

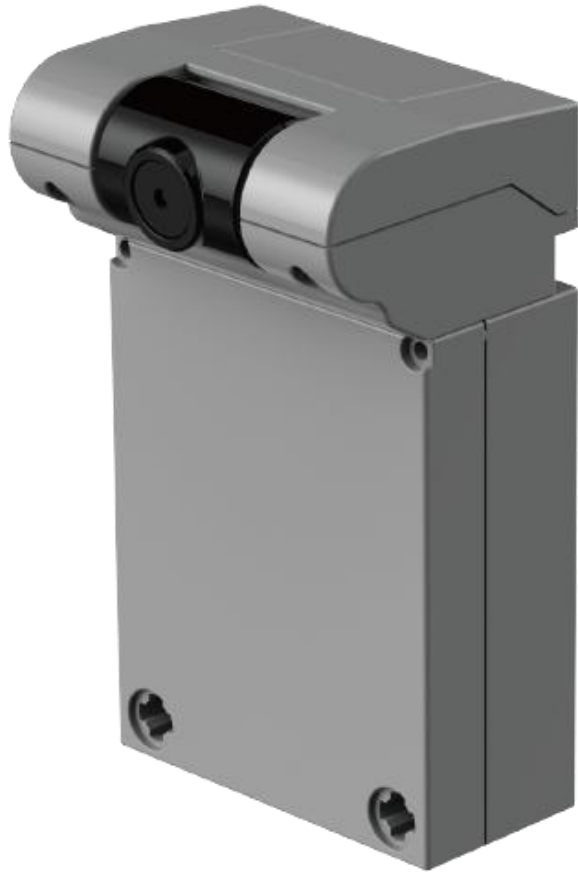


Nashenbot Training

--- AI Camera



1.1 Introduction



The roof-mounted camera is an AI hardware component supported by both the AISTEAM and AIBOT series:

Also known as an AI intelligent recognition system, it can detect road conditions from a wide angle, easily performing assisted driving tasks in autonomous driving, including road slope judgment, color recognition, TAG code recognition, and custom template matching; providing students with diverse possibilities for exploring and learning about artificial intelligence in various real-world scenarios.

Field of view: 72°

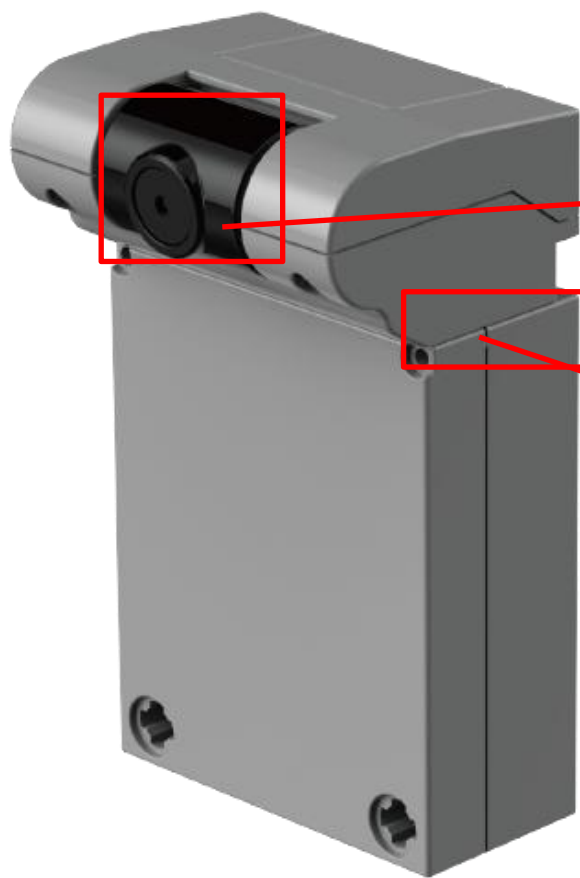
Resolution: 320*240

Recognition method: Multi-scan recognition

Structure: Adjustable height

Data processing: Built-in image data preprocessing function, making communication with the main control unit more efficient and convenient.

1.2 Structure Introduction



Manual viewing angle adjustment: The vertical angle range is 72 degrees. There is a slight resistance every 24 degrees during adjustment, which can be used as a reference for angle adjustment.

Height manually adjustable: The vertical height range is 3cm. Changing the height will change the viewing angle.

Important Notes: All camera recognition relies on a fixed field of view, meaning a fixed height and angle. Any program debugging should begin with checking the camera's field of view. If the program's behavior suddenly changes drastically, remember to check if the angle or height has been incorrectly altered; you can also manually mark it if necessary.

1.3 Ports and Connection



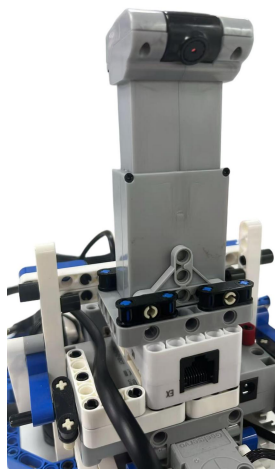
Type-C connector: Primarily used for real-time camera image reading and display, typically used during debugging. It's connected to the cable shown in the right-hand diagram, with the other end connected to a computer's USB port.



PH connector (2.0) 6PIN interface: mainly used for direct connection to the main controller for data transmission.



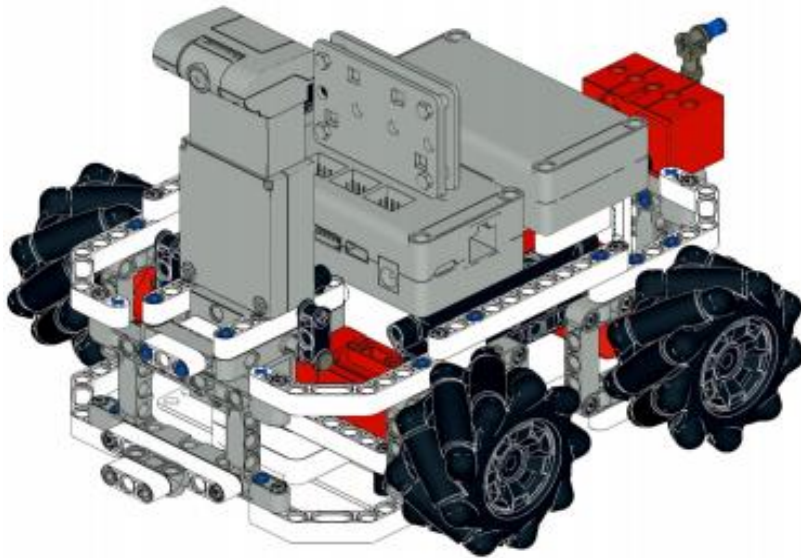
These two holes are used to mount the camera using building blocks. However, in actual assembly, it's recommended not to use only these two holes for mounting; instead, use them in conjunction with a frame beam or similar fixture for a more secure setup.



Important Notes: The two cables above cannot be used simultaneously. Only one cable can be used at a time. Otherwise, it may cause two devices, such as the main controller and the computer used for programming, to simultaneously supply power or transmit data to the camera, which may cause the camera to become unusable or freeze. The camera must be restored to factory settings before it can be reprogrammed and used.

1.4 Building Method

Note: Please refer to the following assembly diagram. Different base models can be replaced.



Building
manual of a camera.pdf

2.1. Usage Procedure

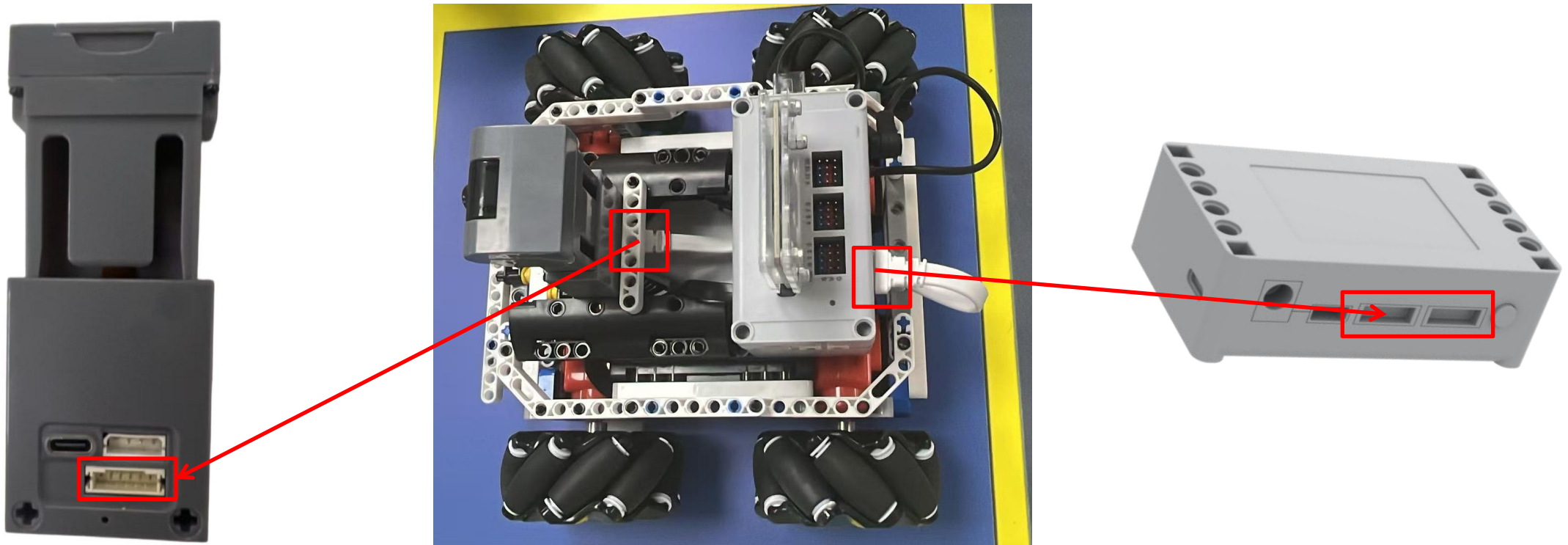
Road slope determination and TAG code recognition are all functions with built-in templates in the camera, so they can be used directly without prior image data acquisition, requiring steps 1, 2, 3, 4, 5, 7, 8, 9, and 10. However, functions such as color recognition and custom template matching involve data that changes in real time, requiring data acquisition first. The camera then automatically converts the data into recognition standards before it can be used, requiring steps 1-10.

The use of the camera mainly involves the following steps:

1. Securely fix the camera to the robot, and connect the PH cable to the main controller.
2. Place the robot in a fixed position (if it is automatically programmed, it needs to automatically move to this position to prevent changes in the viewing angle caused by subsequent position changes).
3. Open the AICODE programming software on the computer.
4. Do not power on the robot; keep it in a power-off state. Connect one end of the USB cable to the back of the camera and the other end to the computer.
5. Open the image threshold editor in the AICODE menu, click "Connect," and you will see the real-time image. Adjust the camera to confirm that a valid image is visible.
6. Debug, collect, and record the required data as needed.
7. Disconnect the USB cable from the back of the camera and connect the cable to the main controller.
8. Power on the main controller.
9. Write and download the program.
10. Test, adjust, and optimize.

2.1. Usage Procedure

1. Securely fix the camera to the robot, and connect the PH cable to the port P1 or P2 of main controller.



2.1. Usage Procedure

2. Place the machine in a fixed position (if it is automatically programmed, it needs to automatically move to this position to prevent changes in the viewing angle caused by subsequent position changes).

Figure 1

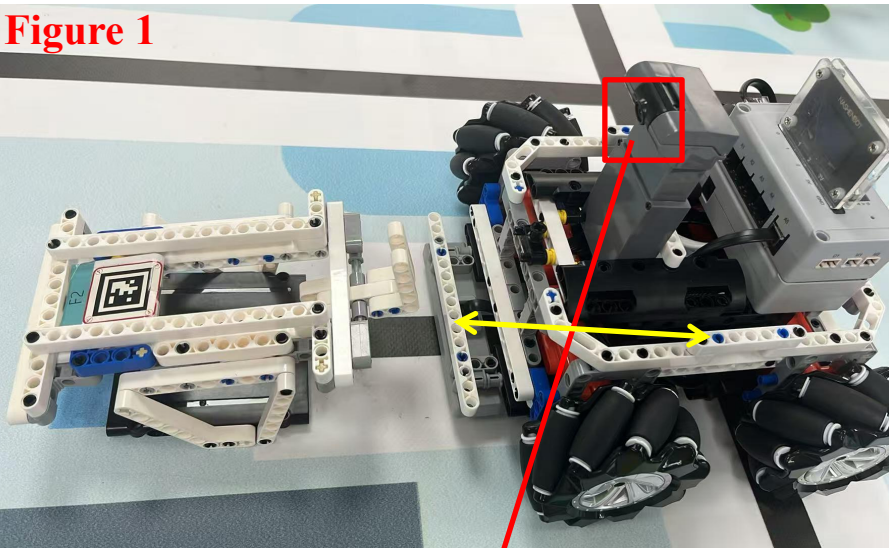
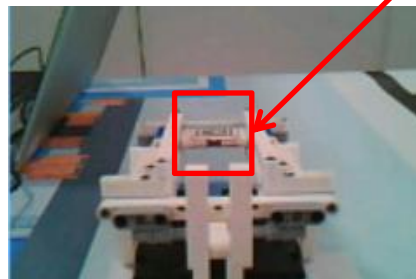
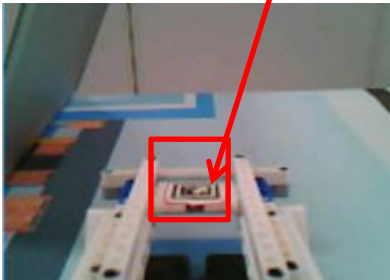
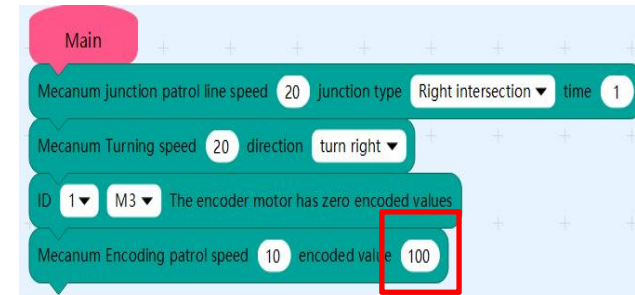
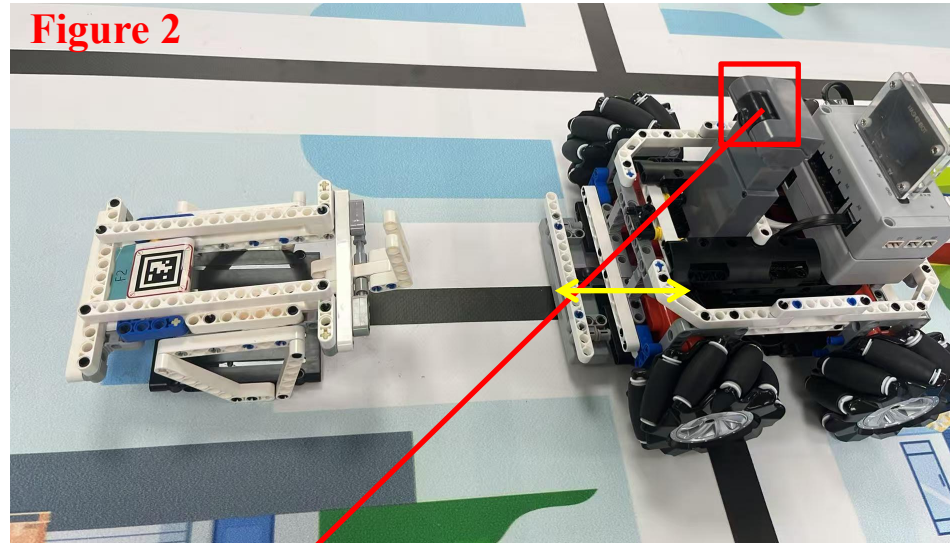


Figure 2

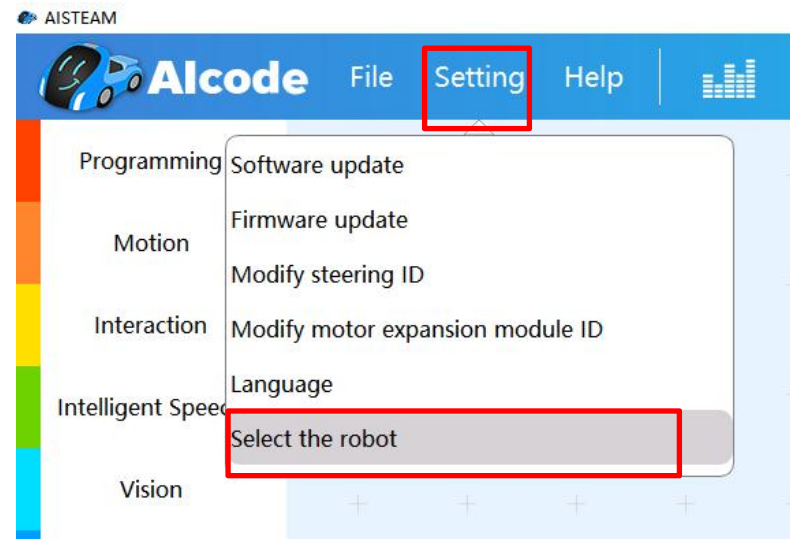


Referring to this case, both Figures 1 and 2 actually use grayscale line following to reach the intersection, then move forward a certain coded value. The camera angle and height are fixed; changing the position and distance completely alters the perspective, rendering valid tag code recognition invalid.

To maintain the fixed distance as in Figure 2, the camera height needs to be increased and the angle changed.

