

Upper limit

100.00

Extended params

Roundness

<

>

25

Edge polarity

Any ▼

Edge type

Strongest ▼

Output physical results

ON

Select calibration file

Add

1. Range settings

- **Detection area:** Use the tool  to click the center of the circle to be measured to generate a fixed-size circular detection area, which can then be manually adjusted. If the clicked area is too close to the image boundary, the circular detection area may extend beyond the image, causing the drawing to fail.
- **Mask area:** If parts of the detection area need to be masked, click the "Edit" button to select the polygon tool to draw a masking area.

2. Recognition settings

- Set the sensitivity of the circle detection according to your actual needs.

Issue V1.2.0 (2025-04-08)

User Guide

Copyright © SHENZHEN DOBOT CORP LTD

87

3. Result judgment

- Set the OK range for the measurement result. If the measured pixel diameter falls within this range, the result is OK; otherwise, it is NG.

4. Extended parameters

If circle detection result is not ideal after running, you can set the **Extended parameters**.

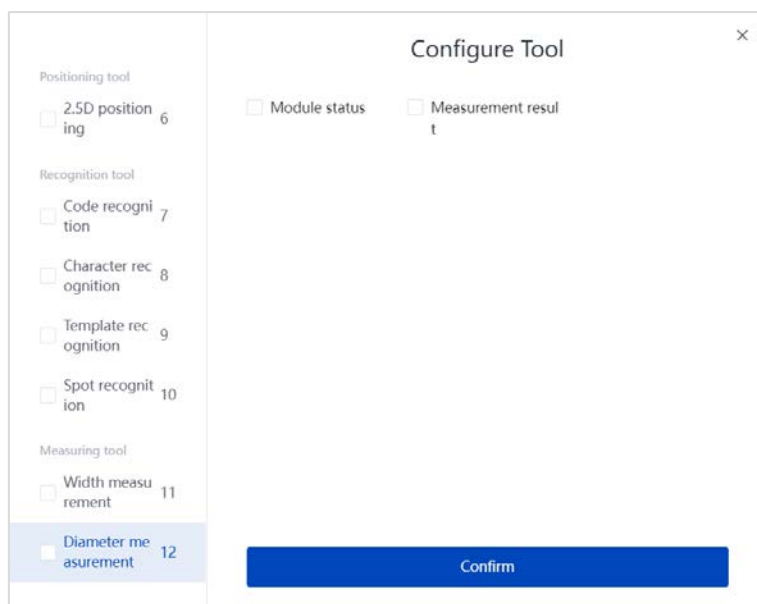
- **Roundness:** The minimum ratio of points contributing to the circle fitting relative to the total number of points. A higher value means the detected circle is closer to a perfect circle. If the roundness of the detected circle exceeds this value, it is considered a circle; otherwise, it is not considered a circle.
- **Edge polarity:** Specify the color transition of the detected edge. Edge polarity is related to direction, which is defined by the arrow in the ROI area.
 - Black to White: The edge between areas transitioning from lower to higher grayscale values.
 - White to Black: The edge between areas transitioning from higher to lower grayscale values.
 - Any: Both types of edges (Black to White and White to Black) are detected.
- **Edge type:** Specify the detection type of edge.
 - Strongest: Detects only the set of edge points with the largest gradient threshold within the range and fits a circle.
 - Maximum: Detects only the set of edge points that are farthest from the center of the circle within the range and fits a circle.
 - Minimum: Detects only the set of edge points closest to the center of the circle within the range and fits a circle.

5. Output physical results

- When the **Output physical results** option is OFF, the tool outputs the diameter as a pixel result. When the option is ON, the tool outputs the diameter as a physical result.
- **Select calibration file:** This option appears only when **Output physical results** is ON.
 - Click the "Add" button to open a pop-up window for selecting a calibration file. Select the desired calibration file and click "Confirm" to add it. If no 2D calibration files are available, a message will appear: "No calibration file can be added yet, please calibrate first".
 - The first added calibration file is automatically set as the current calibration file. Up to 4 calibration files can be added.

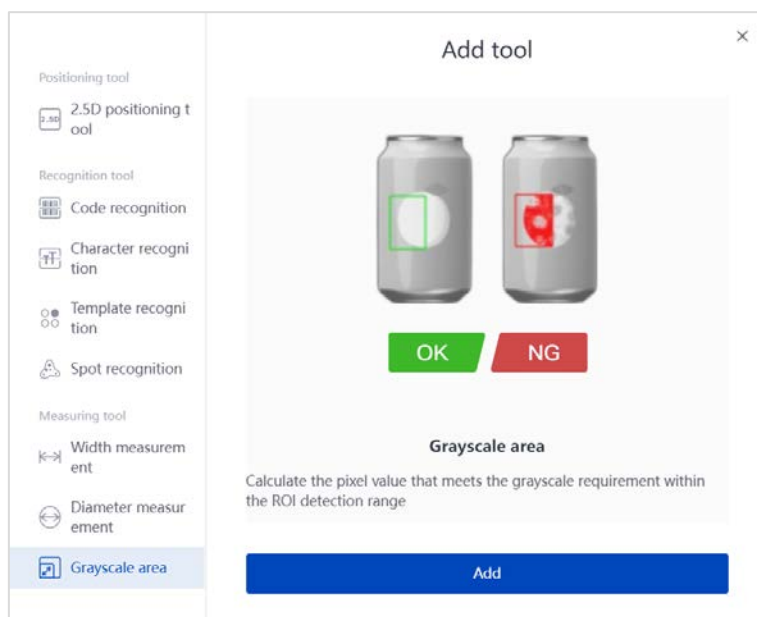
Diameter measurement output results:

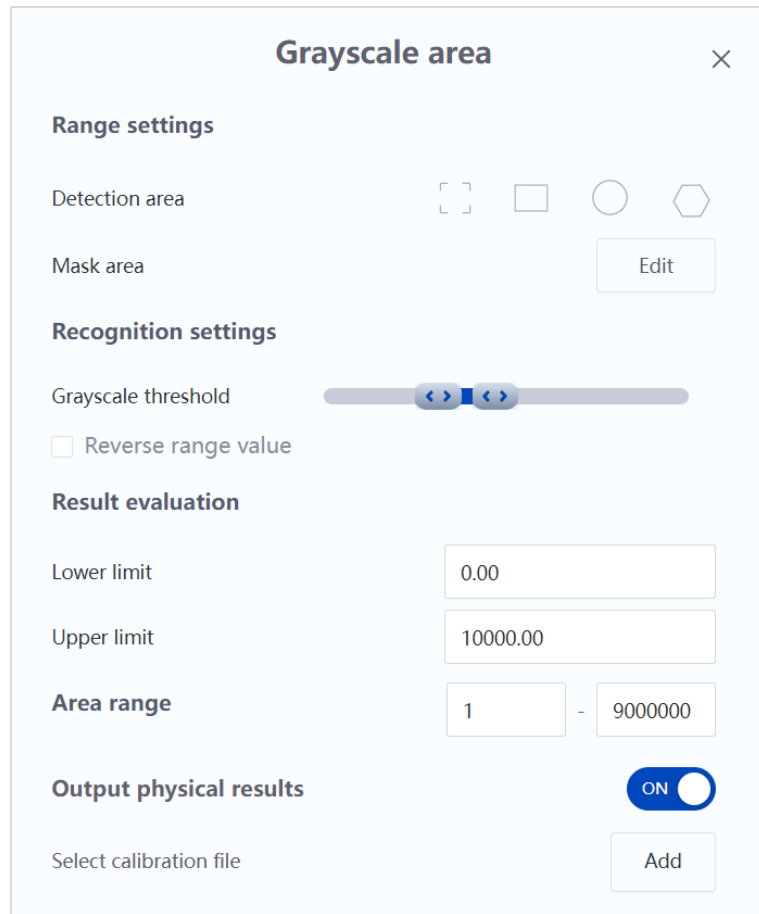
Module status (OK/NG), Measurement result (Pixel value).



7.3.3 Grayscale area

The grayscale area tool calculates the pixel area within the detection area that meets the specified grayscale range.





1. Range settings

- **Detection area:** Set the detection area based on your actual needs. By default, it covers the full screen, but you can also use the rectangle, circle, or polygon tool to define a specific area.
- **Mask area:** If parts of the detection area need to be masked, click the "Edit" button to select the polygon tool to draw a masking area.

2. Recognition settings

- **Grayscale threshold:** Adjust the grayscale threshold based on actual detection requirements. The system defaults to the middle range of two values. To use the range of both ends, enable the "**Invert range**" function.

3. Result judgment

- Set the OK range for the measurement result. If the measured average pixel area falls within this range, the result is OK; otherwise, it is NG.

4. Area range

- Set the grayscale area range. Pixels outside this range will be discarded.

5. Output physical results

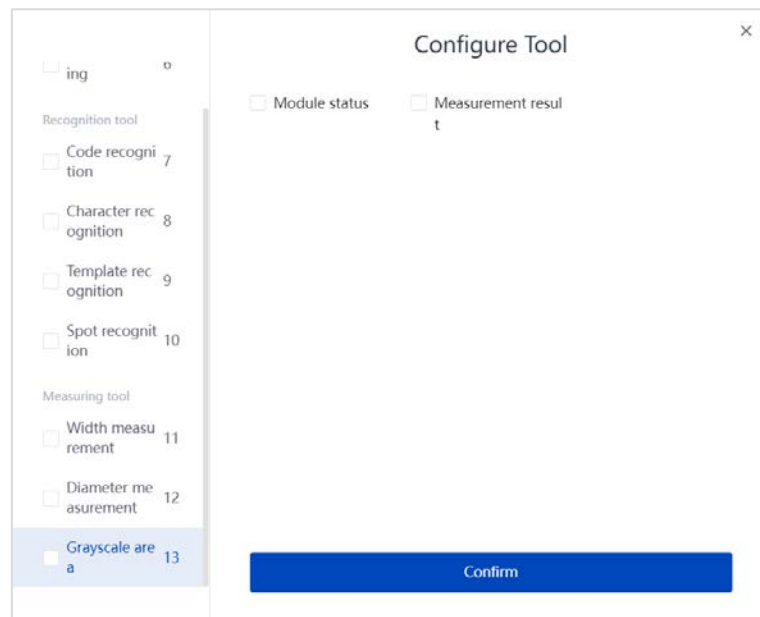
- When the **Output physical results** option is OFF, the tool outputs the grayscale area as a pixel result. When the option is ON, the tool outputs the grayscale area as a physical result.
- **Select calibration file:** This option appears only when **Output physical**

results is ON.