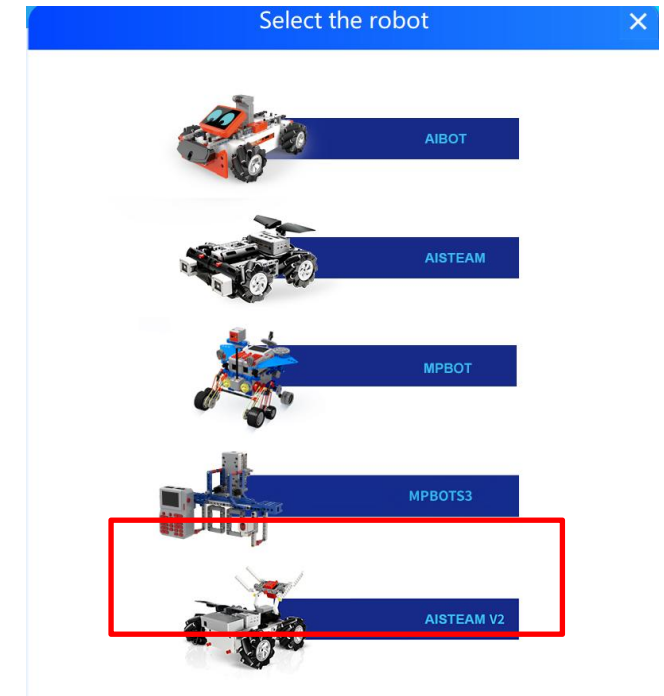
2.1. Usage Procedure

3. Open the AICODE programming software on the computer and select the right robot series.



Step 1: Locate the AICODE programming software on your computer and double-click to open it.

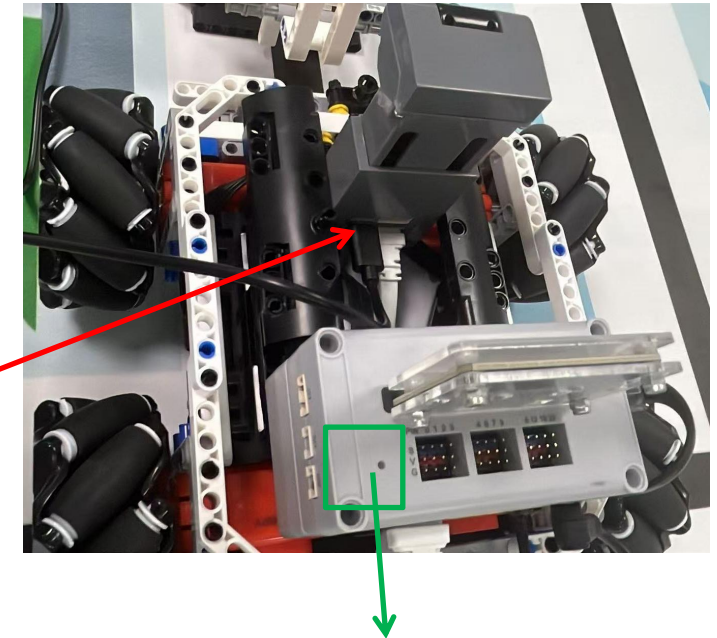
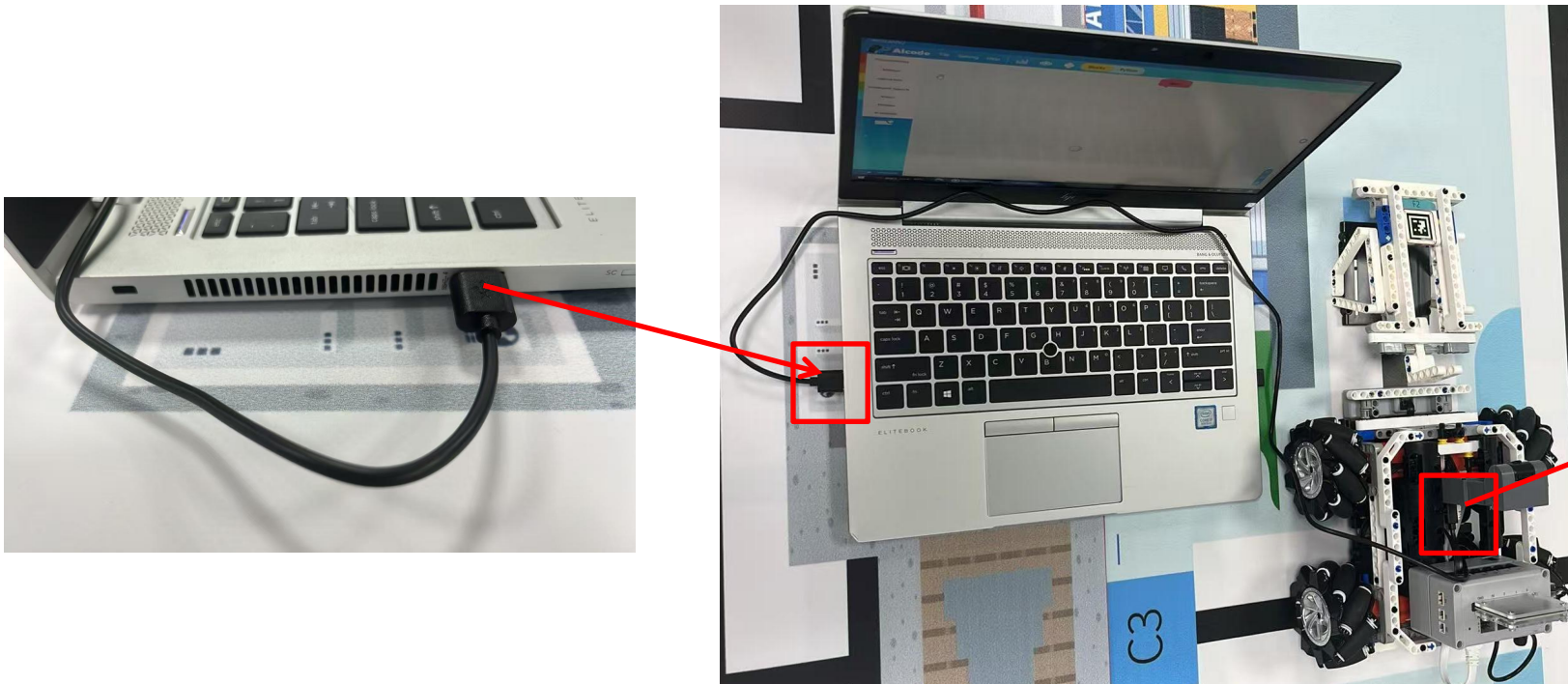
Step 2: Find the robot selection option in the settings.



Step 3: Select the right robot series

2.1. Usage Procedure

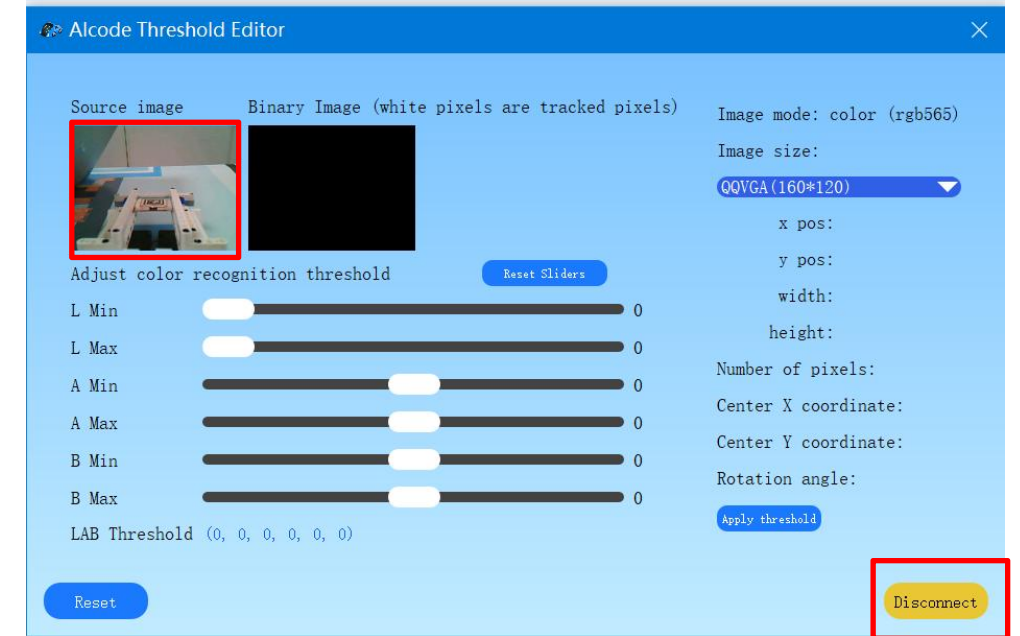
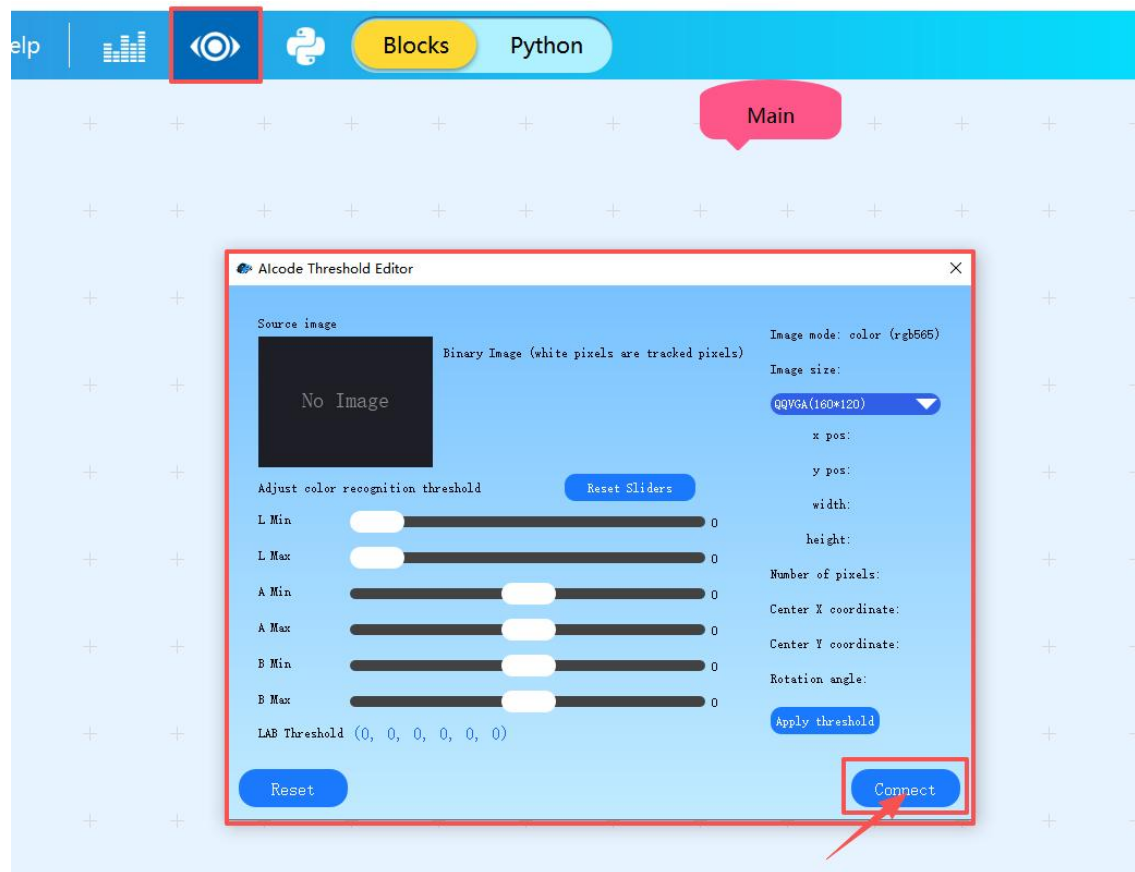
4. Do not power on the robot; keep it in a power-off state. Connect one end of the USB cable to the back of the camera and the other end to the computer.



Note: This main control light will not be on if the machine is not powered on.

2.1. Usage Procedure

5. Open the image threshold editor in the AICODE menu, click "Connect," and you will see the real-time image.

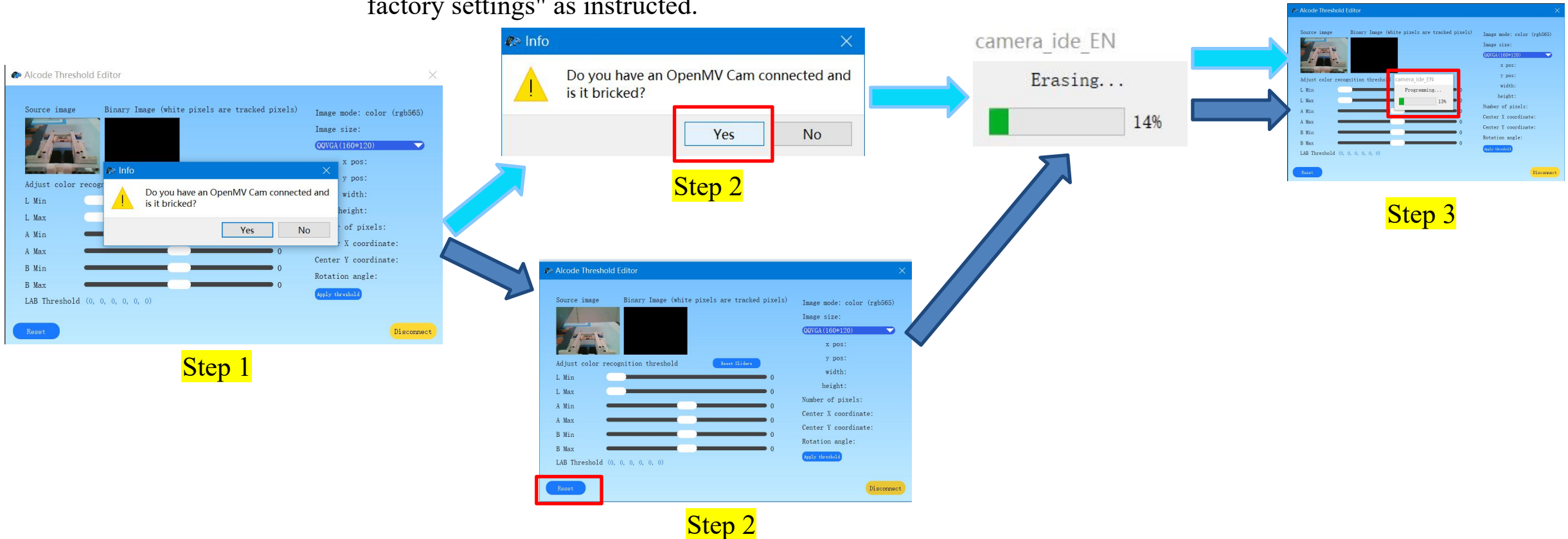


If you can't see it, adjust the viewing angle or height. If you still can't see it after adjusting these two, you'll need to move the machine to a different position.

2.1. Usage Procedure

5. If no image appears, it may be due to previous incorrect operation causing the camera to lag; a factory reset may be necessary.

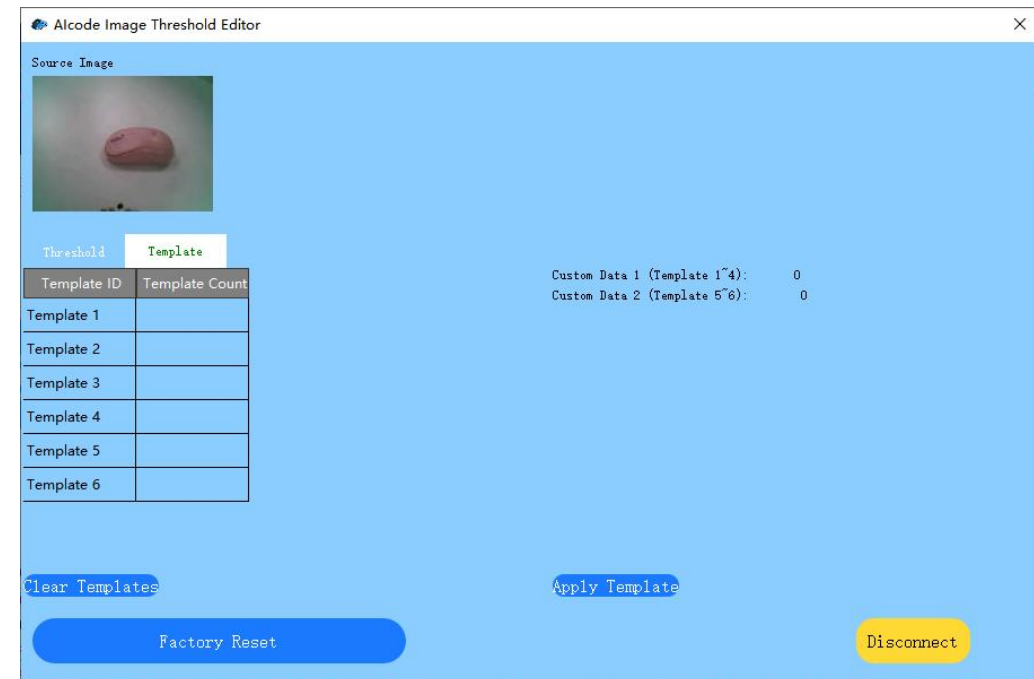
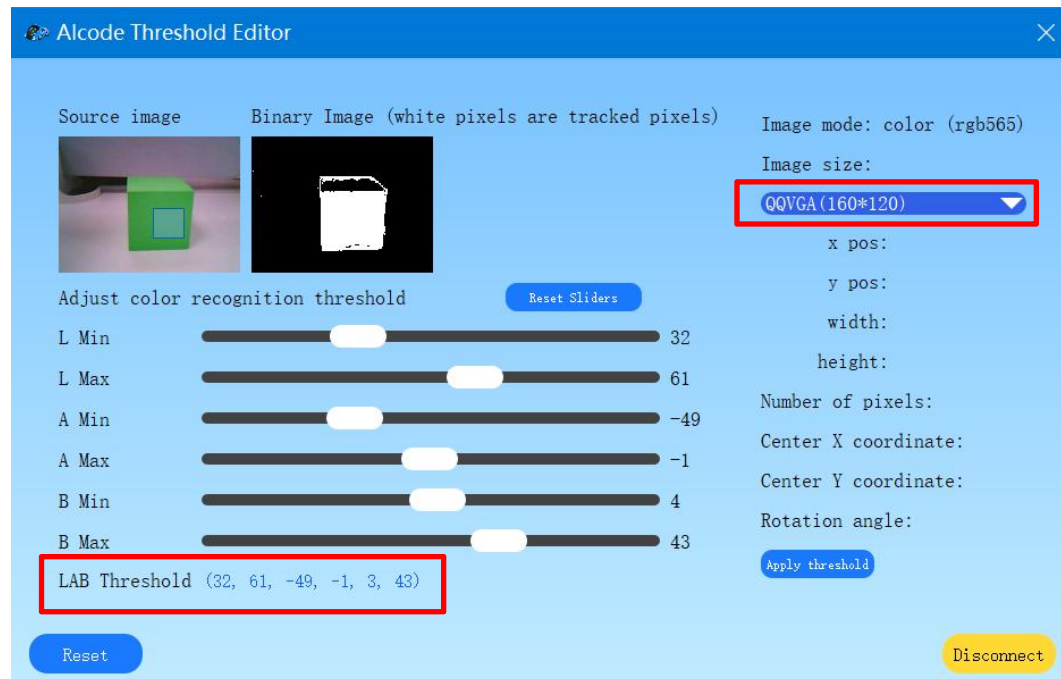
Scenario 1: If this prompt appears, you can click "Automatically restore factory settings" as instructed.