- Click the "Add" button to open a pop-up window for selecting a calibration file. Select the desired calibration file and click "Confirm" to add it. If no 2D calibration files are available, a message will appear: "No calibration file can be added yet, please calibrate first".
- The first added calibration file is automatically set as the current calibration file. Up to 4 calibration files can be added.

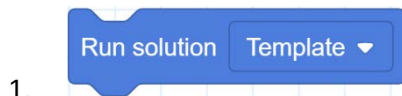
Grayscale area output results:

Module status (OK/NG), Measurement result (Pixel value).



8. Blockly/Script programming

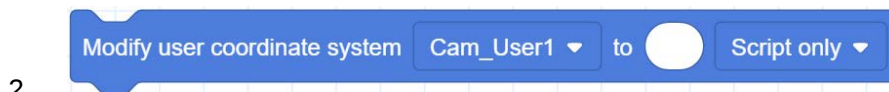
8.1 Block description



1.

Description: Run the selected solution, take photos, execute the tools, and save the output results. In Blockly programming, this block can be used to switch solutions.

Parameter: Solution created in the SmartCamera plugin.



2.

Description: Modify the specified user coordinate system to the coordinate system output by the 2.5D positioning or 2D positioning tool.

Parameter:

- 1) The user coordinate system to be modified, only user coordinate systems created by the SmartCamera plugin are supported.
- 2) Enter the new value.
- 3) Select whether this modification should be globally saved or valid only within the script (If set to "Script only", the modification will only take effect while the current project is running and will revert to its previous state once the project stops).



3.

Description: Get the output result of the executed solution. If there is no result, it returns null. For recognition tools that can detect multiple targets, this block allows specifying which target's output result to obtain. This block must be used below the



block. If the selected solution does not match the currently running solution, a warning will appear: "Inconsistent with current solution".

Parameter:

- 1) The executed solution.
- 2) The configured output results of the executed solution.
- 3) Specify which target's output result to obtain. For tools that can only output one result, this parameter is invalid and the first result is always output.

Return: The specified output result based on the designated target.

- If the positioning tool fails to find coordinates, an error will occur, and the program will stop. Other tools will return "nil" if they fail to obtain the specified result.
- The obtained module status is either "OK" or "NG", coordinate format is {X, Y, Z, RX, RY, RZ}, and other outputs return their respective numeric or string values.

NOTE

If the recognized code information or character information contains a newline character (\r or \n) or a semicolon (;), it may cause a failure in parsing the returned result.

Get 2.5D positioning code S/N

4.

Description: Obtain the serial number of the 2.5D positioning code recognized by the 2.5D positioning tool in the currently running solution.

Return: The recognized 2.5D positioning code serial number. If no 2.5D positioning code is recognized or if the solution does not contain a 2.5D positioning tool, it returns 0.

Sync and update user coordinate system 1 ▼ based on Cam_User1 ▼ , Script only

5.

Description: Based on the relationship between the workplane's user coordinate system and the camera's user coordinate system (i.e., the global values stored in the controller), it calculates and updates the camera's user coordinate system. It then updates the workplane's user coordinate system based on the latest camera user coordinate system from the script. Applicable scenarios: 2D positioning based on user coordinate system. Linkage scenarios where the user coordinate system is updated using 2.5D positioning. The modification is valid only when the project is running, and the modified coordinate system changes back to the original one once the project stops.

Parameter:

- 1) The index of the workplane's user coordinate system to be updated.
- 2) The camera's 2.5D user coordinate system to be calculated and based on.

8.2 Script commands

1. **Command:** RunVX500Project("project_name")

Description: Run the selected solution, take photos, execute the tools, and save the output results. If no solution is running, an error occurs, indicating that no script is currently in execution.

Parameter:

Parameter	Description
project_name	Solution created in the SmartCamera plugin.

Example:

```
RunVX500Project("2.5D_test")
```

2. **Command:** GetVX500ModelRes(moduleName, id, resultType, tagIndex)

Description: Get the output result of the currently running solution.

Parameter:

Parameter	Description
moduleName	Tool name. Range: ◦ "matchlocate": 2D positioning ◦ "macapriltag": 2.5D positioning ◦ "dlocrdetect": Character recognition ◦ "idemodule": Code recognition ◦ "imagecount": Template recognition ◦ "specklecount": Spot recognition ◦ "diametermeasure": Diameter measurement ◦ "widthmeasure": Width measurement ◦ "greyarea": Grayscale area
id	Tool ID (the number assigned to the added tool in the solution).
resultType	The type of output result. Different tools take different values, see the table below for tool-specific values.
tagIndex	Optional parameter. For recognition tools that can detect multiple targets, this parameter allows specifying which target's output result to obtain. Default is 1.

Tool name VS resultType values:

Tool name	resultType values
"matchlocate": 2D positioning	◦ "state": Module status, "OK" or "NG" ◦ "count": Recognition count ◦ "coord": Coordinates, in the format {X,Y,Z,RX,RY,RZ}
"macapriltag": 2.5D positioning	◦ "state": Module status, "OK" or "NG" ◦ "coord": Coordinates, in the format {X,Y,Z,RX,RY,RZ}
"dlocrdetect": Character recognition	◦ "state": Matching status, "OK" or "NG" ◦ "content": Character information
"idemodule": Code recognition	◦ "state": Module status, "OK" or "NG" ◦ "centerX": Code center pixel X ◦ "centerY": Code center pixel Y ◦ "count": Code count ◦ "content": Code information ◦ "type": Code type ◦ "ang": Code pixel angle
"imagecount": Template recognition	◦ "state": Module status, "OK" or "NG" ◦ "count": Number of matches ◦ "matchX": Match point X ◦ "matchY": Match point Y ◦ "matchAng": Match box angle ◦ "score": Score
"specklecount": Spot recognition	◦ "state": Module status, "OK" or "NG" ◦ "count": Number of spots ◦ "centroidX": Centroid X (px) ◦ "centroidY": Centroid Y (px) ◦ "area": Area ◦ "blobAngle": Rectangle angle

Tool name	resultType values
"diametermeasure": Diameter measurement	"state": Module status, "OK" or "NG" "diameter": Measurement result (Diameter)
"widthmeasure": Width measurement	"state": Module status, "OK" or "NG" "width": Measurement result (Width)
"greyarea": Grayscale area	"state": Module status, "OK" or "NG" "area": Measurement result (Grayscale area)

Return: The specified output result. If the positioning tool fails to find coordinates, an error will occur, and the program will stop. Other tools will return "nil" (Lua's null value) if they fail to obtain the specified result.

Example:

```
local coord1 = GetVX500ModelRes("macapriltag", 1, "coord")
```

NOTE

If the recognized code information or character information contains a newline character (\r or \n) or a semicolon (;), it may cause a failure in parsing the returned result.

3. **Command:** SetVX500CamUser("Cam_User", User_table)

Description: Modify the specified user coordinate system.

Parameter:

Parameter	Description
Cam_User	The name of the user coordinate system to be modified.
User_table	The new coordinate system, can be obtained using the <i>GetVX500ModelRes</i> command.

Example:

```
local coord1 = GetVX500ModelRes("macapriltag", 1, "coord")
SetVX500CamUser("Cam_User1", coord1)
```

4. **Command:** GetVX500CodeID()

Description: Obtain the serial number of the 2.5D positioning code recognized by the 2.5D positioning tool in the currently running solution.

Return: The recognized 2.5D positioning code serial number. If no 2.5D positioning code is recognized or if the solution does not contain a 2.5D positioning tool, it returns 0.

5. **Command:** SetVX500User(user_index, "Cam_User")

Description: Based on the relationship between the workplane's user coordinate system and the camera's user coordinate system (i.e., the global values stored in the controller), it calculates and updates the camera's user coordinate system. It then updates the workplane's user coordinate system based on the latest camera user coordinate system from the script. Applicable scenarios: 2D positioning based on user coordinate system. Linkage scenarios where the user coordinate system is updated using 2.5D positioning. The modification is valid only when the project is running, and the modified coordinate system changes back to the original one once the project stops.

Parameter:

Parameter	Description
user_index	The index of the workplane's user coordinate system, which must be pre-added in DobotStudio Pro. If the specified coordinate system index does not exist, the project will stop and return an error.
Cam_User	The index of the camera's 2.5D user coordinate system, which must be pre-added in DobotStudio Pro. If the specified coordinate system index does not exist, the project will stop and return an error.

Example:

```
SetVX500User(1, "Cam_User1")
```

9. Vision Application Cases

NOTICE

- The SmartCamera plugin V1.2.0 must strictly be used with the [Compatible versions](#) listed in the Preface section. If the camera firmware version was previously V3.0.1, any old solutions created before the upgrade will not work with the current version. Please delete the old solution and create a new one.
- Please follow the introduction outlined in the [Hardware/Software Installation and Connection](#) section to complete the following operations:
 1. Connect the computer and camera.
 2. Install the camera plugin.
 3. Complete the network configuration.
 4. Upgrade the camera firmware (optional, see [Firmware upgrade](#)).
 5. Test the camera functions.

9.1 2.5D positioning case

9.1.1 Technical overview

2.5D positioning is a self-developed algorithm by Dobot. It uses a 2D camera to capture a **2.5D positioning code** and calculates the target's 3D spatial coordinates based on the information from the 2.5D positioning code for precise positioning and grabbing.

This method solves the problem of inaccurate positioning caused by height variations in the environment (such as uneven or tilted ground), which conventional 2D positioning cannot handle. For example, if the robot arm is mounted on an AGV, once the AGV moves to a workstation, the robot arm can accurately position the target point as long as the 2.5D positioning code is within the camera's field of view and is correctly recognized, without the need for precise positioning of the AGV or a flat ground surface.

NOTICE

Ensure that the target is in a fixed position relative to the 2.5D positioning code.

Differences between 2D positioning, 2.5D positioning, and 3D positioning:

2D positioning		Recognize the target Output X, Y, RZ
-----------------------	---	--

2.5D positioning		Not recognize the target, but recognize the 2.5D positioning code, and calculate the 3D posture (X, Y, Z, RX, RY, RZ)
3D positioning		Recognize the target Output 3D spatial coordinates (X, Y, Z, RX, RY, RZ)

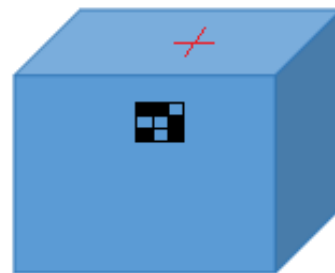
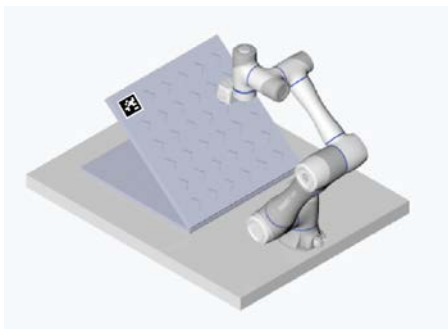
9.1.2 Environmental requirements

- **Robot arm installation**

When using the 2.5D positioning function only, the camera can be mounted at any angle.