Scenario 2: If this prompt does not appear, you can click to manually restore factory settings.

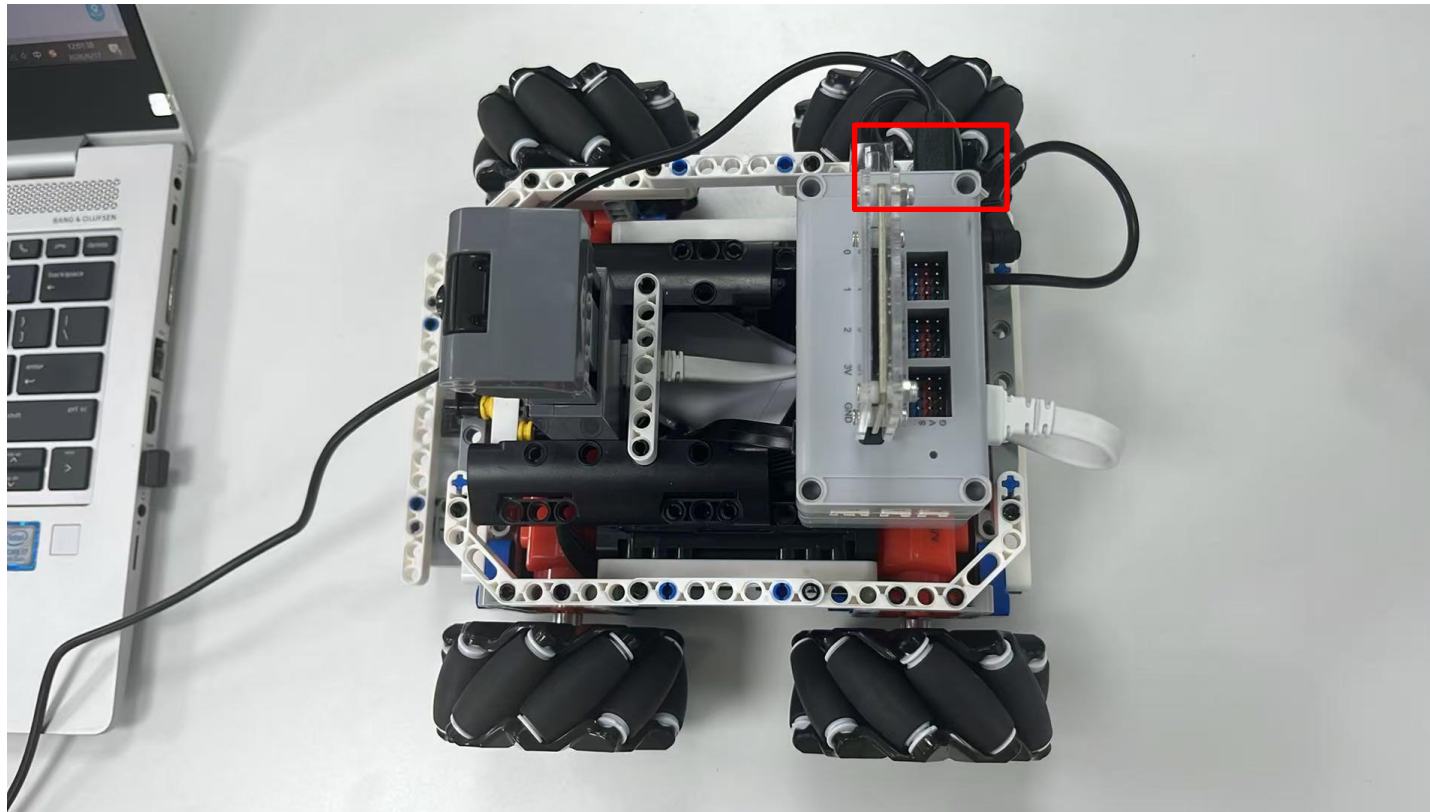
2.1. Usage Procedure

6. Debug and collect and record the required data according to the requirements (color recognition requires obtaining the maximum and minimum values of L, A, and B; template matching requires inputting template images).



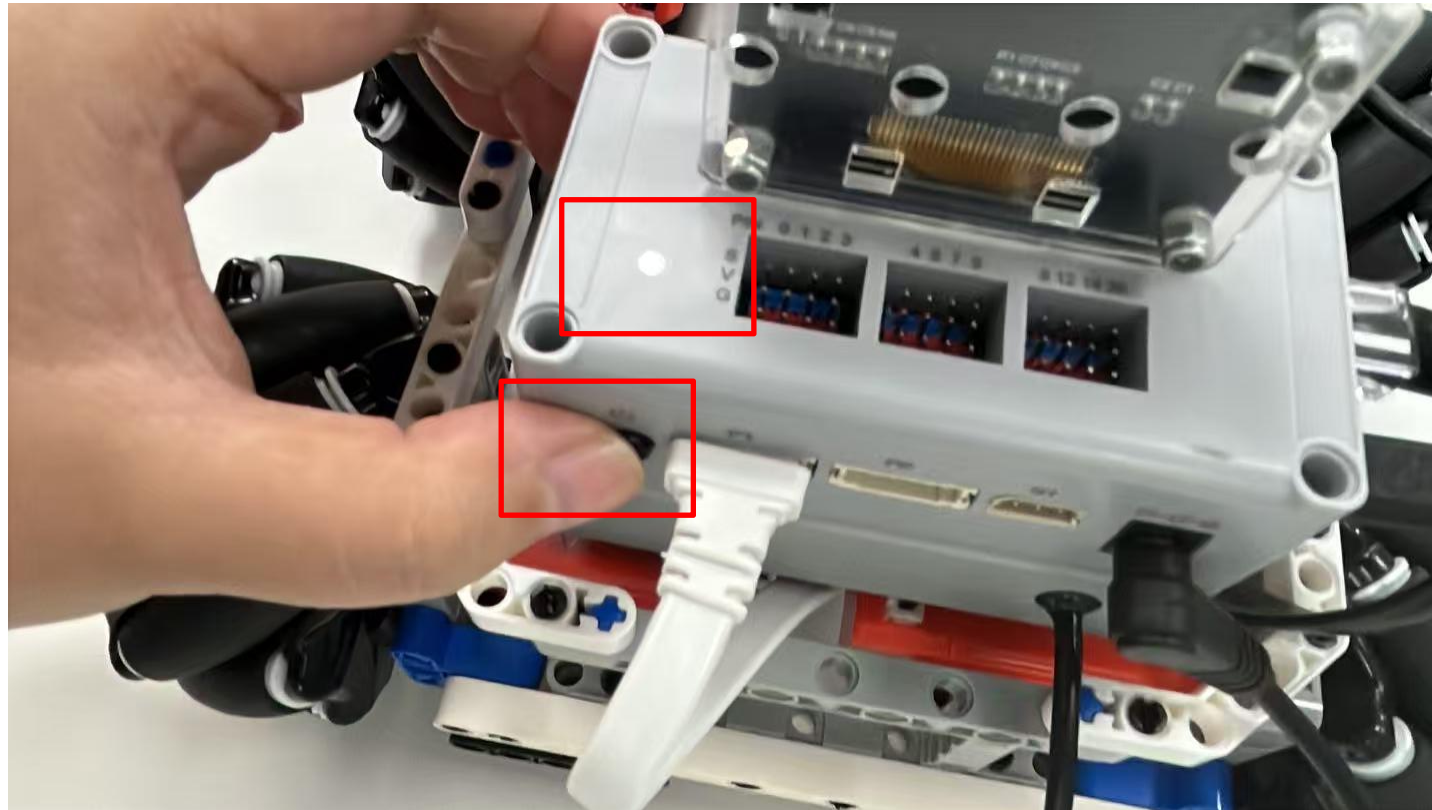
2.1. Usage Procedure

7. Disconnect the USB cable from the back of the camera and connect the cable to the main controller.



2.1. Usage Procedure

8. Power on the main controller (press and hold the black button for 1-3 seconds until the white indicator light illuminates to indicate that it is powered on).



2.1. Usage Procedure

9. Program writing and downloading

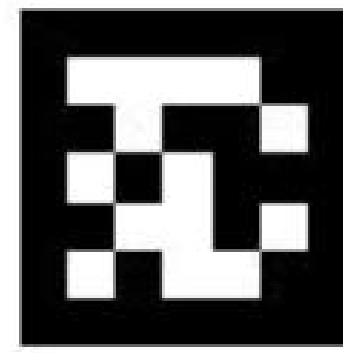
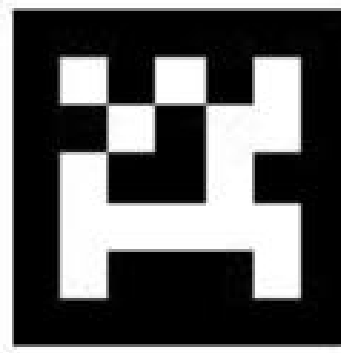
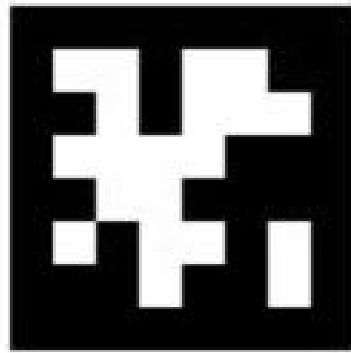
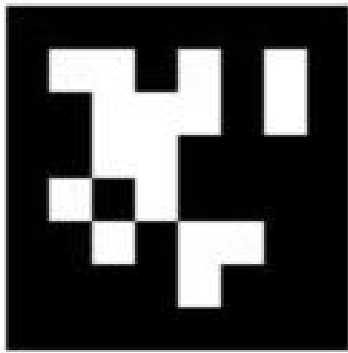
10. Testing, adjustment, and optimization



2.2. TAG code recognition

A tag is an identification label attached to physical or digital content. Through specific codes or keywords, it enables the marking, classification, identification, and rapid tracking of objects. It works similarly to QR codes in our daily lives.

We have our own tag library, and the following are some commonly used tags, corresponding to numbers 1-4, often used for setting random variables in competitions.

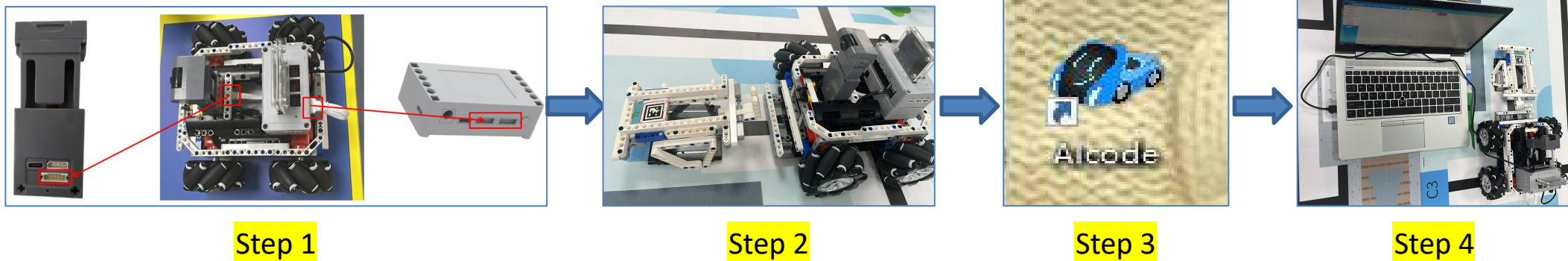


2.2. TAG code recognition

The typical process for tag code recognition is as follows:

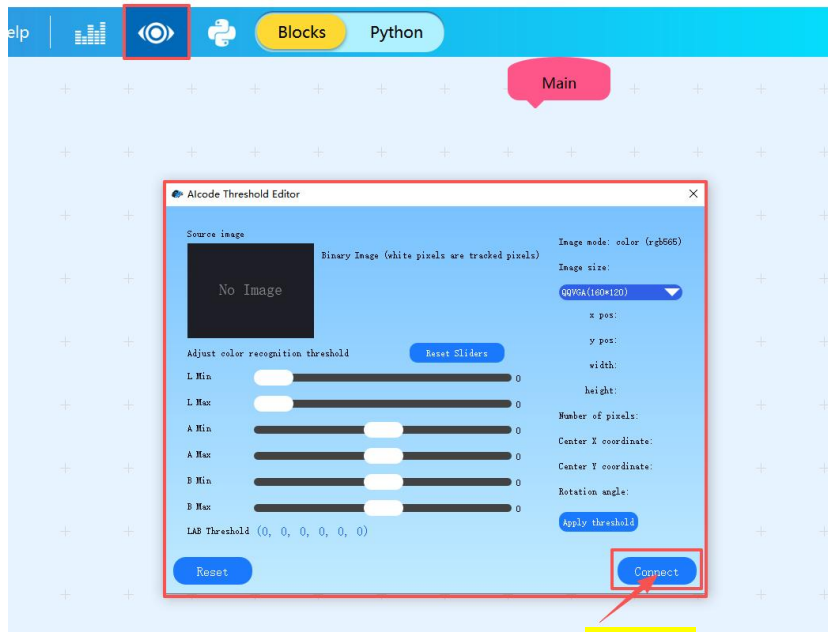
1. Securely mount the camera on the robot and connect the PH cable to the main controller.
2. Place the robot in a fixed position (if it's automatically programmed, the program should automatically move it to this position to prevent changes in viewpoint caused by subsequent positional changes).
3. Open the AICODE programming software on the computer.
4. Do not power on the robot; keep it powered off. Connect one end of the USB cable to the back of the camera and the other end to the computer.
5. Open the image threshold editor in the AICODE menu, click "Connect," and you will see the real-time image. Adjust the camera to confirm that a valid image is visible.
6. Disconnect the USB cable from the back of the camera and connect it to the main controller.
7. Power on the main controller.
8. Write and download the program.
9. Test, adjust, and optimize.

The other steps are the same; let's look at the differences between steps five and eight.

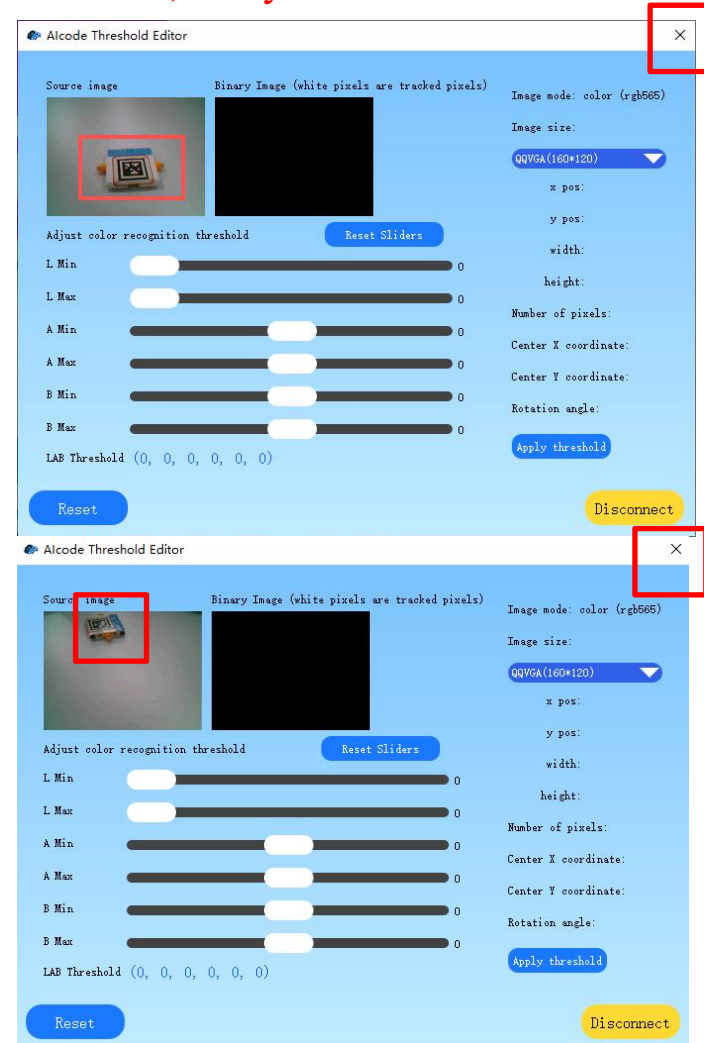


2.2. TAG code recognition

5. Open the image threshold editor in the AICODE menu, click “Connect”, and you will see the live feed. Adjust the camera to confirm that a valid image is visible.