- **Target requirements**

When using 2.5D positioning, the target must remain fixed relative to the 2.5D positioning code. The 2.5D positioning code and the target point do not need to be on the same horizontal plane.



9.1.3 2.5D positioning process

- **Environment description**

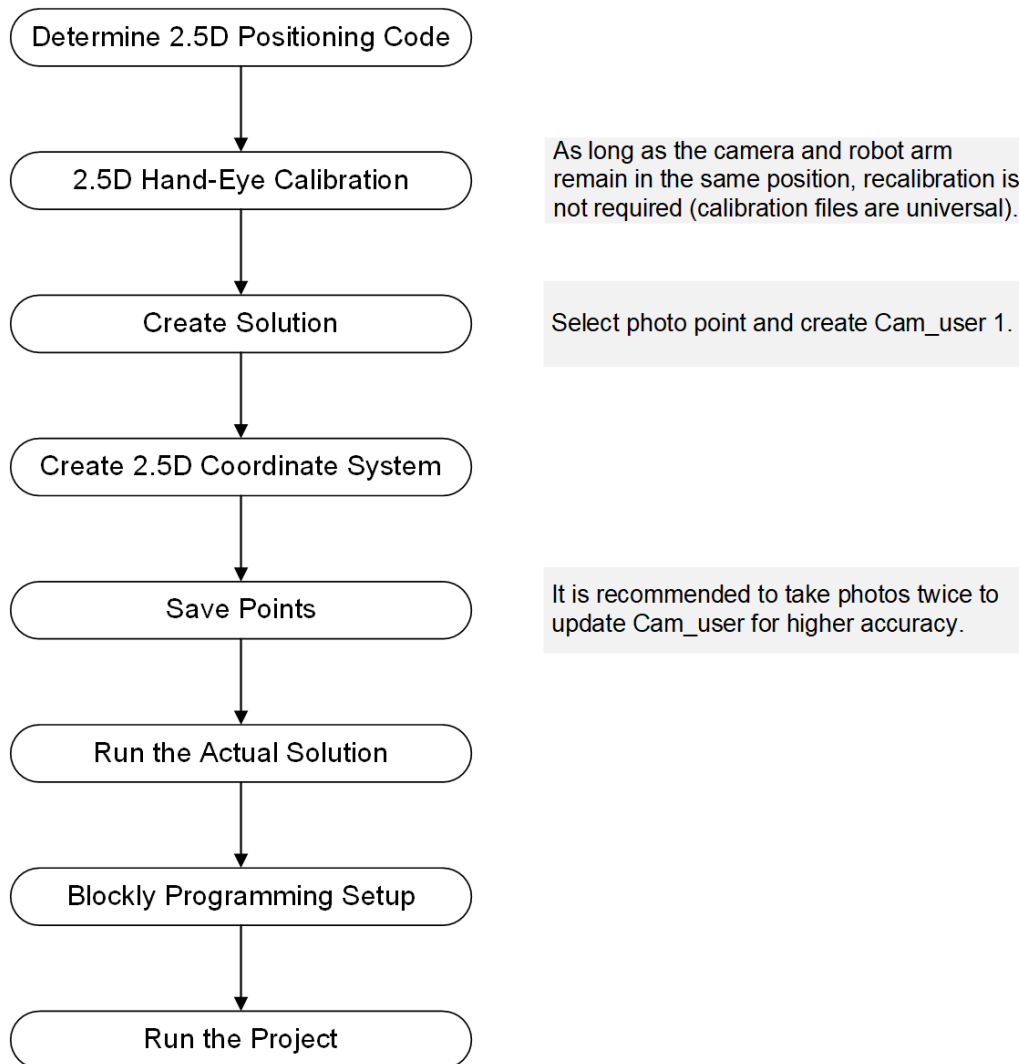
The environment for this demo is shown below. The robot arm is equipped with an eccentric needle tip at the end, and the goal of this demo is to align the needle tip with the center of the cross ("+").

 NOTICE

- This setup simulates a real-world application where the gripper and the robot arm's end effector are eccentric. The same process applies if they are concentric.
- In practical applications, when simulating an AGV-mounted robot arm, the metal plate remains stationary (it can represent a CNC machine or another fixed workstation), while the AGV moves with the robot arm.
- If simulating a stationary robot arm, locate a part with variable positions for assembly. Treat the metal plate as a jig (labeled with a 2.5D positioning code) to fix the part, with the cross ("+") serving as the assembly alignment point.



- 2.5D positioning process flow



NOTICE

As long as the relative position between the robot arm and VX500 smart camera remains unchanged, **the hand-eye calibration only needs to be performed once**. Multiple solutions can be created for different applications, and the same 2.5D calibration file can be used across them.

1. Determine 2.5D positioning code

Based on the actual 2.5D positioning code in use, select the corresponding code size in the "Camera settings" page. See [2.5D positioning code](#) for details.

Solution
Calibration and Coordinates
Camera settings

Camera settings

Disconnect

Device aliaszty_ruanjian_1
IP address192.168.5.43
Gateway192.168.5.1
Subnet mask255.255.255.0
Firmware versionV3.1.0 250326
Factory settings

Firmware upgrade

One-click restore

Camera direction

If you only use the 2.5D positioning function, you don't need to consider the camera installation method. You can install it at will without choosing.

If you're using the 2D application, please choose one of the following two installation methods.


☒ Camera towards base

☐ Camera back towards base

2.5D positioning code

This value is used for calculations when using the 2.5D positioning code in the future

50mm x 50mm (Standard)

2. 2.5D hand-eye calibration

Refer to the [2.5D auto calibration](#) process to complete the calibration and save the calibration file.

2.5D auto calibration

① Set center point
② Set initial photo point
③ Collect calibration point

① Please ensure there are no obstacles within the motion range of robot arm.
 ② After starting the collection, the robot arm will automatically move to the 25 points generated around the calibration board. [Take](#)
[back](#)^

Collect calibration point

● (25/25)

Re-collect

Result

● Excellent

Average translation error (mm): X: 0.200; Y: 0.105; Z: 0.197;
 Max translation error (mm): X: 0.522; Y: 0.347; Z: 0.406;
 Average rotation angle error (°): Rx: 0.105; Ry: 0.175; Rz: 0.015;
 Max rotation angle error (°): Rx: 0.205; Ry: 0.356; Rz: 0.039;

File name

2.5D_
.yaml

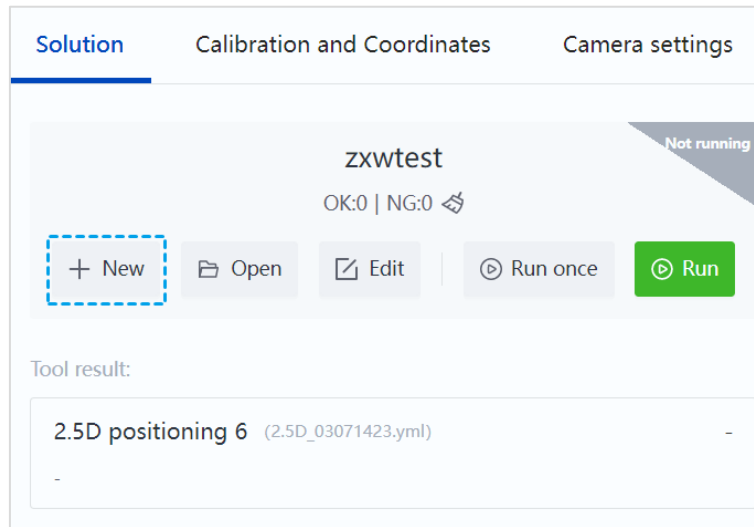
Last step

Save calibration file

Cancel

3. Create solution

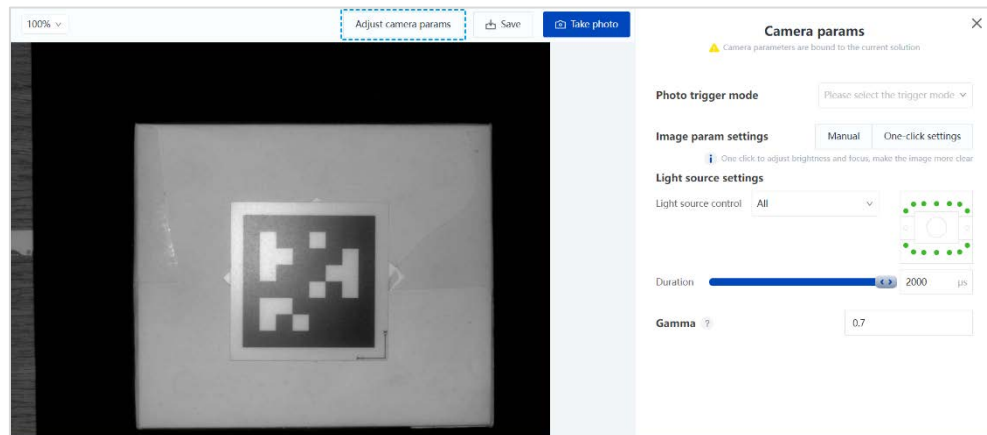
Step 1: Go to the "Solution" homepage and click the "New" button to enter the new solution creation page.



Step 2: Click "Adjust camera params" to enter the camera parameter settings page.

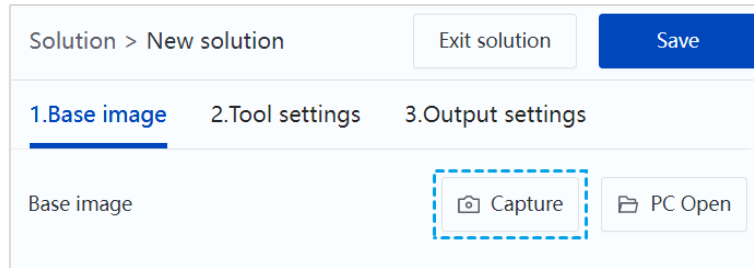
- Click "One-click settings" to automatically adjust the brightness and focus for a clearer image.
- Set the "Photo trigger mode" to "Auto photography".
- Set the "Light source control" to "All".

At this point, the camera will continuously capture and update images, allowing you to see the position of the 2.5D positioning code in real-time.