Step 1



This is a valid screen; just confirm it and then close it.

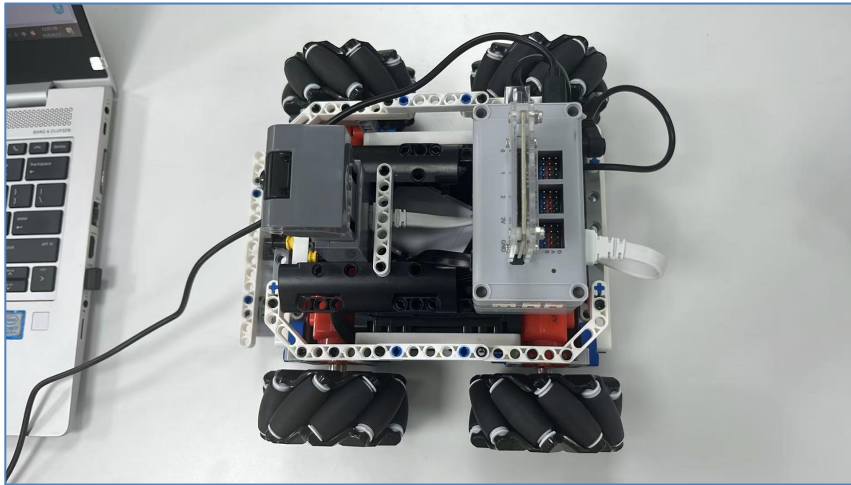
Step 2

This is an invalid view. For example, like this, the viewpoint should be upward and the distance closer. After confirming, close it.

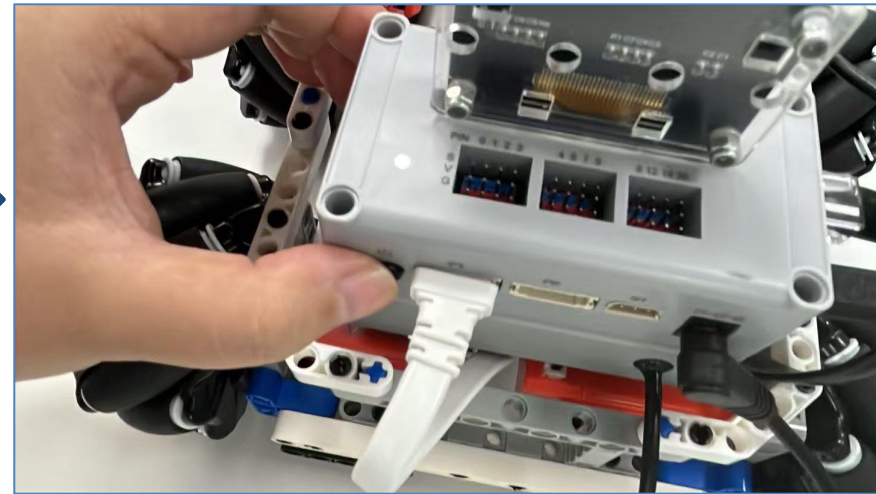
2.2. TAG code recognition

6. Disconnect the USB cable from the back of the camera and connect the cable to the main controller.

7. Power on the main controller.



Step 1

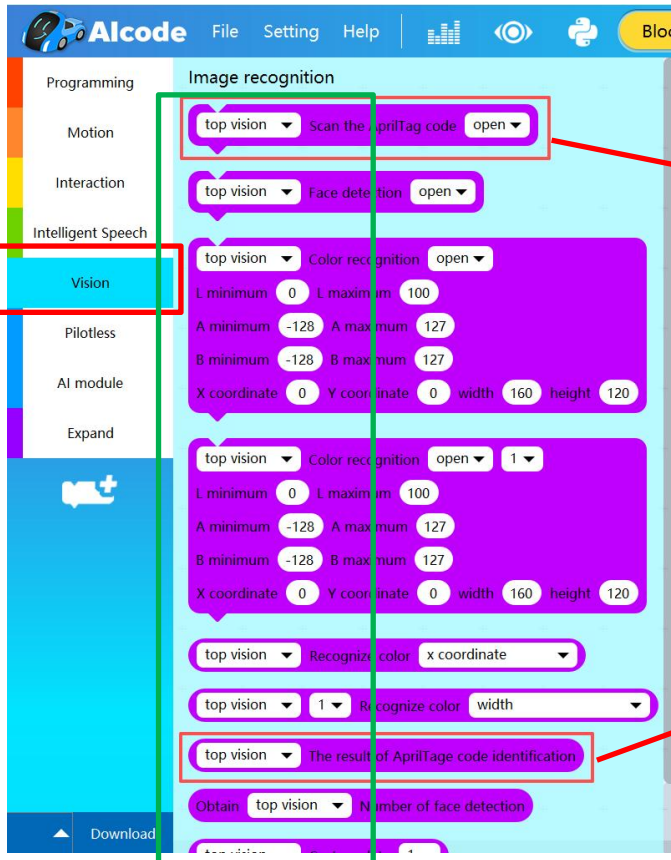


Step 2

2.2. TAG code recognition

8. Program writing and downloading 9. Testing, adjustment, and optimization

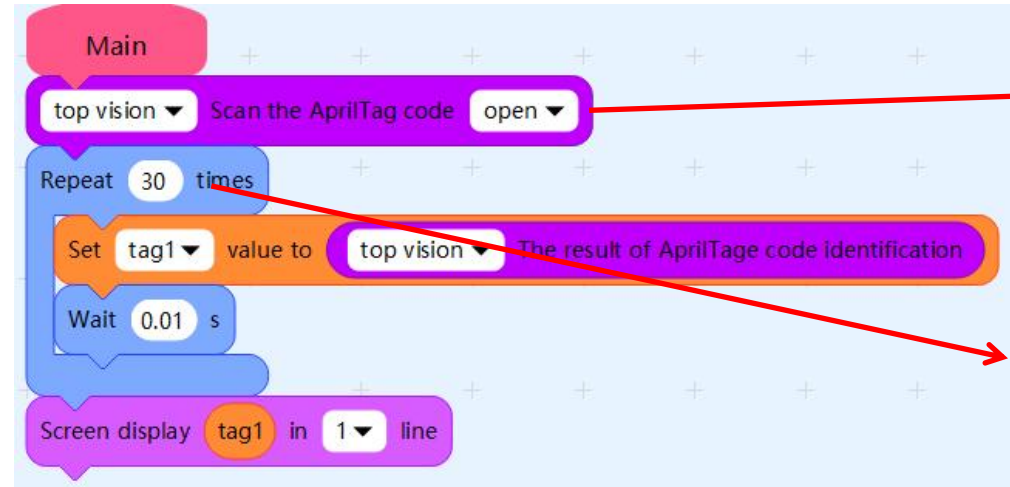
Example program: Recognizes TAG codes at fixed positions and displays them on the screen.



Enable TAG code recognition function

The result of TAG code recognition corresponds to a number, which is often used for judgment.

Step 1:



Enable TAG code recognition function

Repeated identification dozens of times is to prevent misidentification while the vehicle is still moving, thus ensuring greater accuracy.

Step 2:



Download and operate

Step 3:



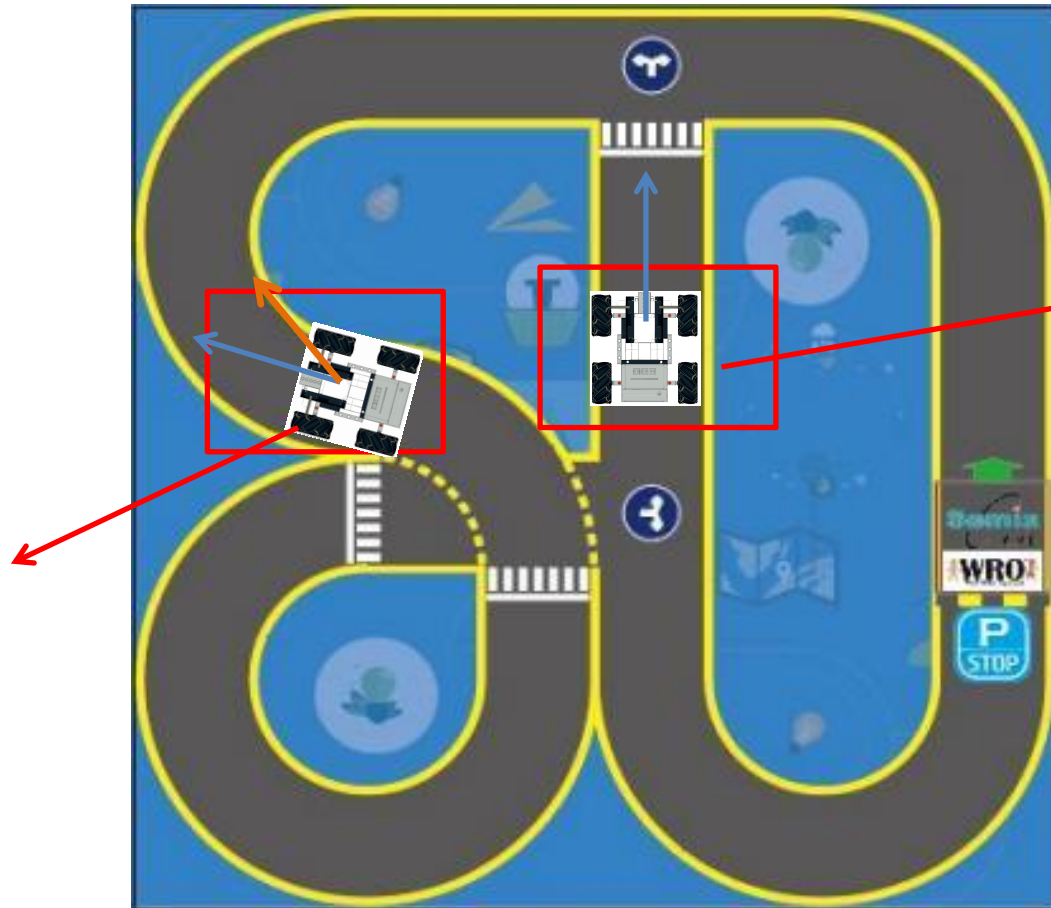
If successful, the corresponding number will be displayed on the screen; if unsuccessful, the position and viewing angle will be checked.

Note: This is a roof-mounted camera. We also have a front-facing camera; the function is the same, but the appearance is different. When programming, be sure to select the corresponding camera.

2.3. Autonomous driving - Road slope judgment

The principle of autonomous driving is that the car adjusts its direction of travel according to the inclination of the lines to ensure that it does not deviate from the guidance of the lines. In the diagram, the slope of the lines refers to the double yellow lines on both sides of the road.

This is a diagonal section; the line slopes to the right. You need to make a slight right turn while moving forward to avoid veering off the path.

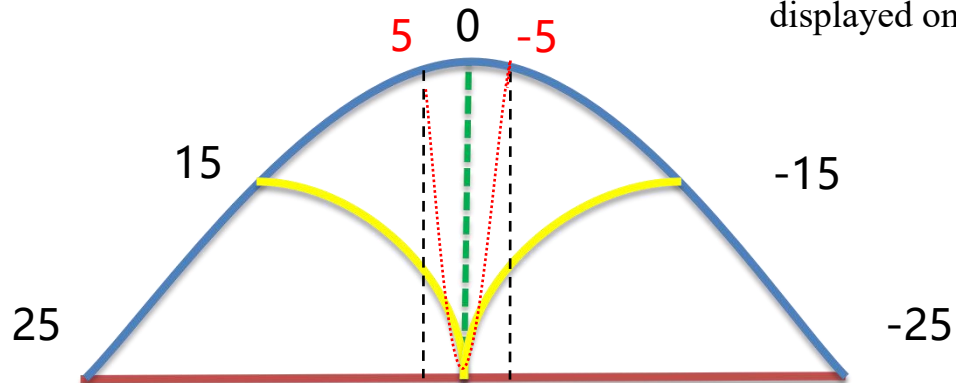


This is a straight section; proceed directly.

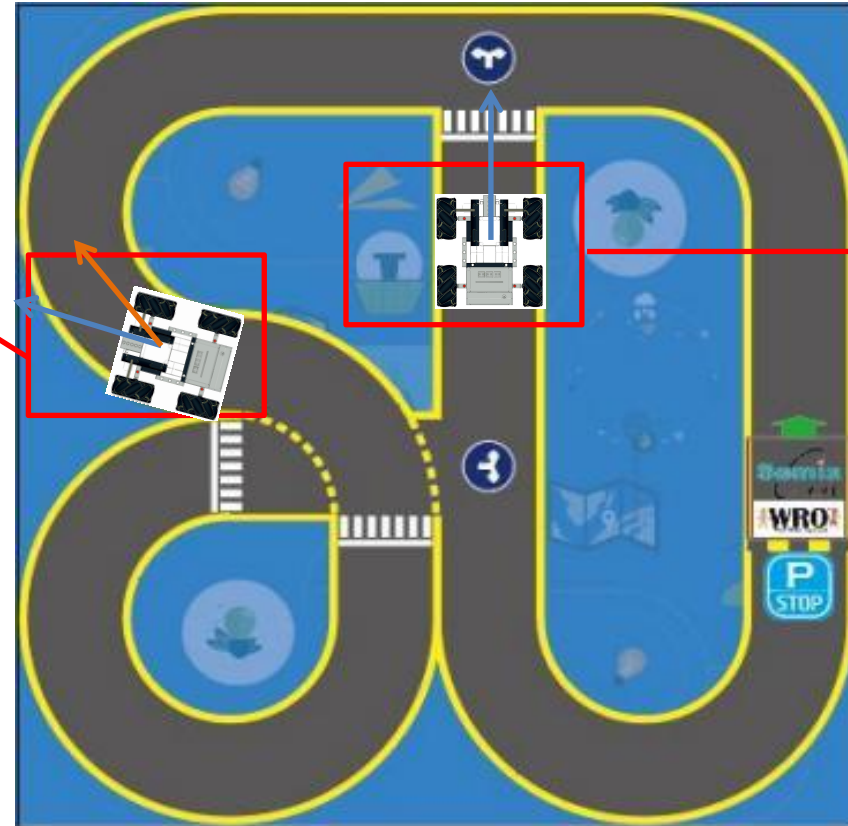
2.3. Autonomous driving - Road slope judgment

The principle of autonomous driving is that the car identifies the degree of inclination of the road, also known as slope. A center position indicates normal tilt (0°), a leftward veer indicates a positive tilt (positive slope, maximum value 25, approximately 75 degrees), and a rightward veer indicates a negative tilt (negative slope, maximum value -25, approximately -75 degrees). Positive and negative values only indicate direction; slopes greater than 75 degrees cannot be automatically corrected by autonomous driving. The car's speed and direction are then adjusted according to the slope in the programming.

Left is positive, right is negative; left is subtractive, right is additive.



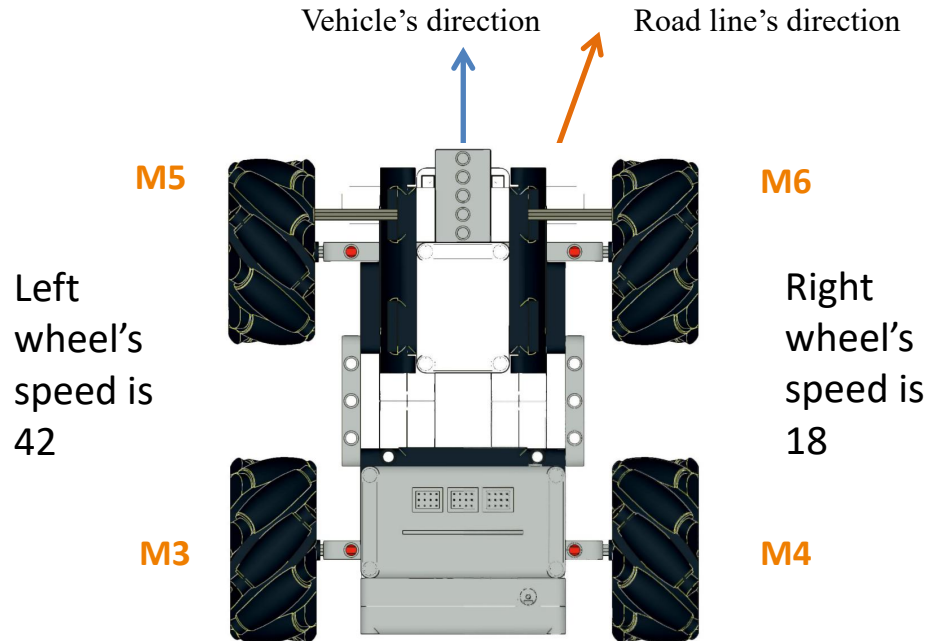
This is a right-sloping area, where the road direction deviates to the right compared to the direction of travel, with a slope of approximately -8 (the specific slope value can be displayed on the screen).



This is a straight section where the direction of travel is the same as the road direction, and the slope is 0.

2.3. Autonomous driving - Road slope judgment

How does the slope correspond to the automatic adjustment of the direction of motion? We have summarized an automatically calculated formula.



Think

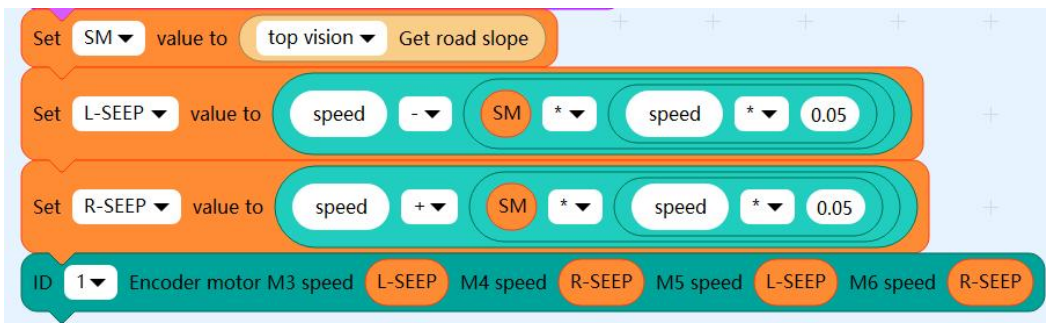
Left wheel = Base speed - Road slope * Kp

Right wheel = Base speed + Road slope * Kp

Assuming the line deviates to the right, the vehicle needs to veer to the right to move forward. The left wheel should move faster, and the right wheel slower. Substituting the base speed of 30 and the slope of -8,

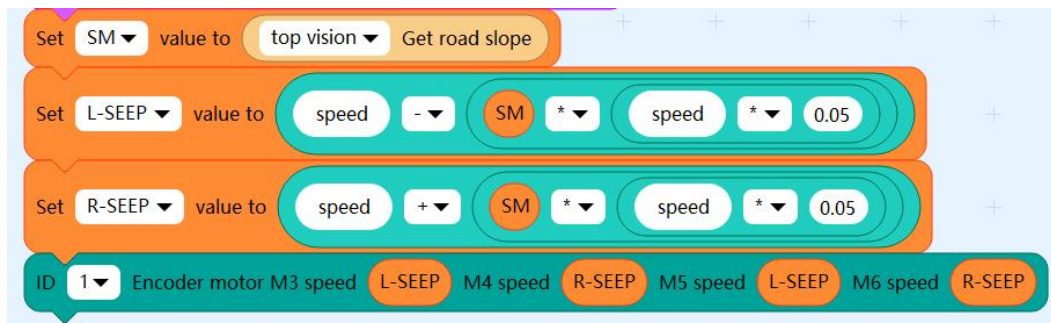
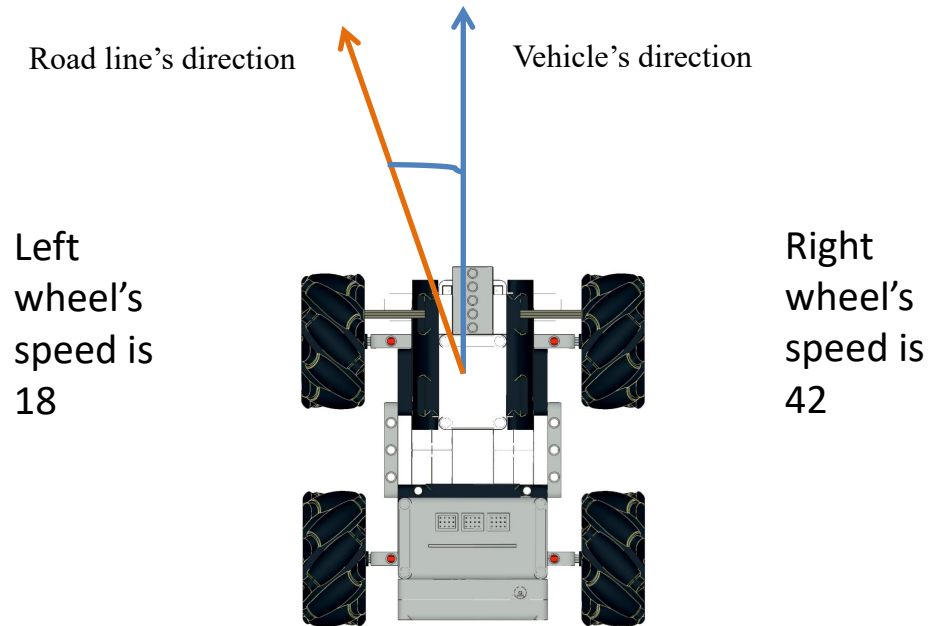
Left wheel speed is: $30 - (-8 \times 30 \times 0.05) = 30 - (-12) = 42$

Right wheel speed is: $30 + (-8 \times 30 \times 0.05) = 30 + (-12) = 18$



2.3. Autonomous driving - Road slope judgment

How does the slope correspond to the automatic adjustment of the direction of motion? We have summarized an automatically calculated formula.



Think

Left wheel = Base speed - Road slope * Kp

Right wheel = Base speed + Road slope * Kp

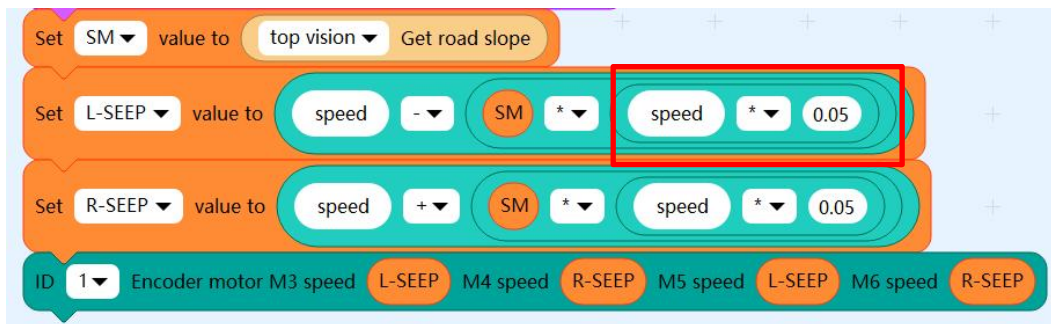
Assuming the line deviates to the left, the vehicle needs to veer to the left to move forward. The left wheel should move slower, and the right wheel faster. Substituting the base speed of 30 and the slope of -8,

Left wheel speed is: $30 - (8 \times 30 \times 0.05) = 30 - (12) = 18$

Right wheel speed is: $30 + (8 \times 30 \times 0.05) = 30 + (12) = 42$

2.3. Autonomous driving - Road slope judgment

How does the slope correspond to the automatic adjustment of the direction of motion? We have summarized an automatically calculated formula.



Think

$$\text{Left wheel} = \text{Base speed} - \text{Road slope} * K_p$$

$$\text{Right wheel} = \text{Base speed} + \text{Road slope} * K_p$$

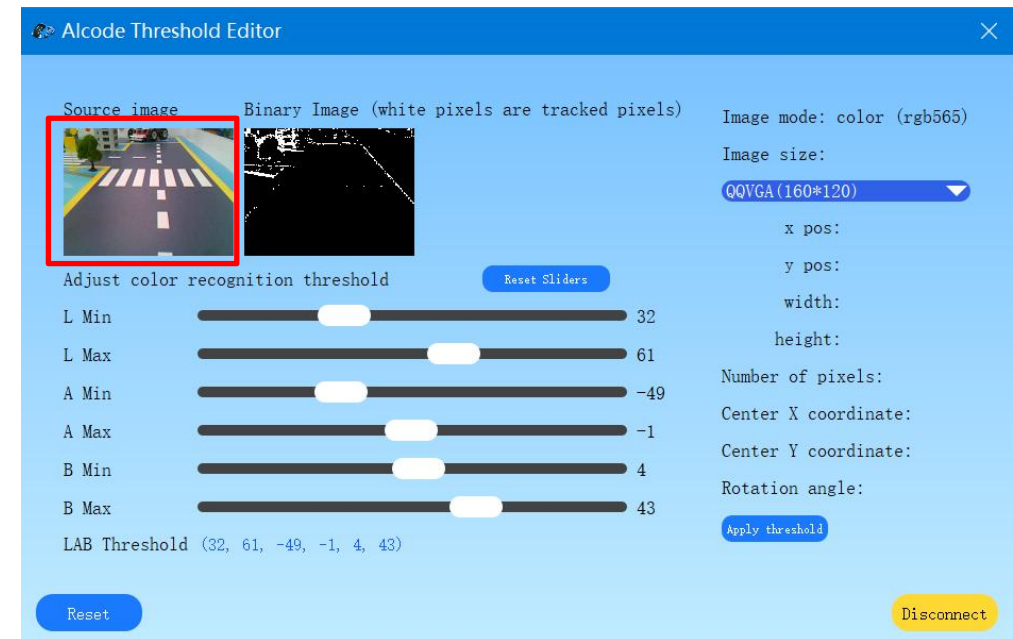
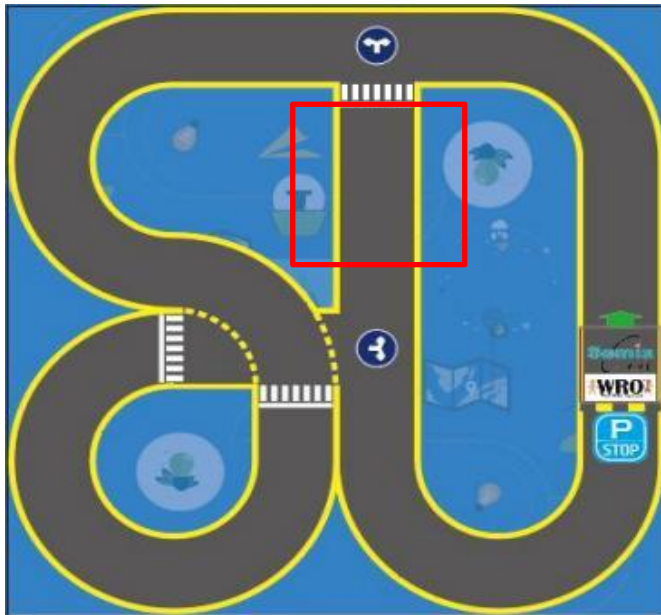
This K_p , also known as the speed adjustment coefficient, can be set according to the base speed. For example, if the base speed is 30, it's generally set to 0.05; if it's 60, it can be set to around 0.1. The setting depends on the actual effect, mainly because the sign of the speed indicates the direction of movement. The road slope multiplied by K_p cannot exceed the base speed.

2.3. Autonomous driving - Road slope judgment

Autonomous driving is commonly used for road alignment. Typical road alignments are double parallel lines, as shown below:

1. Securely mount the camera to the robot, and connect the PH cable to the main controller.
2. Place the robot in a fixed position (if it's programmed, it needs to automatically move to this position to prevent subsequent position changes from altering the viewpoint).
3. Open the AICODE programming software on the computer.
4. Do not power on the robot; keep it powered off. Connect one end of the USB cable to the back of the camera and the other end to the computer.
5. Open the image threshold editor in the AICODE menu, click "Connect," and you will see the real-time image. Adjust the camera to confirm that a valid image is visible.

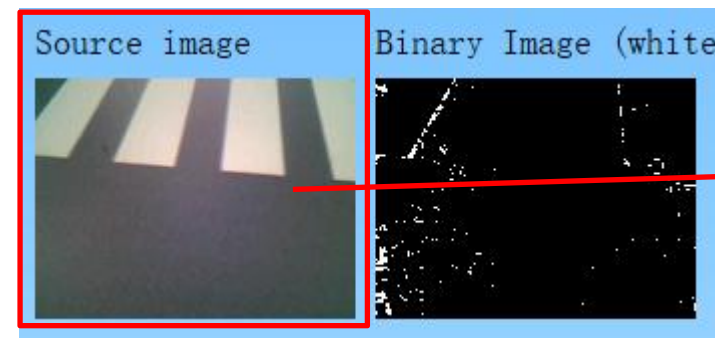
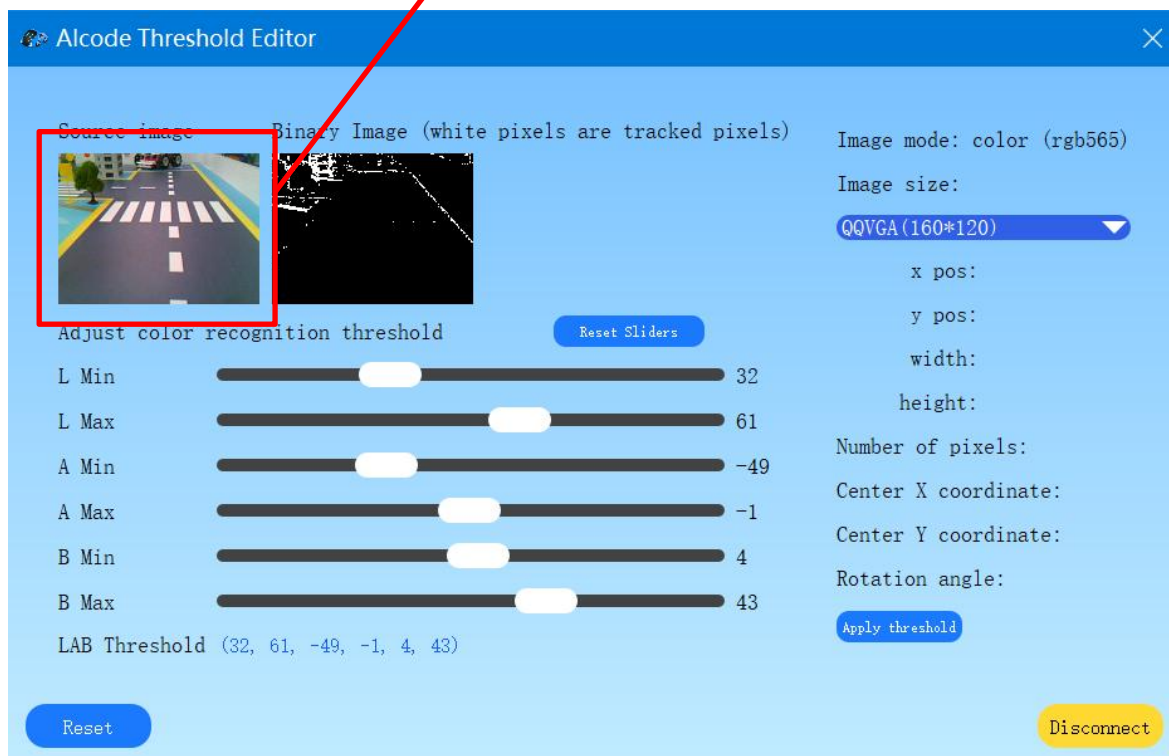
When the car is driving on it, the camera's perspective is as shown in the right image. Because the camera is looking down at an angle, what you see in the image changes slightly.



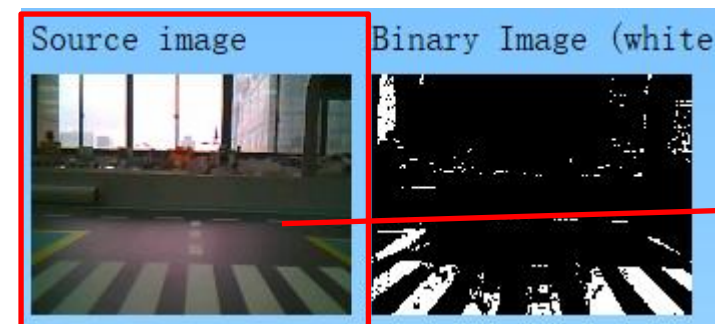
2.3. Autonomous driving - Road slope judgment

6. Debug, collect, and record the required data as needed.

This is a valid shot; the diagonal lines on both sides of the road are clearly visible, and the angle is suitable.



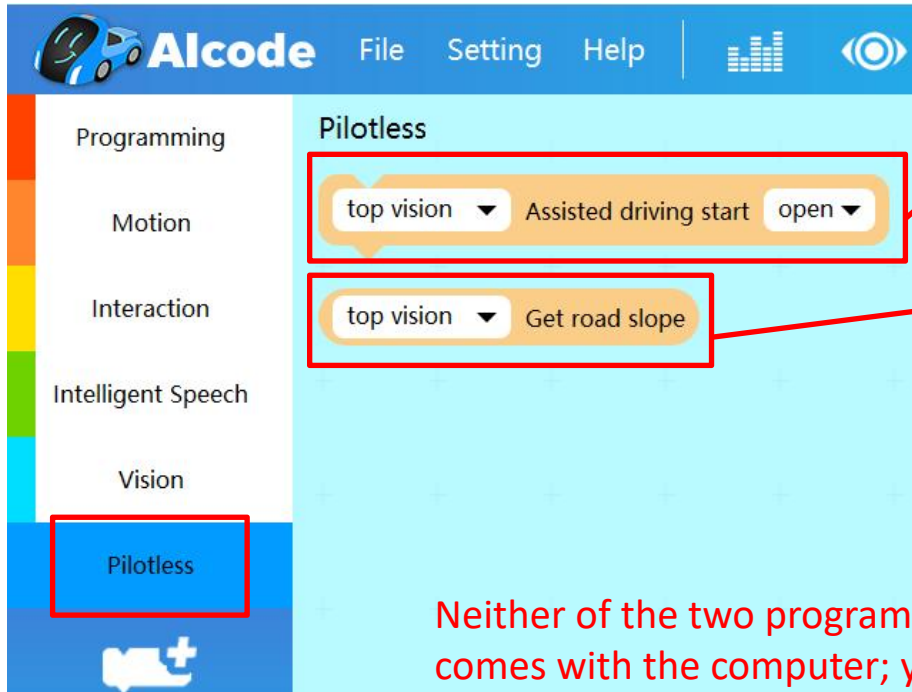
An invalid viewpoint, lacking information, usually indicates the angle is too low.