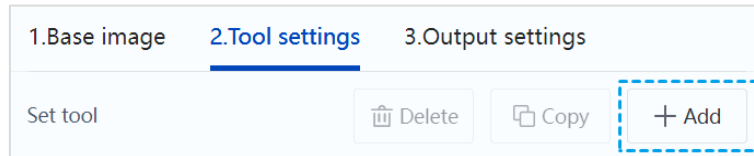


Step 3: Adjust the robot arm's posture so that the 2.5D positioning code is centered in the field of view and occupies approximately 40%–50% of the image width (This point will be used as the photo point in later applications).

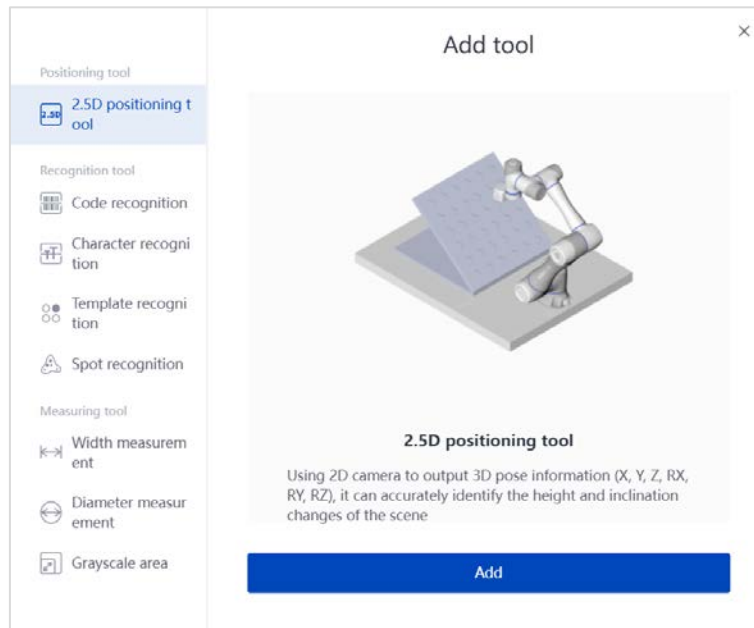
Step 4: Once the camera parameters are adjusted, return to the "New solution > Base image" page and click "Capture" to obtain the base image. If the 2.5D positioning code has not moved within the camera's field of view, the captured image should match the focused image.



Step 5: Go to the "Tool settings" page and click "Add" to open the "Add tool" page.



Step 6: In the left navigation panel, select the "2.5D positioning" tool and click "Add". This will open the "Add 2.5D positioning tool" page.



NOTE

If there is no 2.5D calibration file, the 2.5D positioning tool module will not be displayed on the "Add tool" page.

Step 7: After adding the tool, go back to the "Tool settings" page, where the newly added 2.5D positioning tool will now be displayed.

1.Base image
2.Tool settings
3.Output settings

Set tool
Delete
Copy
Add

1	Code recognition6	
2	2.5D positioning7 (2.5D_03071423.yml)	

Step 8: Go to the "Output settings" page and enter a solution name.

1.Base image
2.Tool settings
3.Output settings

Solution name

Solution status
All Tools OK

Tool output result
Configure

Tool	Output result	Operation
No result yet, please firstConfigure		

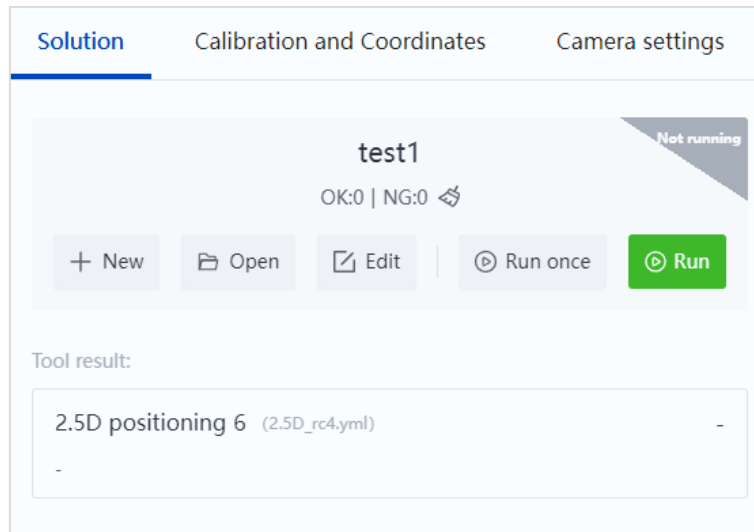
Step 9: Click "Configure" to open the tool configuration page. Select the required output information and click "Confirm".

Positioning tool
2.5D positioning 6

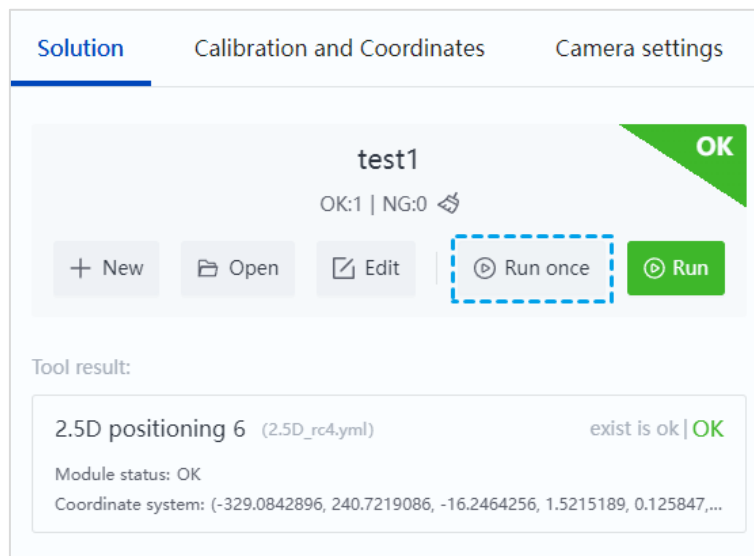
Configure Tool
Module status
Coordinate system

Confirm

Step 10: Once the solution configuration is complete, click "Save" to automatically return to the "Solution" page. The "Solution" page will now display the newly created solution name, along with the corresponding tool results.



Step 11: Click "Run once" and the configured output results will be displayed under the "Tool result" section.



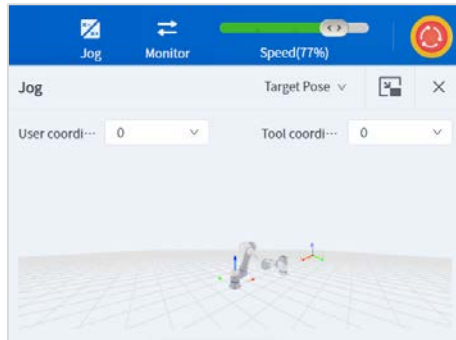
4. Create 2.5D coordinate system

Refer to the [Creating coordinate system based on 2.5D positioning code](#) section to create the 2.5D coordinate system.

5. Save points

Step 1: Save the second photo point

- ① Open the "Jog" page in DobotStudio Pro and set both the User Coordinate System and Tool Coordinate System to "0". Then, go to "Application > Points" page, click "Add point", and name it "2_5D_PhotoPoint_Base".



- ② In the "Jog" page, set the User Coordinate System to "Cam_user" and the Tool Coordinate System to "0". Then, go to "Application > Points" page, click "Add point", and name it "2_5D_SecPhotoPt".

Programming *

Points *

Mode:

Joint

Linear

 Run to

NO.	Point	Alias	User	Tool
1	P1	2_5D_AutoCalPhot...	0 ▾	0 ▾
2	P2	2_5D_PhotoPt_Base	0 ▾	0 ▾
3	P3	2_5D_SecPhotoPt	3(Cr✓	0 ▾

Step 2: Save the first photo point

- ① In the "Jog" page, set both the User Coordinate System and Tool Coordinate System to "0", and raise the robot arm. Open the Dobot+ camera plugin to check the camera's field of view. Ensure that in actual applications, the 2.5D positioning code can still be captured and recognized, even when its relative position to the robotic arm changes significantly (within the range where the captured image may vary).

NOTE

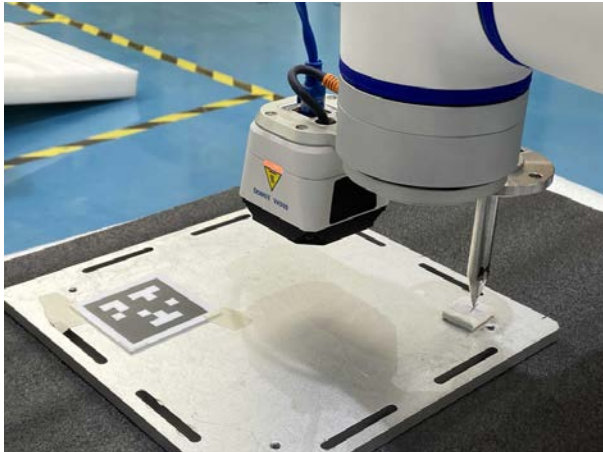
You can use the 2.5D positioning code to create a coordinate system and verify whether the positioning code can still be recognized. If the code is not recognized, lower the robot arm. **The farther the first photo point is from the second photo point, the lower the positioning accuracy. Please adjust accordingly.**

- ② Go to the "Points" page, save this point and name it "2_5D_FirstPhotoPt").

Programming *		Points *		
Mode:		Joint	Linear	 Run to
NO.	Point	Alias	User	Tool
1	P1	2_5D_AutoCalPhot...	0 ▾	0 ▾
2	P2	2_5D_PhotoPt_Base	0 ▾	0 ▾
3	P3	2_5D_SecPhotoPt	3(C	0 ▾
4	P4	2_5D_FirstPhotoPt	0 ▾	0 ▾

Step 3: (Optional) Save the target teaching point (base coordinate system)

- ① In the "Jog" page, set both the User Coordinate System and Tool Coordinate System to "0". Adjust the robot arm's posture so that the needle tip aligns with the center of the cross ("+") on the metal plate. Add this point and name it "TargetPosition_Base".



- ② Raise the robot arm by increasing the Z-value so that the needle tip is just above the target point, save this point as a transition point, and name it "TransPtAboveTgt_Base".

Programming *		Points *		
Mode:		Joint	Linear	 Run to
NO.	Point	Alias	User	Tool
1	P1	2_5D_AutoCalPhot...	0 ▾	0 ▾
2	P2	2_5D_PhotoPt_Base	0 ▾	0 ▾
3	P3	2_5D_SecPhotoPt	3(C✓	0 ▾
4	P4	2_5D_FirstPhotoPt	0 ▾	0 ▾
5	P5	TargetPosition_Base	0 ▾	0 ▾
6	P6	TransPtAboveTgt_...	0 ▾	0 ▾

- ③ If there are multiple positioning points, repeat the teaching process for each one.

NOTE

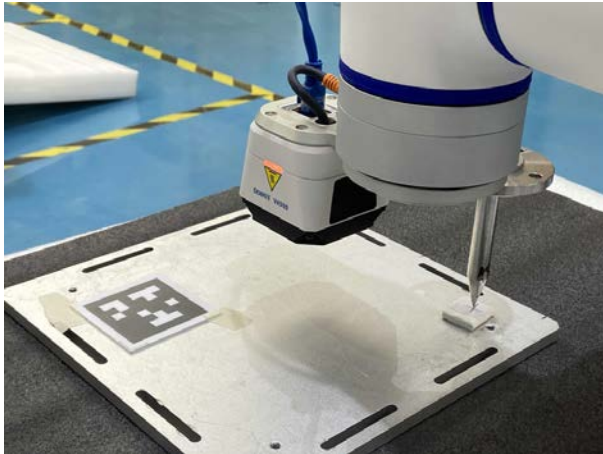
The purpose of saving teaching points in the base coordinate system is to allow fine adjustments when the metal plate or robot base remains fixed, but Cam_user has been updated. If the target teaching points appear inaccurate after the update, the base coordinate system points can be used for correction.

Step 4: Save the target teaching point (Cam_user)

NOTE

You can use the points in the base coordinate system, switch the User Coordinate System directly to "Cam_user", and save the new points.

- ① In the "Jog" page, set the User Coordinate System to "Cam_user" and the Tool Coordinate System to "0". Adjust the robot arm's posture so that the needle tip aligns with the center of the cross ("+") on the metal plate. Add this point and name it "TargetPosition1".

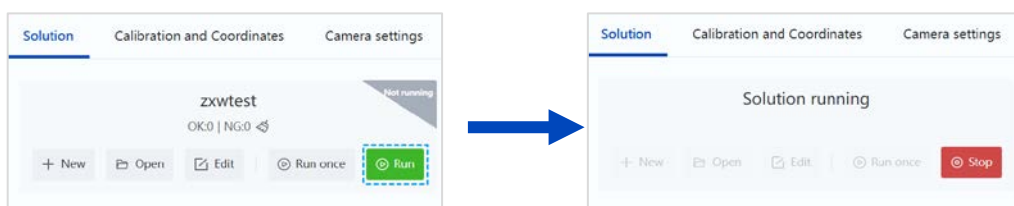


- ② Lower the Z-value so that the needle tip is just above the target point, save this point as a transition point, and name it "TransPtAboveTgt_1".
- ③ If there are multiple positioning points, repeat the teaching process for each one. By following this process, all points in the Cam_user coordinate system will be fully defined.

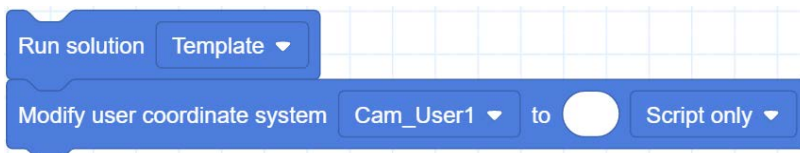
5	P5	TargetPosition_Base	0	0
Optional				
6	P6	TransPtAboveTgt_...	0	0
7	P7	TargetPosition1	3(C✓)	0
Required				
8	P8	TransPtAboveTgt_1	3(C✓)	0

6. Run the actual solution

On the "Solution" page, click the "Run" button to execute the vision solution.



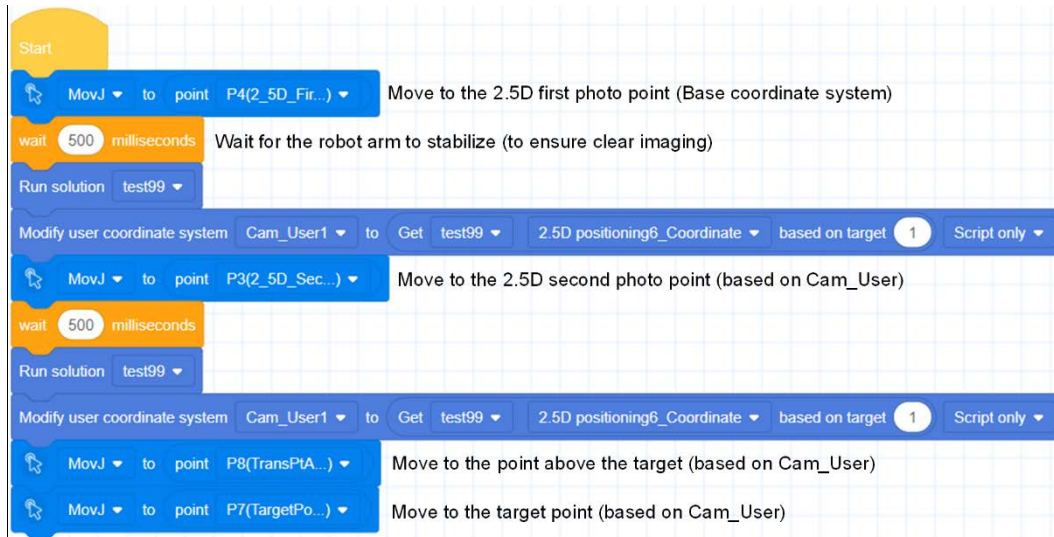
If the vision solution is not running, the corresponding blocks in Blockly programming will not be able to display or retrieve solution information.



7. Build the blocks

During testing, if using an AGV, move the AGV. If the robot arm is fixed, move the 2.5D positioning code and the plane where the target object is located. Check whether the positioning is accurate.

This is the most basic demo for applying 2.5D positioning. More complex applications are built based on this demo.



NOTE

Principle of two-step capture for positioning: When the camera is farther from the positioning code, the positioning accuracy is lower, but the camera's field of view is larger, allowing the positioning code to be detected over a wider area. When the camera is closer to the positioning code, although the positioning code may exceed the field of view, the positioning accuracy improves. Therefore, the first photo point can be set at a farther distance to initially update the coordinate system. Then, based on this updated coordinate system, the camera can move closer to the positioning code. The second photo point then updates the coordinate system with higher accuracy.

In actual application, it is recommended that the second photo point be the same as the point used when creating the 2.5D coordinate system. The height of the first photo point can also be set to be the same as that of the second photo point, ensuring the positioning code remains within the camera's field of view.

9.2 2D positioning case

9.2.1 Technical overview

2D positioning achieves the conversion from image coordinates to physical coordinates on a plane through 2D calibration (where the distance between the camera and the plane cannot be changed and the camera lens must be parallel to the target plane). Based on template matching, it then obtains the coordinates of objects with specific shapes within the camera's field of view. The object can then be directly picked or used to create a coordinate system, where objects within that system can be positioned.

The 2D solution locates the target objects/markers in the workspace by identifying them. The output coordinates only include the XY values and the RZ angle.

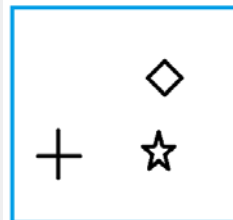
Based on the difference in recognized objects, it can be divided into two types:

- In indirect positioning, the recognized "mark" can be a marker of various shapes.
- In direct positioning, it is suitable for unordered gripping or positioning on various planes.

NOTE

For indirect positioning, the relative position between the marker and the positioning target must remain fixed.

For example, in the figure below, the cross ("+") serves as the marker. During the photo detection, the system detects the coordinates of the cross and creates a user coordinate system. The teaching points are used to teach the coordinates for square and star-shaped gripping points. The positional relationship between them remains unchanged.



• Applicable scenarios

Scenario	Solution
One camera parameter supports clear images for multiple 2D photo points	Create two 2D coordinate systems and add two 2D positioning modules based on one solution
One camera parameter does not support clear images for multiple 2D photo points	Create multiple solutions and switch between them through the VX500 SmartCamera plugin
Lighting inconsistency between day and night, with the goal of continuous 24-hour operation	Create multiple solutions and switch between them through the VX500 SmartCamera plugin

• VX500 underlying logic

- A solution can only have one camera parameter.
- The "2D calibration file" is bound to the "photo point height".
- The "2D coordinate system" is bound to the "2D calibration file".
- The "2D positioning module in the solution" is bound to the "2D coordinate system".

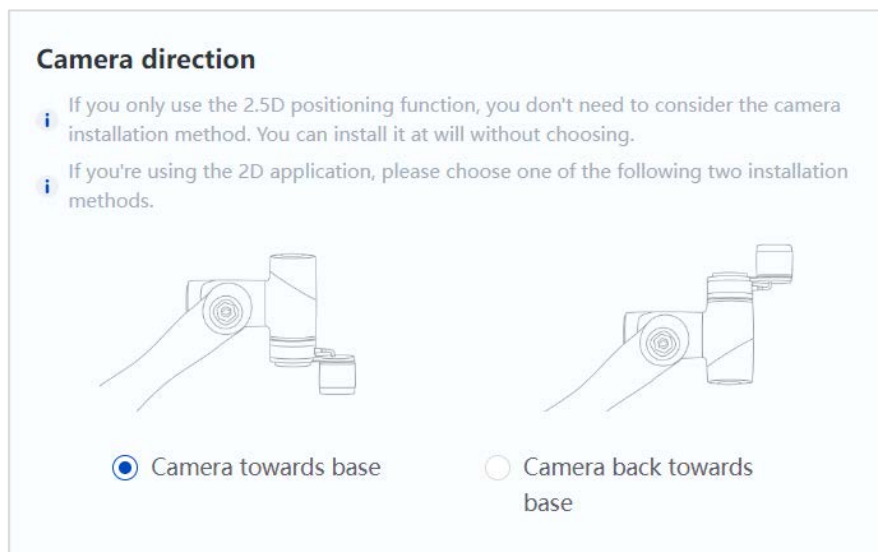
NOTE

When using this calibration file later, it is essential to ensure that the robot arm is always at the specified photo point for each image. If the photo point changes, recalibration is required.

9.2.2 Environmental requirements

• Robot arm installation

If the 2D positioning function is involved, the robot arm can only be installed in the following two ways.