An invalid perspective, where information is too high, can easily interfere with recognition accuracy. This usually occurs when the angle is too high.

2.3. Autonomous driving - Road slope judgment

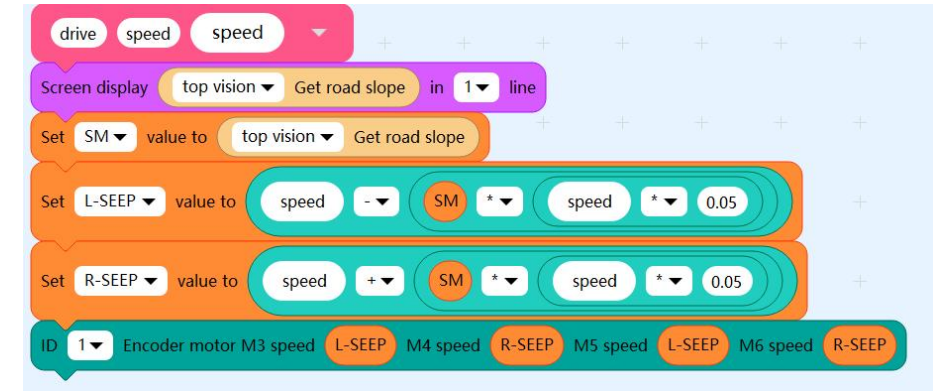
7. Disconnect the USB cable from the back of the camera and connect the cable to the main controller.
8. Power on the main controller.
9. Write and download the program.
10. Test, adjust, and optimize.



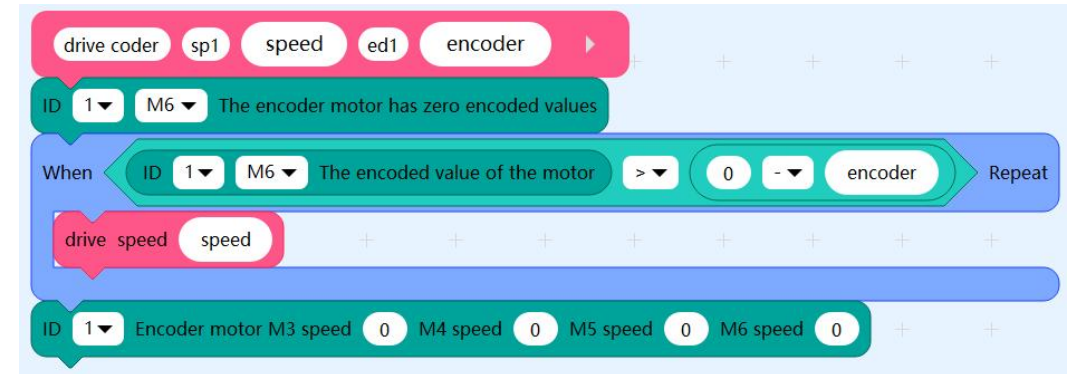
Enable autopilot

Obtain road slope value

Neither of the two programs on the right comes with the computer; you need to create a custom program or open the reference example to use them.



Automatic line-following subroutine (real-time slope acquisition, automatic speed adjustment for following).

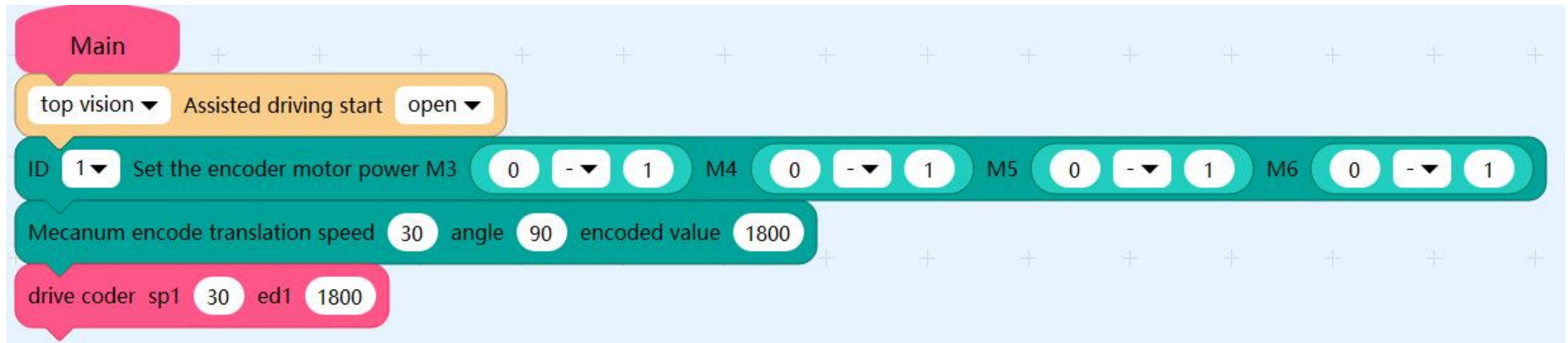


Automatic encoding line-following program (add encoding value, it will follow the line before reaching the encoding value and stop after reaching it, it can also be made into gray-scale trigger, etc.)

2.3. Autonomous driving - Road slope judgment

7. Disconnect the USB cable from the back of the camera and connect the cable to the main controller.
8. Power on the main controller.
9. Write and download the program.
10. Test, adjust, and optimize.

Main program: After leaving the base, visual line following follows the 1800-encoded value.



2.4. Color Recognition

The robot's selection of colors is similar to how humans do classification exercises.

1 Sweet things || Strawberries, ice cream, cake, apples...

2 Search || Does it fall under the category of "sweet things"?

3 Select || Select those that meet the criteria.

4 Change lines || From left to right, from top to bottom

How do you find sweet things?

Line 1

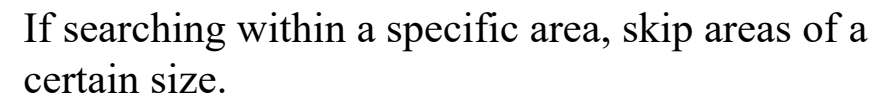
Line 2

Line 3

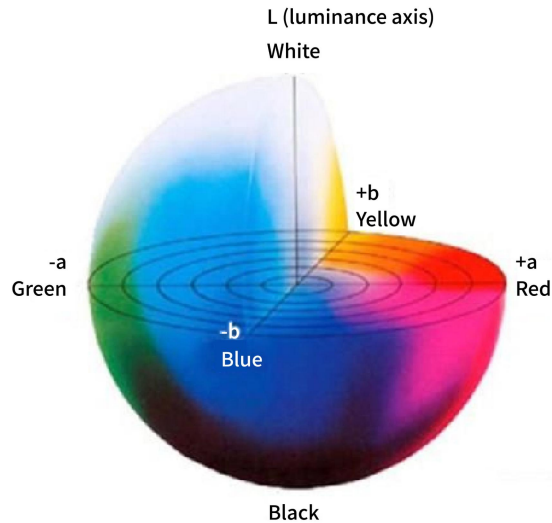
Line 4



NASHENBOT



2.4. Color Recognition



For a program, everything is data. To identify a corresponding color, color data needs to be digitized. Once data that meets the specified standards is found, the color corresponding to that data can be determined.

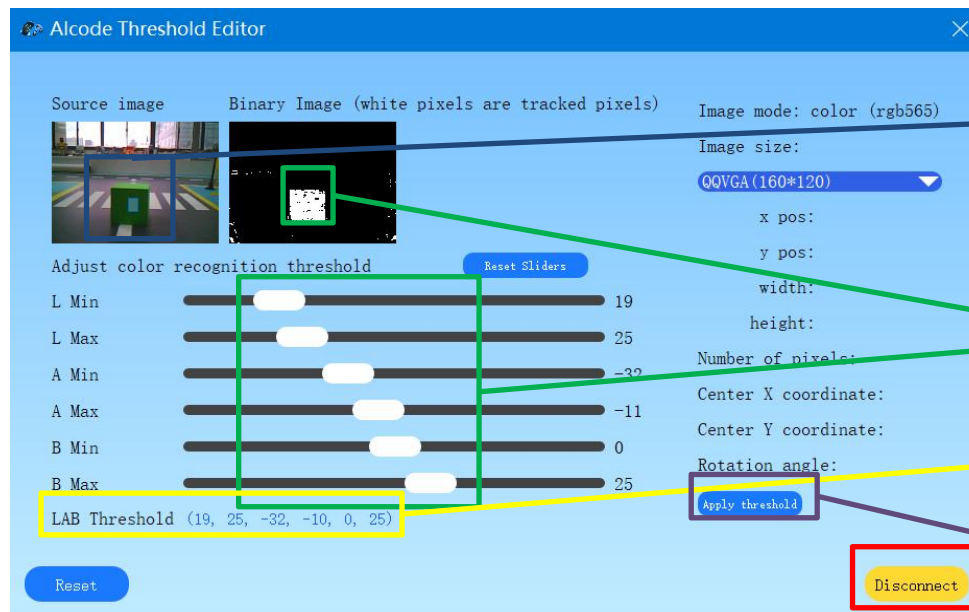
In camera programming, the LAB mode is used. By splitting the target color into three LAB channels, the color information is digitized. Color lookup can be achieved simply by finding the three accessible data ranges for the target color.

L Channel	0		100
A Channel	-128		127
B Channel	-128		127

2.4. Color Recognition

To find the green object in the image, you need to find the descriptive data for that object. According to the LAB channel information, the data used to describe "green" is defined by three dimensions. By selecting the target object's color using the software, you can obtain the target color data based on the threshold editor's display.

1. Securely mount the camera to the machine, and connect the PH cable to the main controller.
2. Place the machine in a fixed position (if it's automatically programmed, it needs to move to this position automatically to prevent changes in viewpoint caused by subsequent position adjustments).
3. Open the AICODE programming software on the computer.
4. Do not power on the robot; keep it powered off. Connect one end of the USB cable to the back of the camera and the other end to the computer.
5. Open the image threshold editor in the AICODE menu, click "Connect," and you will see the real-time image. Adjust the camera to confirm that a valid image is visible.
6. Debug and collect data as needed, and record the required data.



Step 1: Connect the camera

Step 2: Locate the color block you want to identify in the image, and use the mouse to draw a small box to sample the color.

Step 3: Drag these values to make the target color in the image appear white, and the other colors appear black. White indicates valid data.

Step 4: Record these values; you will need to fill them into the programming module later.

Step 5: Apply the data and record it into the camera.

2.4. Color Recognition

7. Disconnect the USB cable from the back of the camera and connect the cable to the main controller.
8. Power on the main controller.
9. Write and download the program.
10. Test, adjust, and optimize.

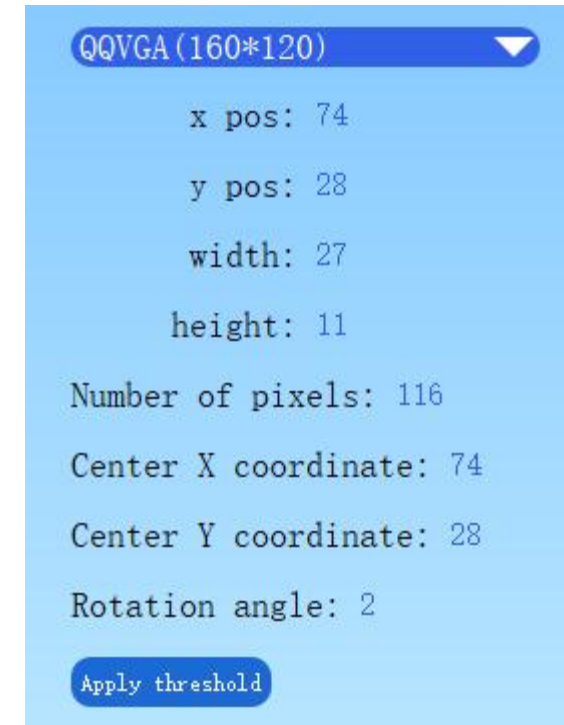


Use when only one color is recognized

When recognizing multiple colors simultaneously

Determine the parameters of the identified color.

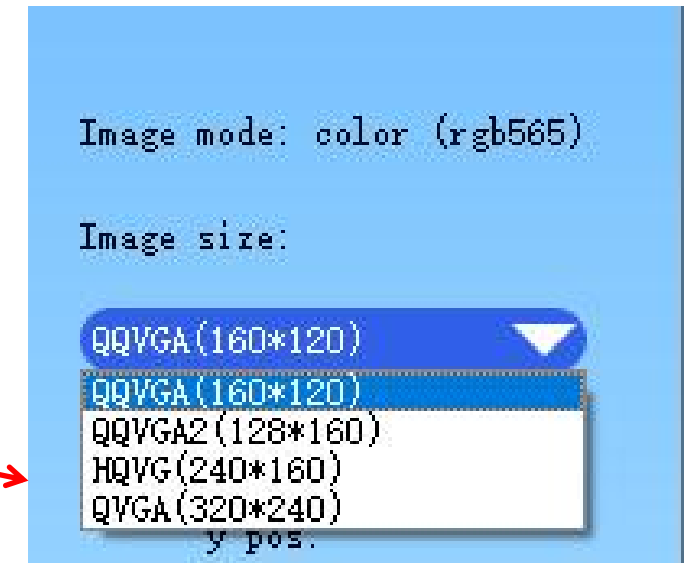
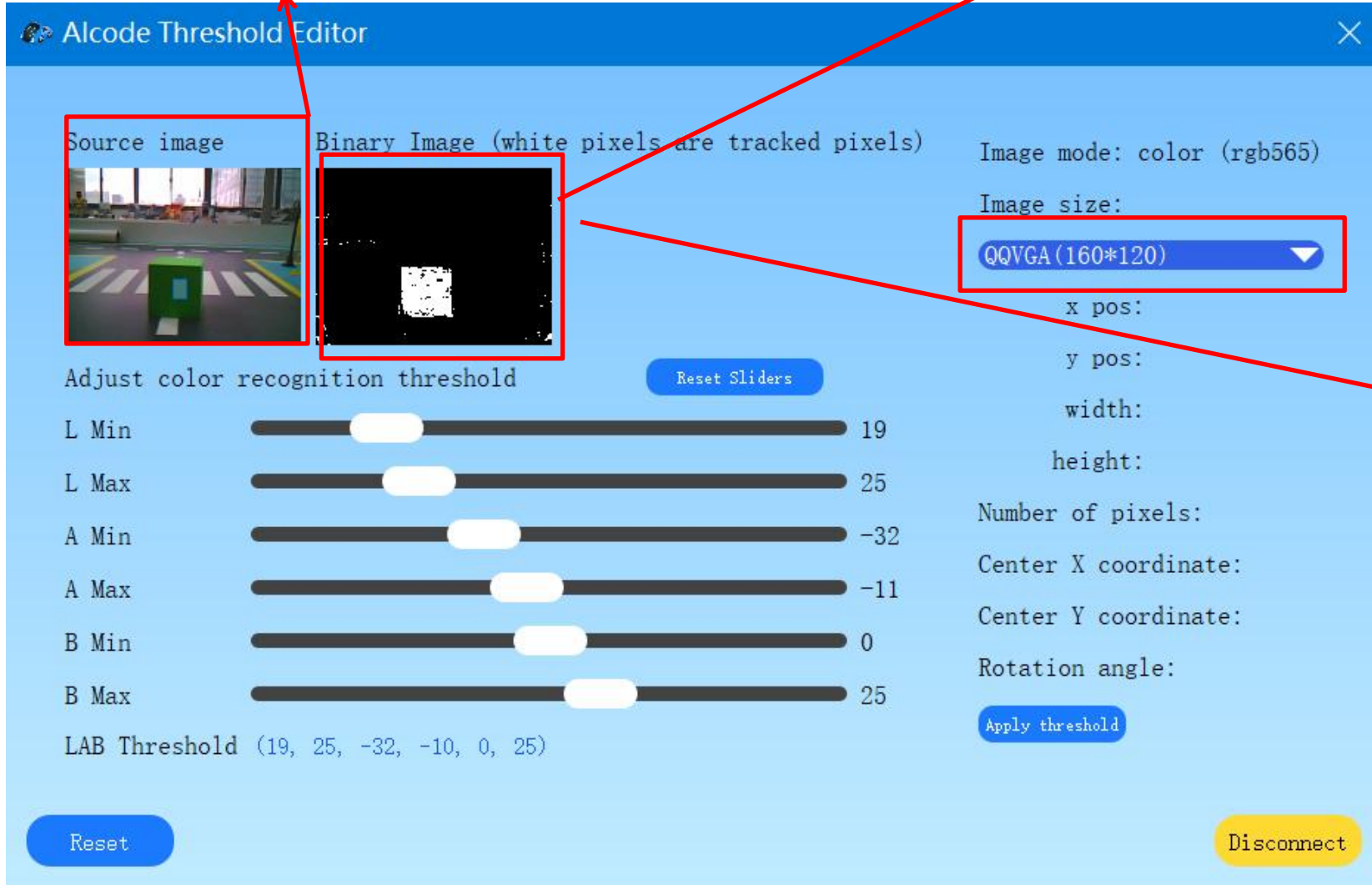
Determine the parameters of the multiple colors being identified.



2.4. Color Recognition

This is what the camera actually saw.

This is what the computer actually sees after filtering out interference.