• 2D workplane requirements

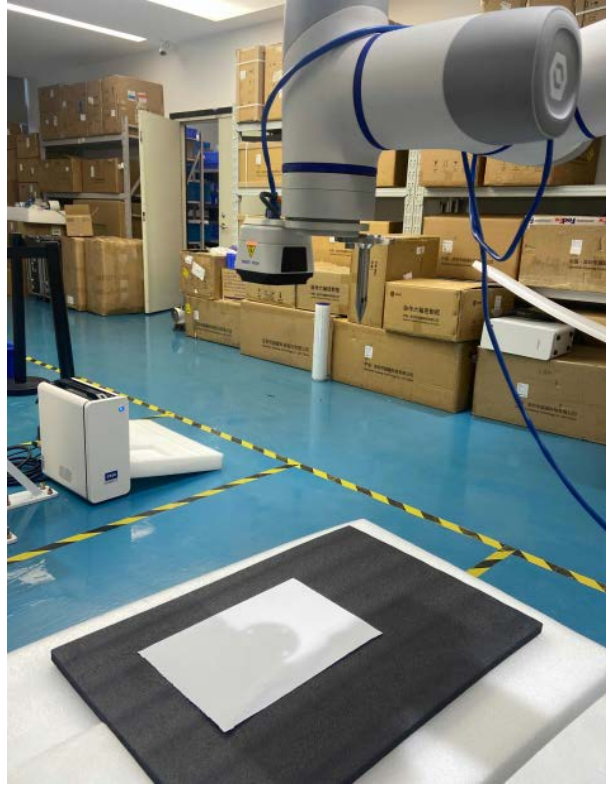
The 2D workplane supports both base coordinate system and custom user coordinate system. However, note that subsequent calibration, positioning, and target point coordinates must remain consistent.

	Scenario A	Scenario B	Scenario C
Software function	2D	2D	2.5D+2D
2D workplane and robot base	Parallel	Tilted	Tilted / Parallel
Relative relationship between robot arm and workplane	Fixed	Fixed	Unfixed (No relationship restrictions between 2.5D positioning code and 2D plane)
Workplane coordinate system	user 0	user 1	user 1
User coordinate system for 2D calibration and positioning	user 0	user 1	user 1
Camuser	2D: Camuser 2	2D: Camuser 2	2.5D: Camuser 1 2D: Camuser 2
2D teaching point	Camuser 2	Camuser 2	Camuser 2

9.2.3 2D positioning process

- **Environment description**

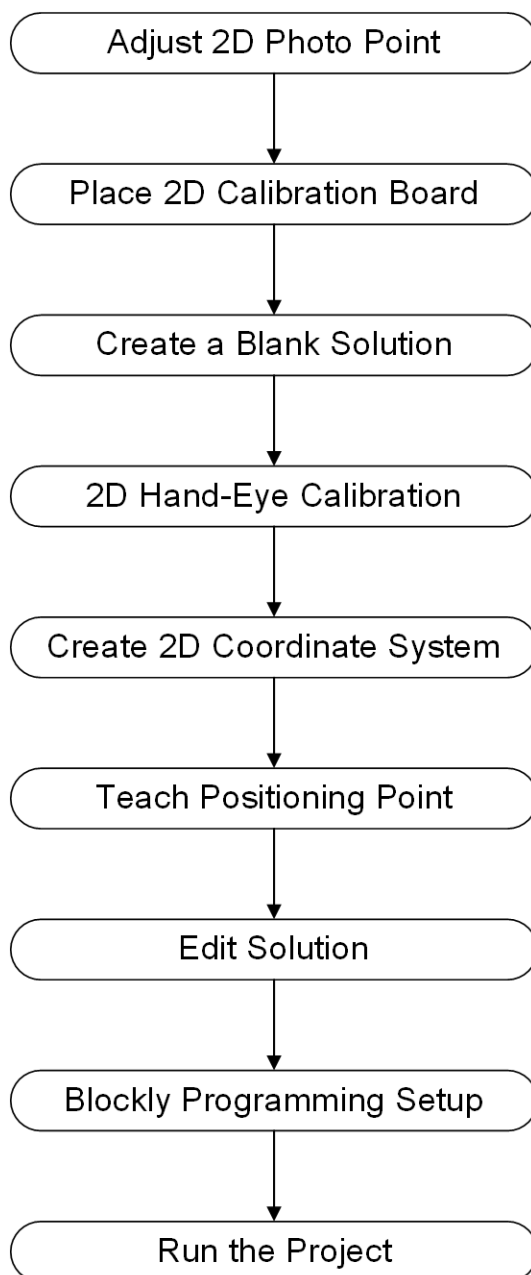
In this demo (as shown in the figure below), the robot arm is equipped with an eccentric needle tip at the end. The goal of this demo is to directly recognize the object and position it accordingly.



NOTICE

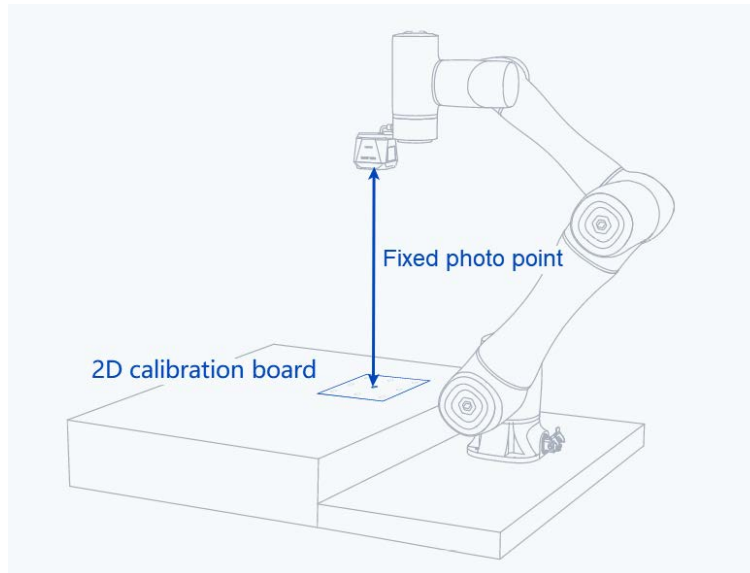
- This setup simulates a real-world application where the gripper and the robot arm's end effector are eccentric. The same process applies if they are concentric.
- The 2D workplane's coordinate system and the robot's base coordinate system are the same.

- **2D positioning workflow diagram**

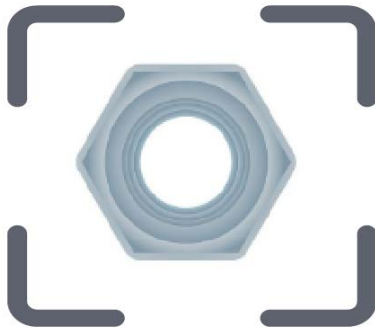


1. Adjust 2D photo point

Step 1: Create a new project in the "Application" page, and create a new point (alias: 2D_PhotoPoint) in the "Points" page. Manually set RX to 180°/-180° and RY to 0°, then move the robot to this point to make the camera lens parallel to the workplane (as shown in the figure below).



Step 2: Adjust the robot's X, Y, Z, and RZ so that the detection area occupies about 90% of the camera's field of view (as shown in the figure below).



Step 3: Update the current point and overwrite it as the "2D_PhotoPoint".

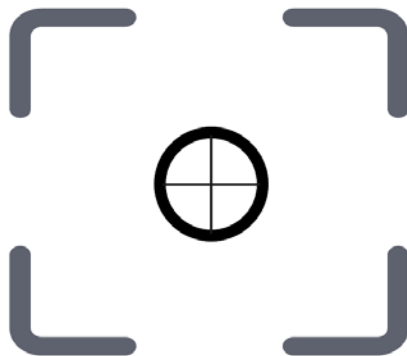


NOTICE

When using the calibration file later, ensure the robot arm is at this point every time you take a photo.

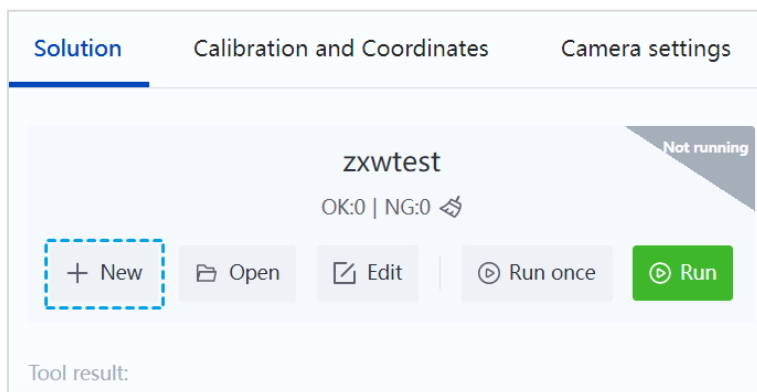
2. Place 2D calibration board

Place the 2D calibration board on the workplane and cover it with white paper, leaving only one circle visible (it's better to avoid using circles for calibration unless the matching point can be precisely set at the center). Ensure the circle is at the center of the camera's field of view.

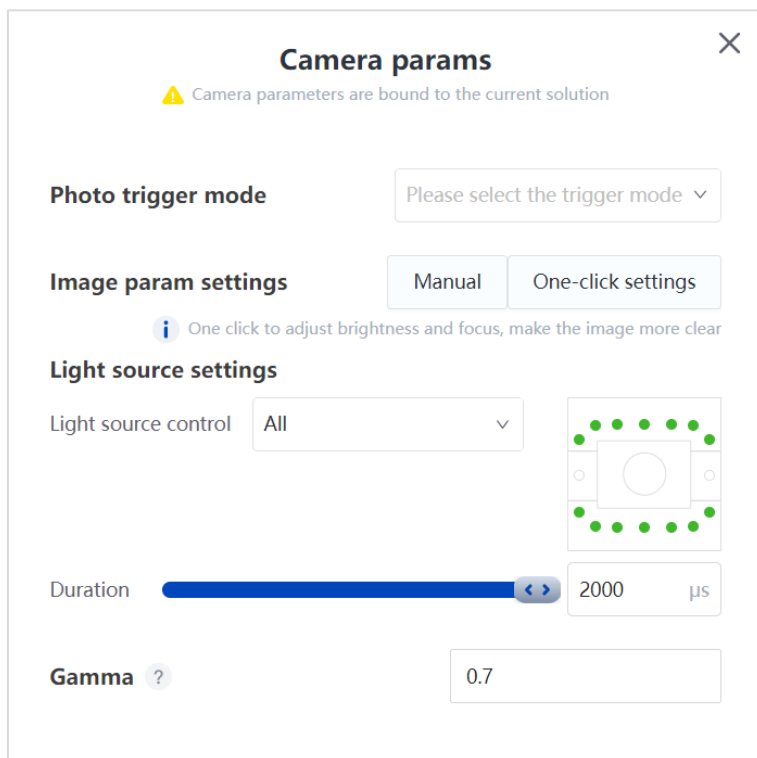


3. Create a blank solution

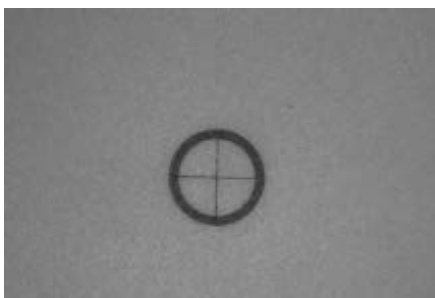
Step 1: Go to the "Solution" homepage and click the "New" button to enter the new solution creation page.



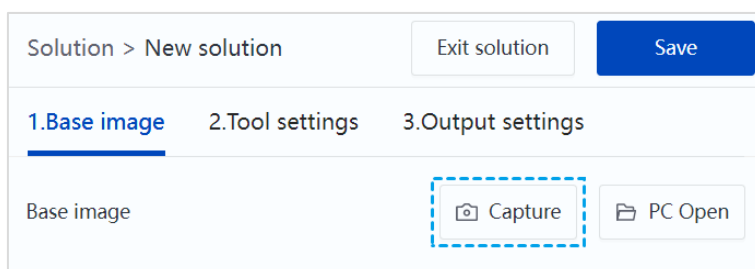
Step 2: Click the "Adjust camera params" button to enter the camera parameter settings page.



Step 3: Click the "One-click settings" button to auto-focus. After the focus is complete, the result should be as shown in the figure below.



Step 4: After adjusting the camera parameters, return to the "New solution > Base image" page, click the "Capture" button to obtain the base image. The image captured by the camera will be the same as the focused image from **step 3**.



Step 5: No tool configuration is required. Go directly to the "Output settings" page, enter the solution name, and click "Save" to complete the creation of the blank solution.

4. 2D hand-eye calibration

Refer to the [2D auto calibration](#) process to complete the calibration and save the calibration file.

2D auto calibration

① Set photo point
② Set template
③ Set offset
④ Collect calibration point

Ensure there are no obstacles within the motion range of robot arm. After starting the collection, the robot arm will automatically move to the 14 points generated by the system to complete the collection.

Collect calibration point
● (14/14)

Result
● Good

Translation error (px):0.957926

Rotation error (px):2.2730179

File name

2D_user0_

.yaml

Cancel
Save calibration file

5. Create 2D coordinate system

Refer to the [Creating coordinate system based on 2D template](#) section to create the 2D coordinate system.

6. Teach positioning point

Step 1: Open the "Jog" page in DobotStudio Pro, set the User Coordinate System to "Cam_User" and Tool Coordinate System to "0". Move the robot arm to align the end tool with the target position that needs to be grabbed. Then, go to "Application > Points" page, click "Add point", and save this point as P2 (Cam_User).

Programming *		Points *								
Mode:		Joint	Linear	Run to	Alias filter		Delete	Cover	+ Add point	
NO.	Point	Alias	User	Tool	X	Y	Z	Rx	Ry	Rz
1	P1		0	0	187.1601	-564.9979	270.5388	179.9915	-0.0001	-173.3080
2	P2		3[C]	0	6.3936	65.3028	270.5388	179.9915	-0.0001	-173.3023
3	P3		3[C]	0	6.3936	65.3028	270.5388	179.9915	-0.0001	-173.3023

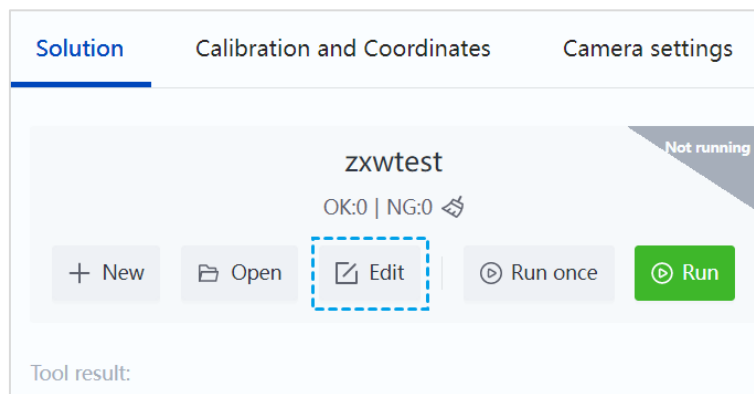
Step 2: Raise the robot arm and add an intermediate point P3 (Cam_User) above point P2.

NOTE

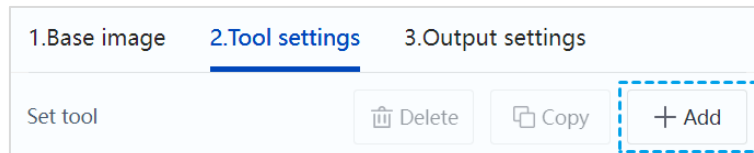
Please add the intermediate transition points based on the actual application, and choose whether the transition point uses the base coordinate system or Cam_User.

7. Edit the solution

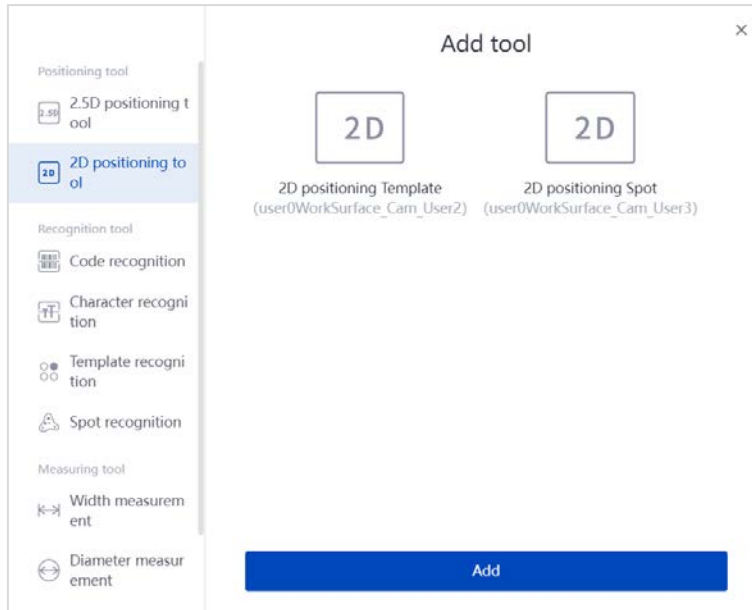
Step 1: Go to the "Solution" page, click the "Edit" button to enter the "Edit Solution" page.



Step 2: Go to the "Tool settings" page, click the "Add" button to open the "Add tool" page.



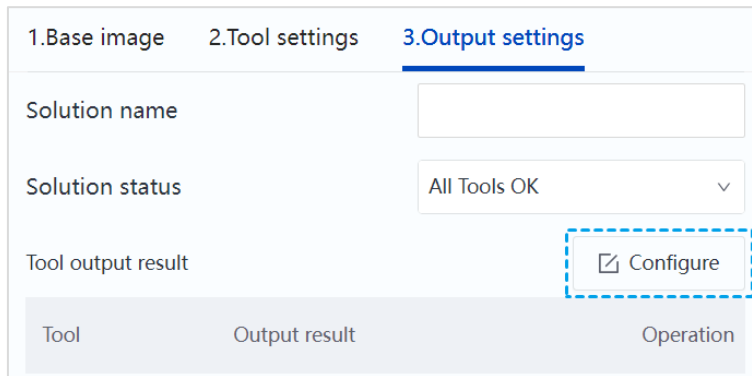
Step 3: The left navigation window will show the 2D positioning tool. Select it and click "Add" to complete the addition of the 2D positioning tool.



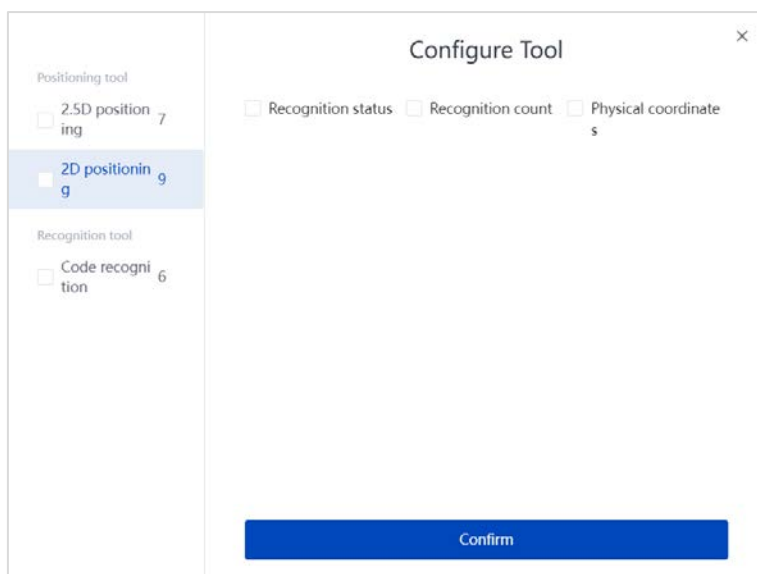
NOTE

This module is bound to Cam_user (index). The newly created Cam_user will be saved to the current solution.

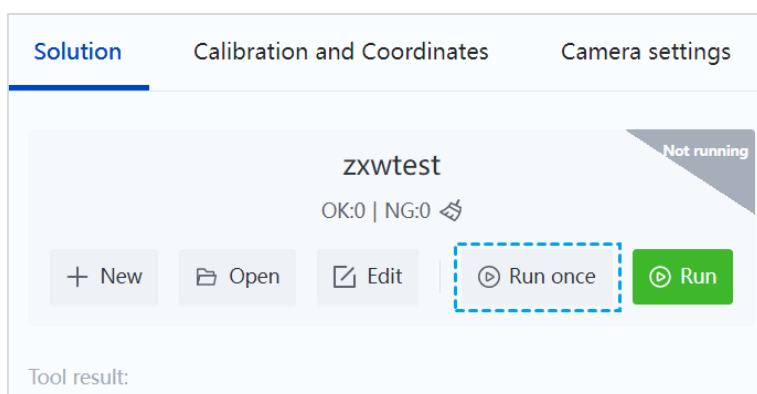
Step 4: Go to the "Output settings" page, click the "Configure" button, and the tool configuration interface will pop up.



Step 5: Select the data you need to output, and click "Confirm" to return to the "Output settings" page. Click the "Save" button to finish editing the solution.