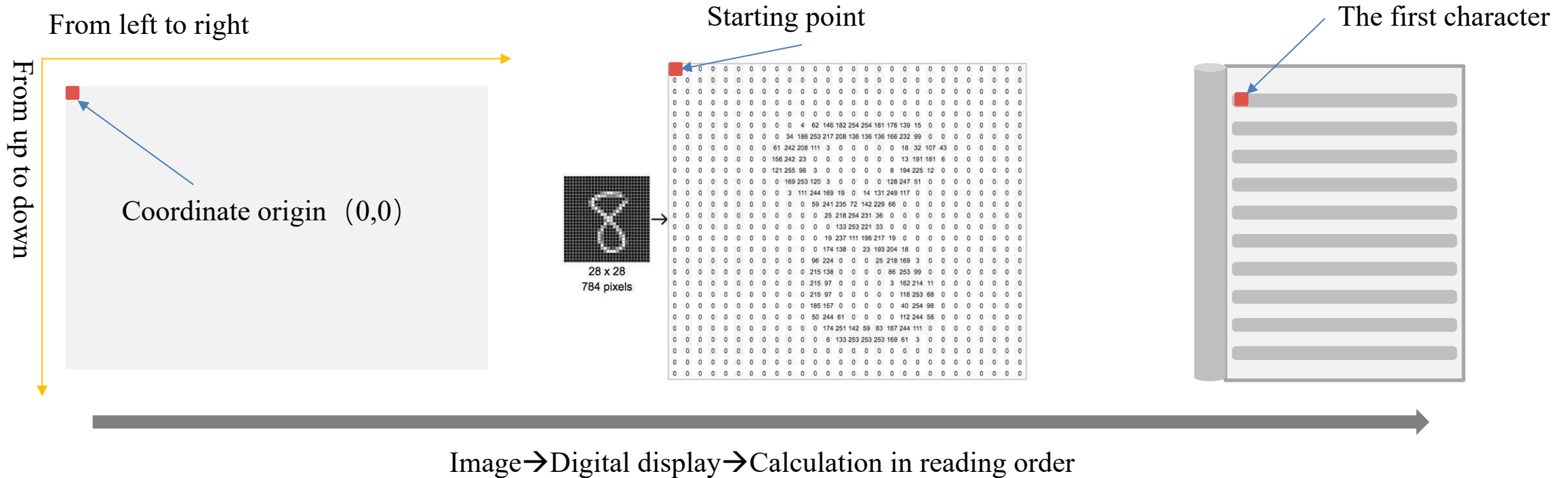


The dimensions of the image displayed on the computer can be set. Imagine it as a black sheet of paper, divided into many different small squares. Different things can be drawn on each square. The white dots represent what you see. Once you see something, you paint the squares on the black paper with white paint.

2.4. Color Recognition

Drawing on this paper requires following the position of the drawing.

Image coordinate system

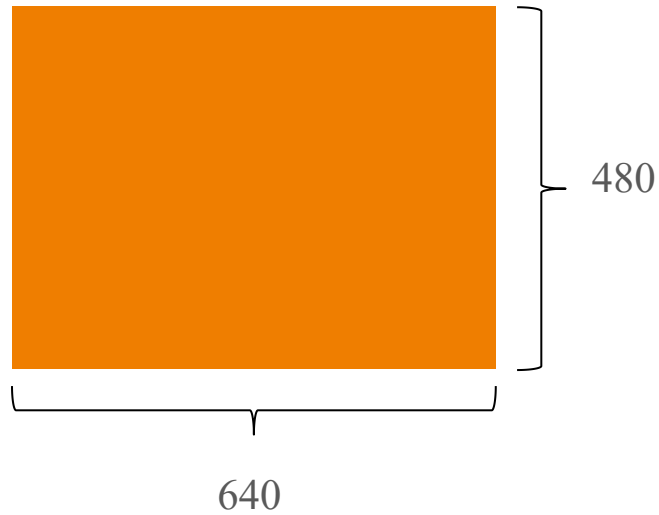


The image is calculated from left to right and from top to bottom. The entire calculation process is consistent with the order in which people read. Therefore, when marking the coordinates of the target object, the image origin is in the upper left corner.

2.4. Color Recognition

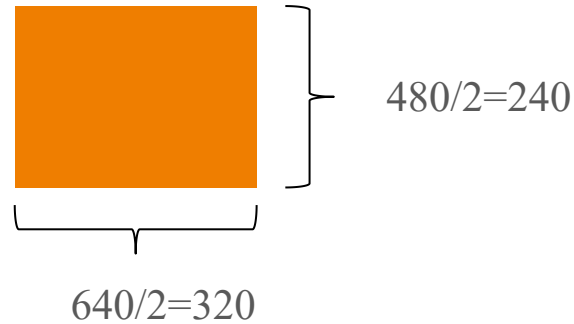
Windows Size

VGA



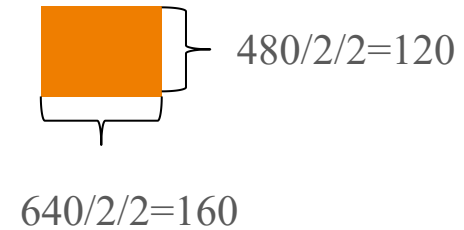
QVGA

QVGA: Q=quarter
The size is one quarter of the original



QQVGA

QVGA: The size is one sixteenth of the original

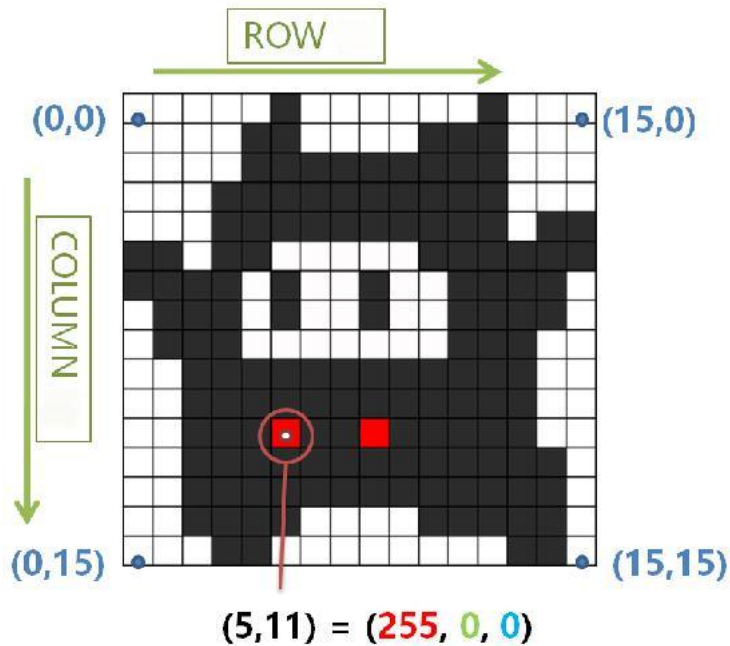


Common preset window sizes are as follows.

Preset Window Size	Window Width	Window Height
VGA	640	480
QVGA	320	240
QQVGA	160	120

2.4. Color Recognition

Pixel



Each image is made up of a matrix of pixels.

The coordinates start from the upper left corner, the horizontal ones are rows, marked as x, and the vertical ones are columns, marked as y.

If we want to access the pixel marked by the red circle in the figure below, how can we find it?

Counting from left to right, it is in the column with serial number 5.

Counting from top to bottom, it is in the row with serial number 11.

So the coordinates of this point are (255, 0, 0)

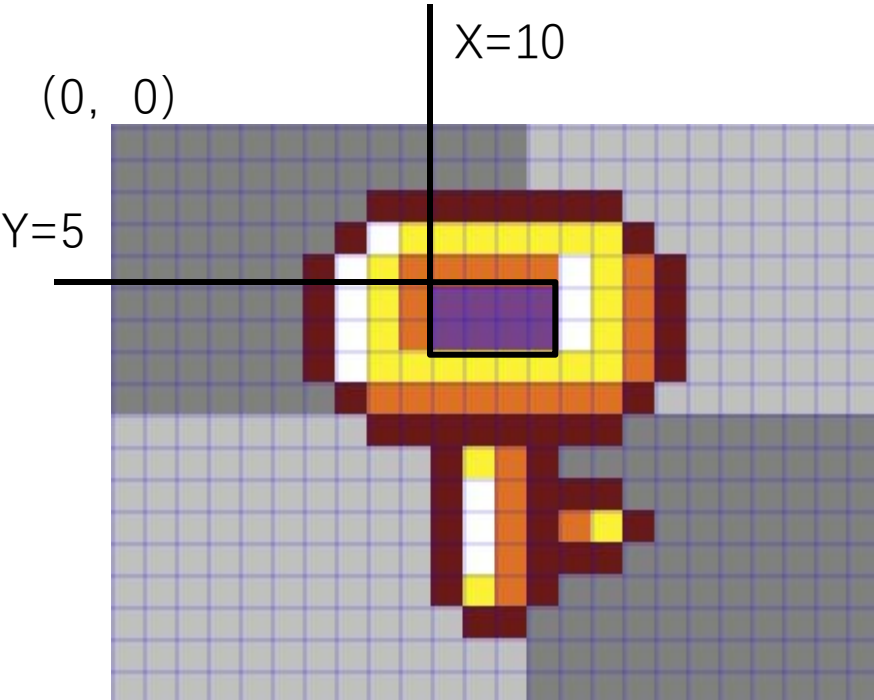
Pixel value

Pixel format varies with the image format

If it is a grayscale image, the returned value is an integer in the range of 0-255

If it is an RGB image, the returned value is a tuple of length 3 (r, g, b), for example (90, 90, 90)

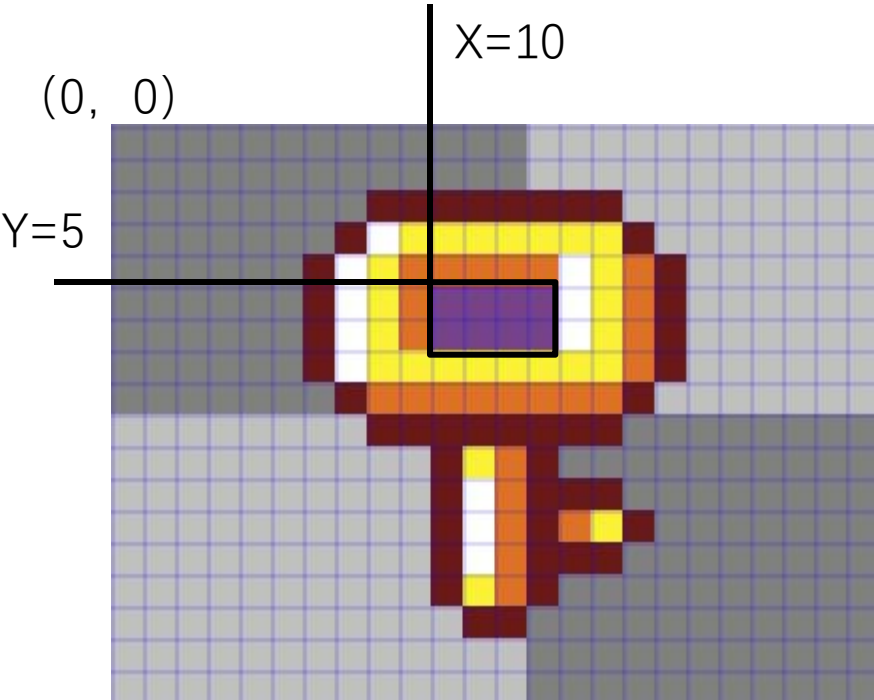
2.4. Color Recognition



Recognized colors

X coordinate	X=10
Y coordinate	Y=5
Width	W=4
Height	H=2
Number of pixels	S=4*2
Center X coordinate	$X_c=10+4/2$
Center Y coordinate	$Y_c=5+2/2$
Rotation Angle	

2.4. Color Recognition

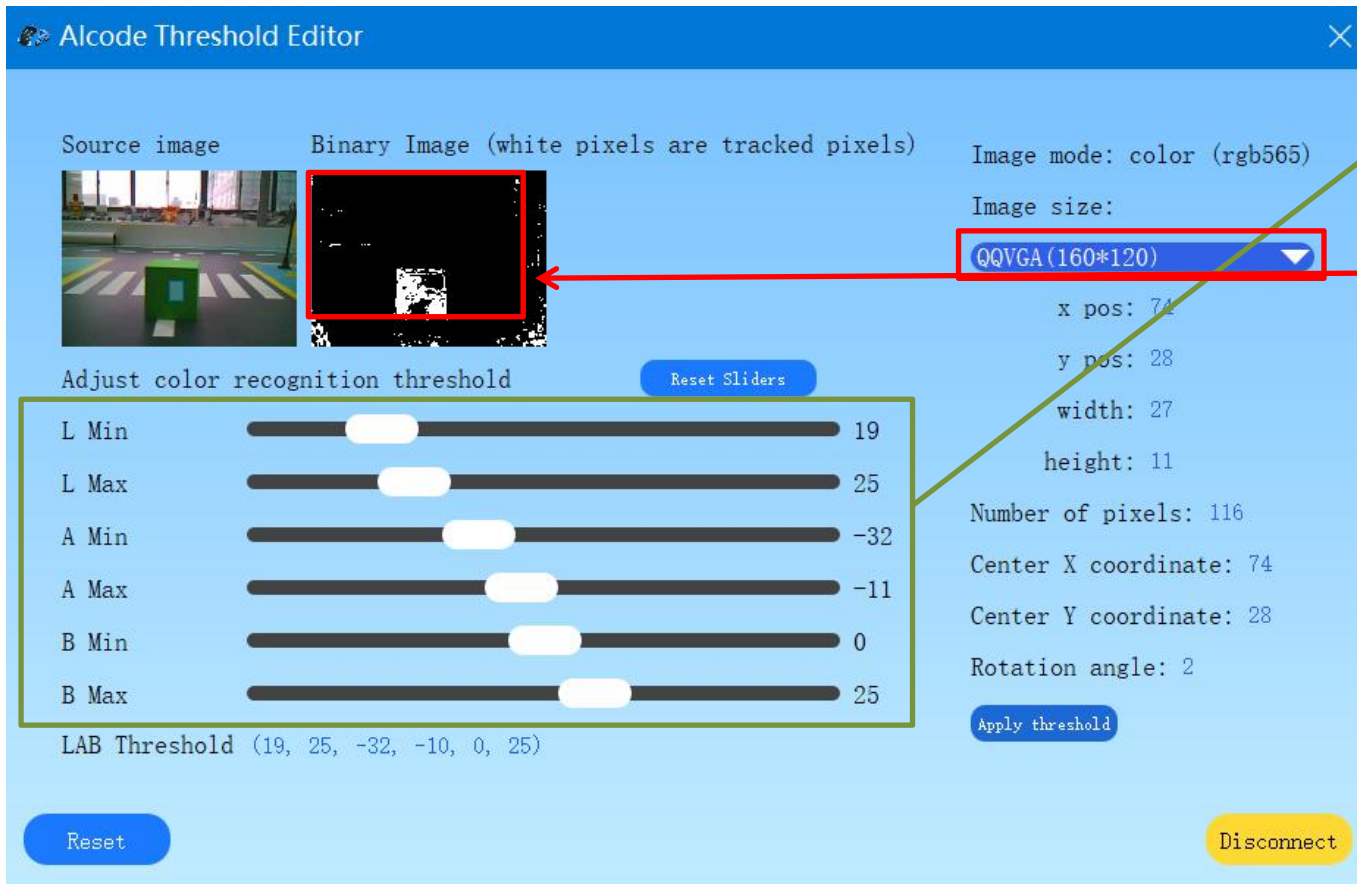


Function of the parameter

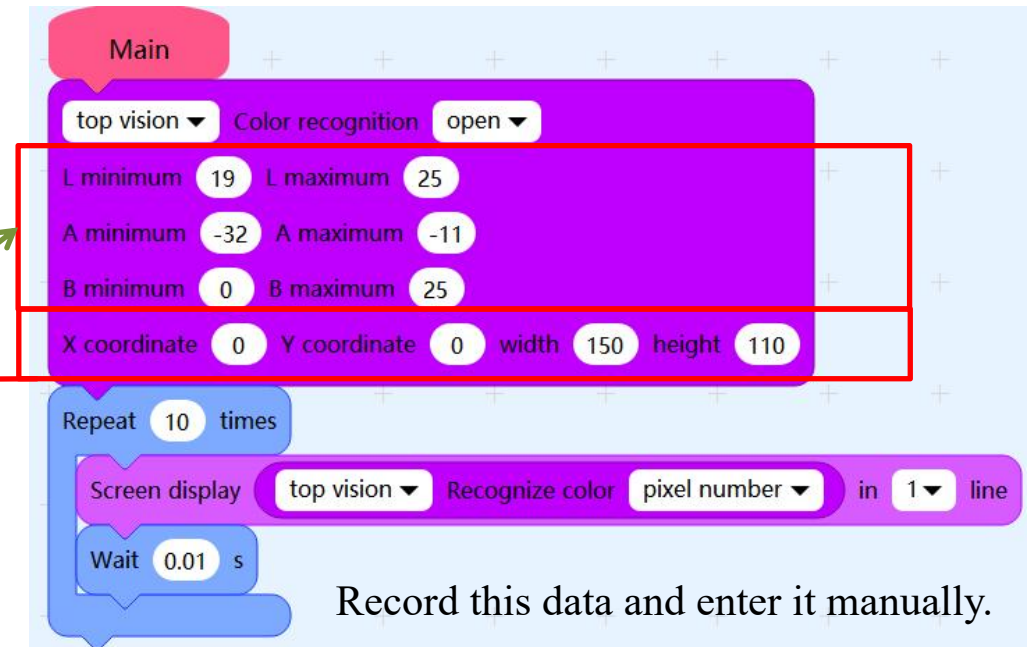
X coordinate	Determine if the left side has left the frame.
Y coordinate	Determine if the above has left the frame.
Width	Judging distance
Height	Judging distance
Number of pixels	Judging distance
Center X coordinate	To determine which side of the screen it is on, follow...
Center Y coordinate	To determine which side of the screen it is on, follow...
Rotation Angle	Whether the judgment is biased or not

2.4. Color Recognition

This indicates that the valid image on the canvas is not 0,0,160,120? There are some distracting points at the bottom of this image; these can be ignored by the machine for greater accuracy.