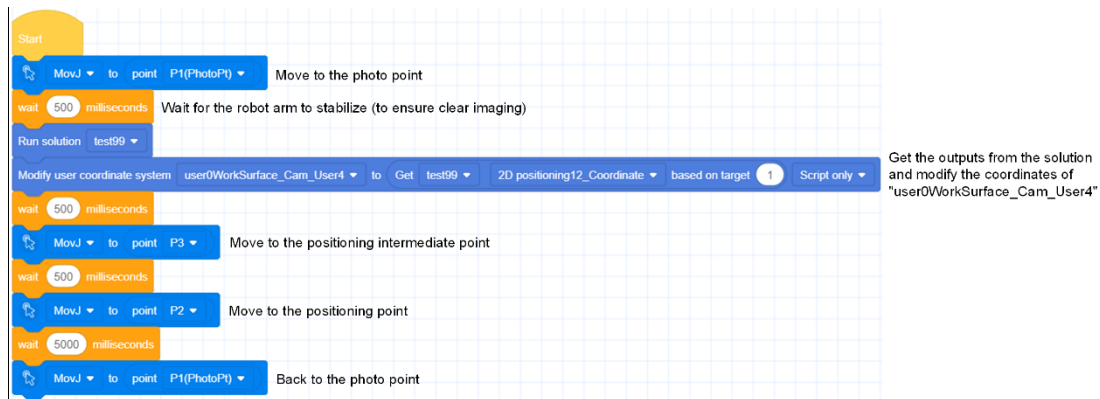


Step 6: On the "Solution" page, click the "Run once" button, and the output result configured in **step 5** will be displayed.



8. Build the blocks

The figure below shows the most basic blockly project.



- P1 is the 2D photo point, which must be the same as the photo point used for 2D calibration and for creating the 2D coordinate system.
- "user0WorkSurface_Cam_User4" is the 2D coordinate system based on template matching. When running the program, the robot will update the coordinate system based on the matched template (the template used when creating the 2D coordinate system).
- P2 is the workpiece picking point. When teaching the point, you need to select the "user0WorkSurface_Cam_User4" coordinate system. In actual application, you can use the workpiece itself as a template or select another feature-rich object as a template, as long as the relative position between the workpiece and the template remains fixed.

9.3 2.5D and 2D linkage positioning case

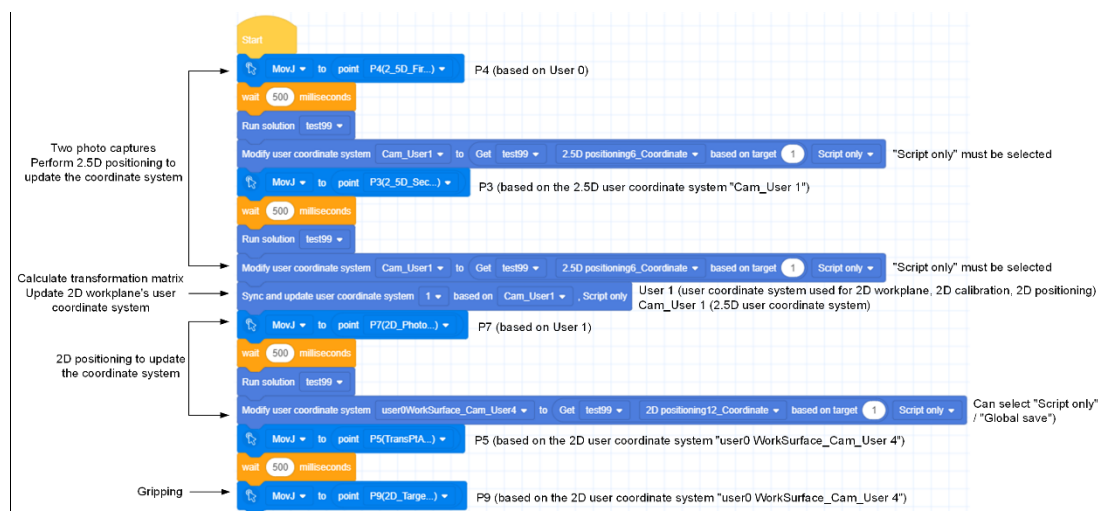
9.3.1 Technical overview

Since 2D positioning cannot compensate for offsets in the Z-axis, it often results in poor accuracy in AGV and robot arm applications. In this case, 2.5D positioning can be used to compensate for the Z-axis offset. First, update the accurate 2D photo point, then perform 2D positioning. This approach solves the problem and achieves high accuracy.

- **Applicable for AGV+CNC scenarios**

	Other Competitors	Dobot Basic Operation A	Dobot Basic Operation B	Dobot Advanced Linkage Operation
Software function	2D positioning	2.5D positioning	2.5D positioning	2.5D positioning + 2D positioning
2.5D positioning code location	--	CNC machine door	Inside CNC machine	CNC machine door
Accuracy	★	★★★	★★★★	★★★★★
Solution issues	2D cannot solve height deviation caused by AGV	Accuracy decreases as the positioning code is further from the target	The positioning code is inside the CNC machine, requiring air blow to remove dust, which affects cycle time	Solves other solution issues

9.3.2 Linkage positioning Demo



Appendix A Technical specifications

Name	Dobot VX500
Vision tool	
Calibration	2.5D calibration, 2D calibration
Positioning	2.5D positioning, 2D positioning
Measurement	Width measurement, diameter measurement, grayscale measurement
Recognition	Code recognition, character recognition, template recognition, spot recognition
Camera	
Sensor type	CMOS, global shutter
Pixel size	3.2 μm x 3.2 μm
Sensor size	1/1.7"
Resolution	2368 x 1760
Maximum frame rate	30 fps
Dynamic range	71.4 dB
Signal-to-noise ratio	41 dB
Gain	0 dB – 15 dB
Exposure time	16 μs – 1 sec
Pixel format	Mono 8
Color	Mono
Electrical characteristics	
Aviation connector cable	8-Pin M8 Connect camera to robot's tool IO, DI*2, DO*2 (PNP), AI*2 (shared with 485) Output voltage 24VDC
Camera's aviation connector interface	8-Pin M8 Replace the robot's original tool IO, used for other eco-tools, DI*2, DO*2 (PNP), AI*2 (shared with 485) Output voltage 24VDC

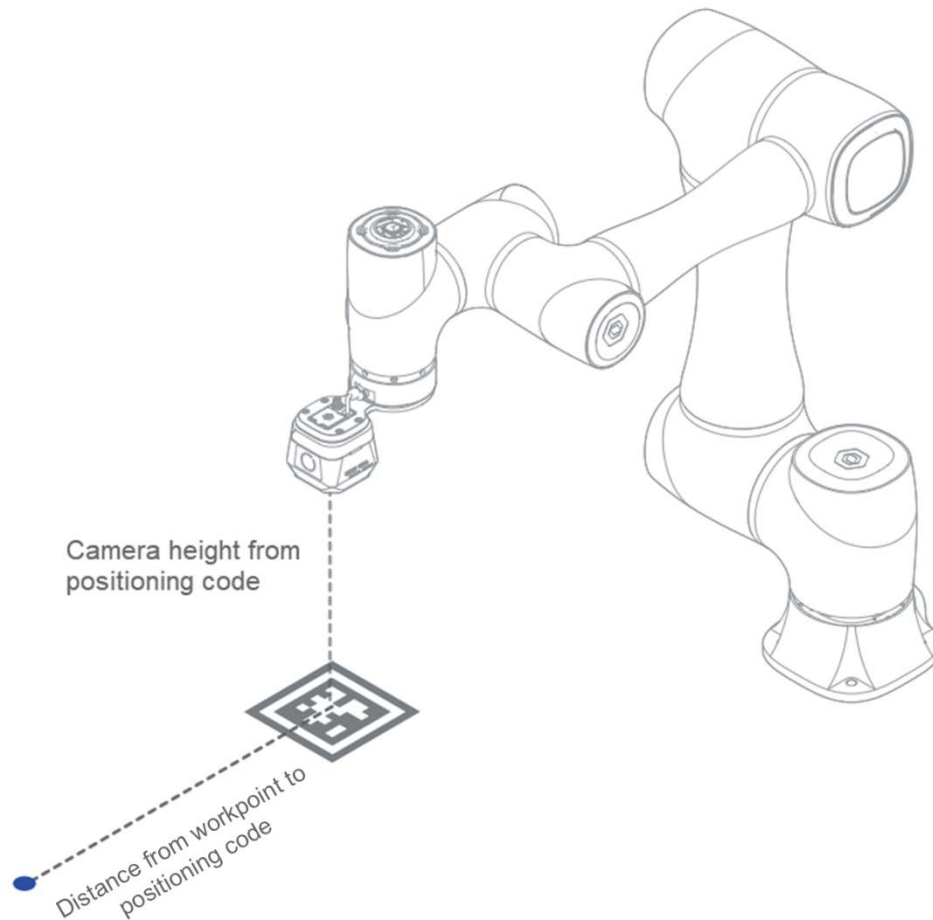
Ethernet port	1, Ethernet (used for camera debugging)
Camera's maximum power consumption	48 W@24 VDC
Structure	
Lens interface	M12-mount, mechanical focus
Focal length	12.4 mm
Light source	14 LEDs: White 2 positioning indicators: Red
Indicator	- PWR: Power indicator - LNK: Network indicator - STS: Status indicator - OK/NG: Result display indicator
Overall dimensions	65.2 mm × 65.2mm × 47mm
Weight	520 g (excluding flange)
IP protection level	IP54 (when lens and cables are correctly installed)
Temperature	Operating temperature: 0 – 50°C Storage temperature: -30 – 70°C
Humidity	20% – 95% RH (non-condensing)
General	
Software	DobotStudioPro, VX500 camera plugin (download from Dobot website)
Operation terminal	PC

Final accuracy for robot arm and VX500 positioning

	Camera height from positioning code	150	250
	Horizontal distance from work point to positioning code		
Simulated moving AMR test (With angle changes)	100	±0.26	±0.37
	300	±0.64	±0.91
Non-moving test (No angle change)	100	±0.058	±0.178
	300	±0.097	±0.237

★ Simulated moving AMR test: In practice, the 2.5D positioning code is rotated, and the maximum rotational posture during testing is $\pm 20^\circ$.

★ This table contains Dobot's test data. In actual applications, environmental lighting, physical characteristics, programming methods, and other factors may affect accuracy variations. The actual values may vary, please refer to the actual settings.



VX500 field of view

Distance from camera lens to workplane	Field of view range
60 mm	37.89 mm x 28.16 mm
3000 mm	1894.4 mm x 1408 mm