Common color recognition programs

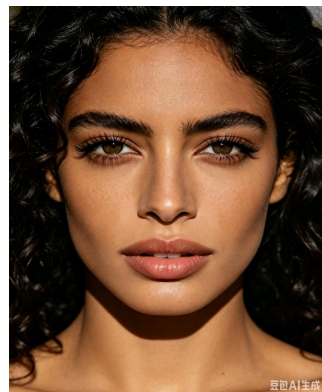


7. Disconnect the USB cable from the back of the camera and connect the cable to the main controller.
8. Power on the main controller.
9. Write and download the program.
10. Test, adjust, and optimize.

2.5. Custom template matching and template matching

When we have complex images to recognize, we need to use machine learning to teach the camera image features, then memorize those features so that it can recognize similar images when they appear.

For example, enabling the camera to recognize different faces and different objects.

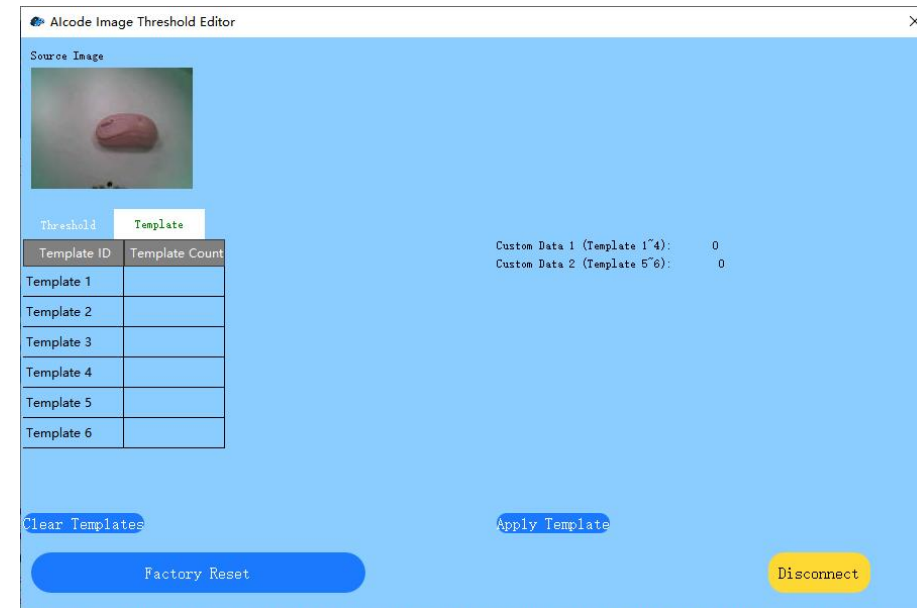


Learn

Recognize

2.5. Custom template matching and template matching

To use this feature, you need to update the camera files first.



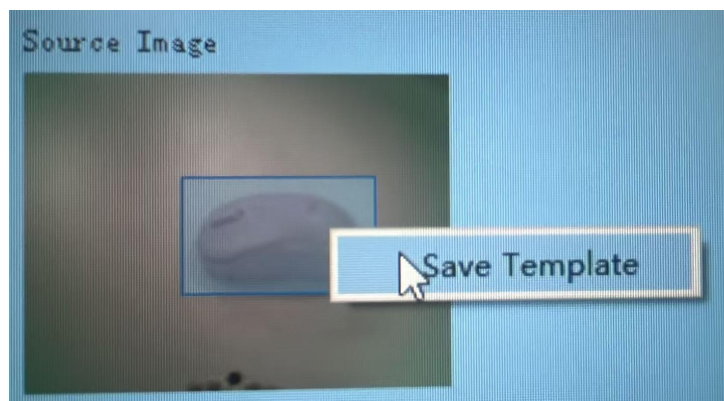
1. Securely mount the camera to the machine, and connect the PH cable to the main controller.
2. Place the machine in a fixed position (if it's automatically programmed, it needs to move to this position automatically to prevent changes in viewpoint caused by subsequent position adjustments).
3. Open the AICODE programming software on the computer.
4. Do not power on the robot; keep it powered off. Connect one end of the USB cable to the back of the camera and the other end to the computer.
5. Open the image threshold editor in the AICODE menu, click "Connect," and you will see the real-time image. Adjust the camera to confirm that a valid image is visible.
6. Debug and collect data as needed, and record the required data.

2.5. Custom template matching and template matching

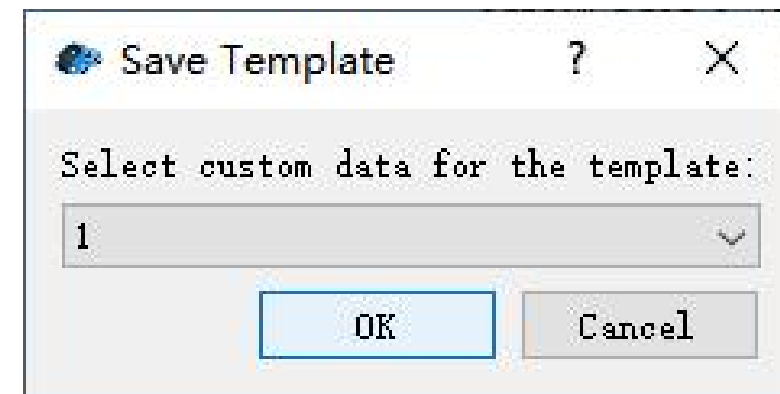
Re-open the image threshold editor, and a template function will now be available.
Point the camera at the pattern you want to recognize and select it with a bounding box.
Right-click and save it.
Select a template number.
Confirm.



Step 1: Select the target object by drawing a box around it.



Step 2: Right-click and save as.

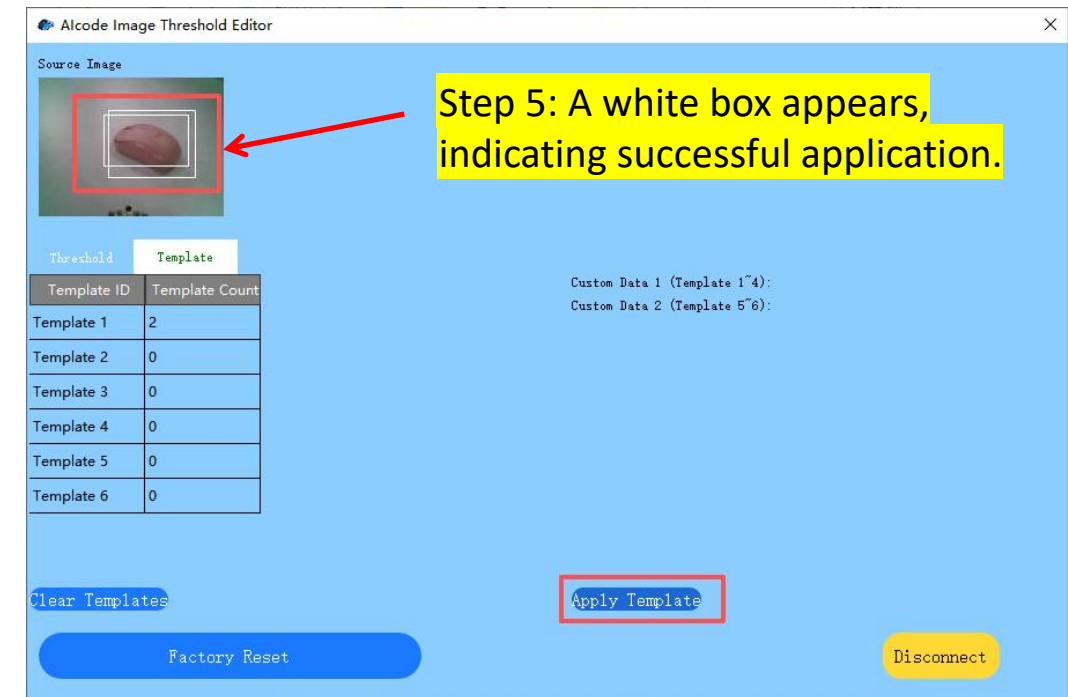
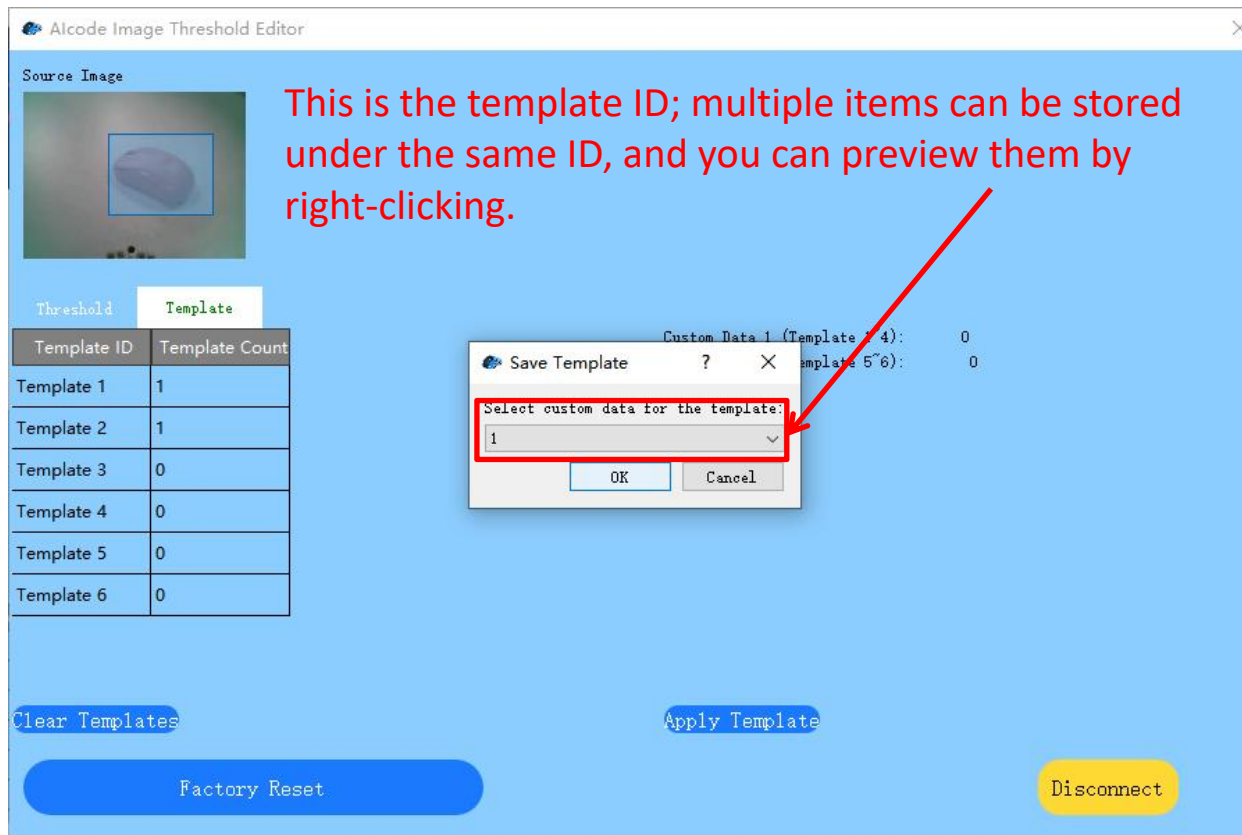


Step 3: Select a template to save to. You can save up to six templates in total, and each template can store multiple images.

2.5. Custom template matching and template matching

Test Results

If recognition is inaccurate, save a few more images from different angles.
Then apply.

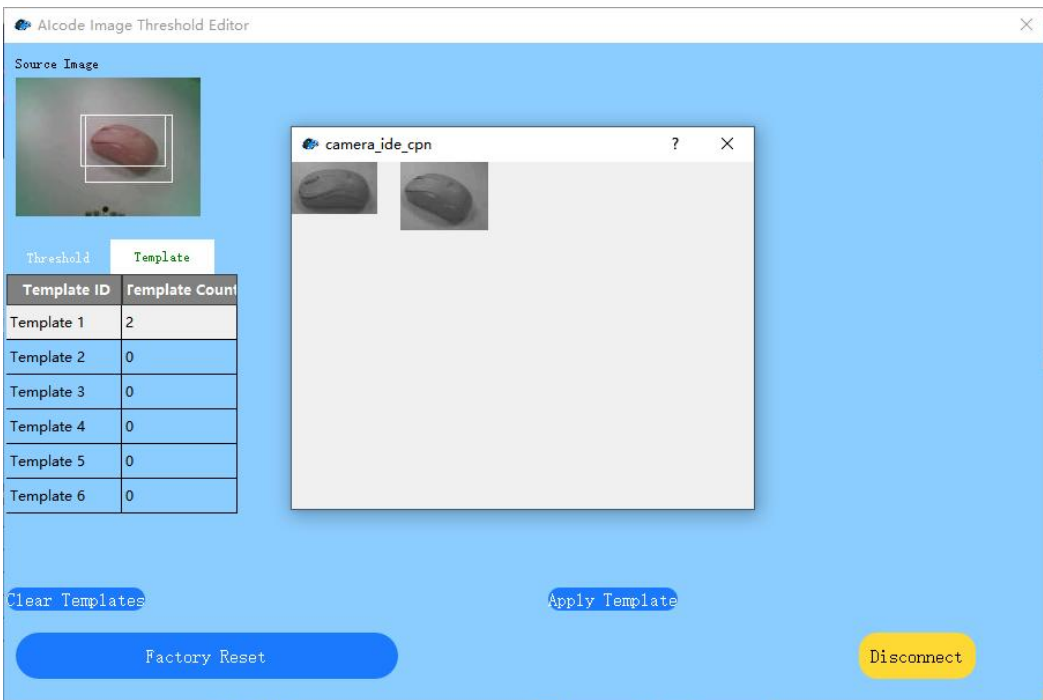
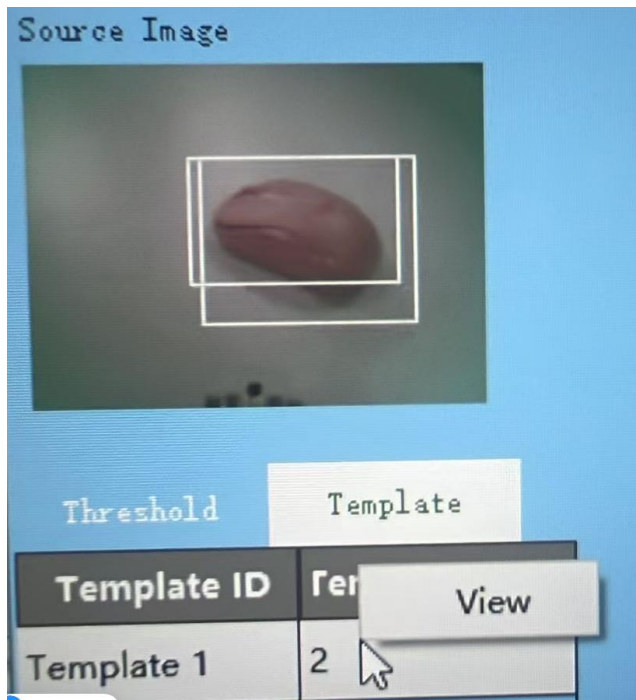


Step 4: Click "Apply," and the data will be saved to the camera.

Step 5: A white box appears, indicating successful application.

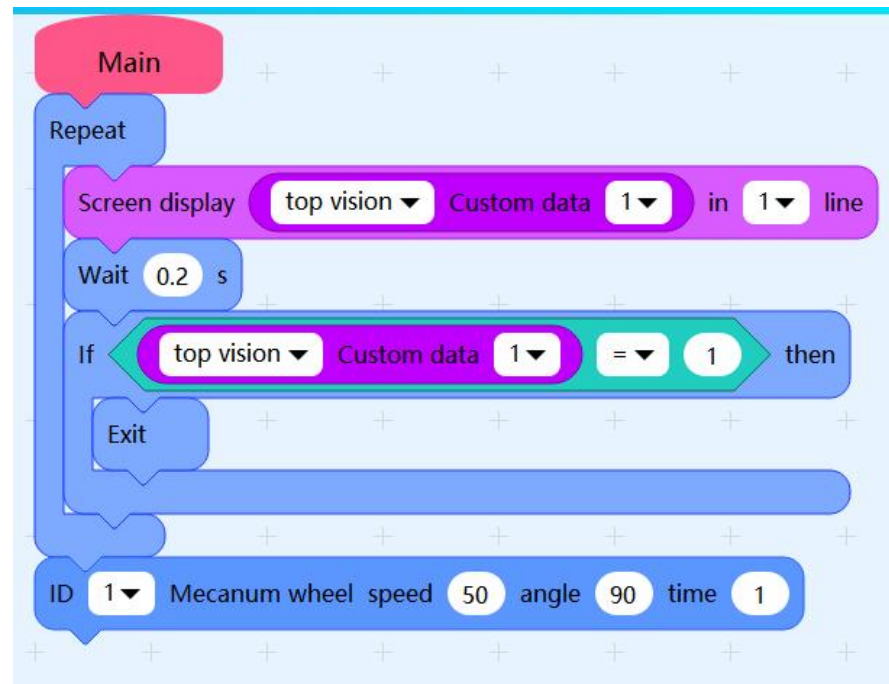
2.5. Custom template matching and template matching

If you do not recall previous image data, you can right-click to preview it; you can also save multiple sets of data using the same procedure described above.



2.5. Custom template matching and template matching

7. Disconnect the USB cable from the back of the camera and connect the cable to the main controller.
8. Power on the main controller.
9. Write and download the program.
10. Test, adjust, and optimize.



The 5th Generation Educational Robot

— Artificial Intelligence Programming Robot

Thanks !