

3. QR code instructions

3. QR code instructions

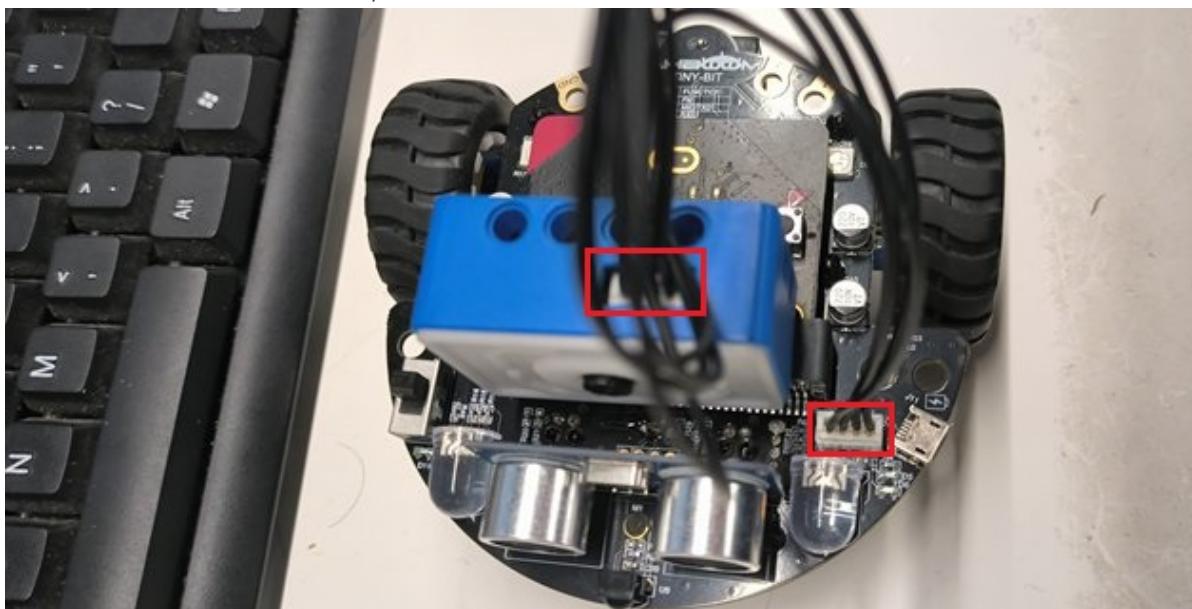
- 3.1 Experimental preparation
- 3.2 Experimental installation
- 3.3 Import and briefly explain the microbit building blocks
 - 3.3.1 Open the programming website
 - 3.3.2 Create a new project
 - 3.3.3 Add the camera building block
 - 3.3.4 Add the car building blocks
 - 3.3.5 Introduction to the main building blocks
- 3.4 Experimental results
- 3.5 Extension: STA mode connection

3.1 Experimental preparation

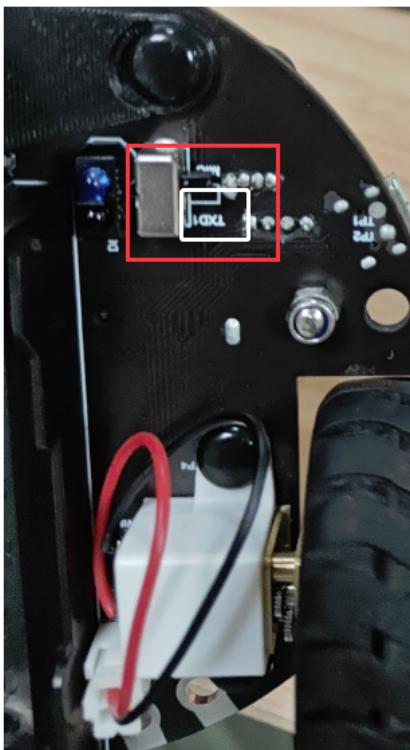
- tinybit car
- microbit
- wifi camera

3.2 Experimental installation

Connect the camera to the car, as shown below



Note: The switch at the bottom of the car must be turned to TXD to work properly with the image transmission camera

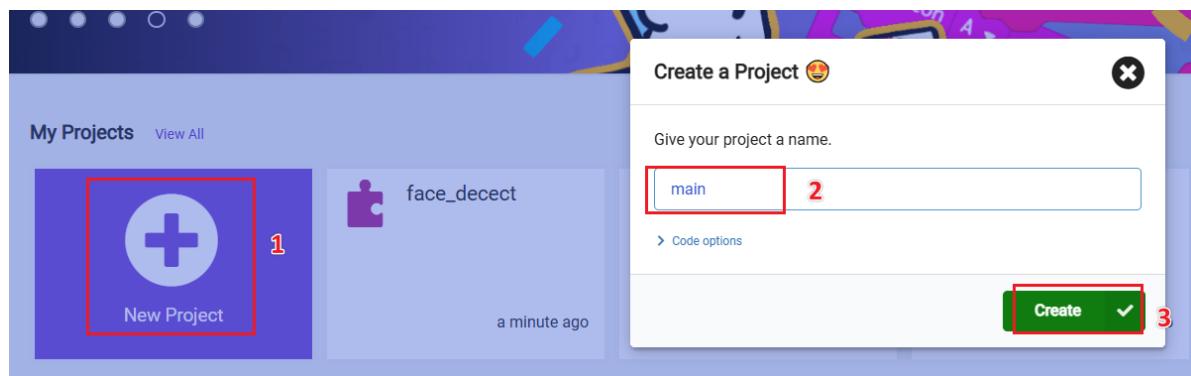


3.3 Import and briefly explain the microbit building blocks

3.3.1 Open the programming website

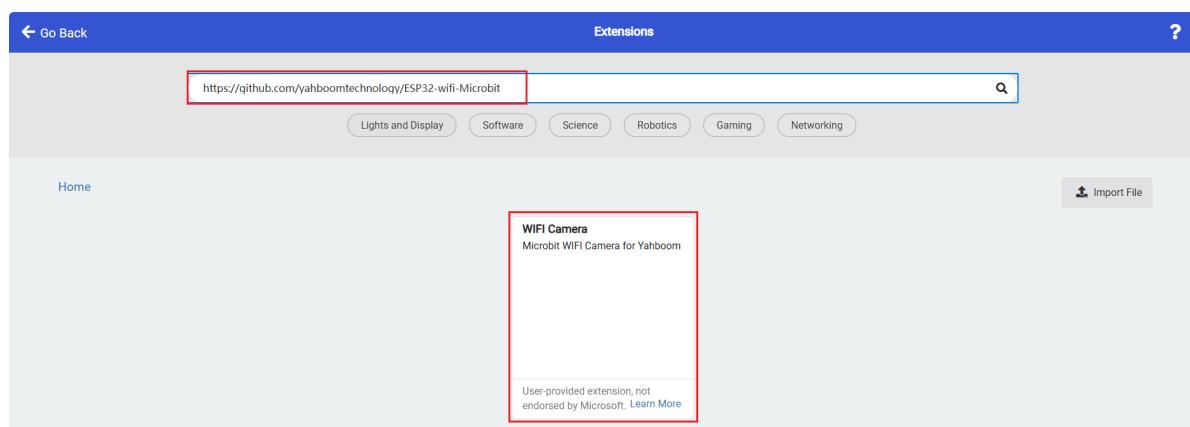
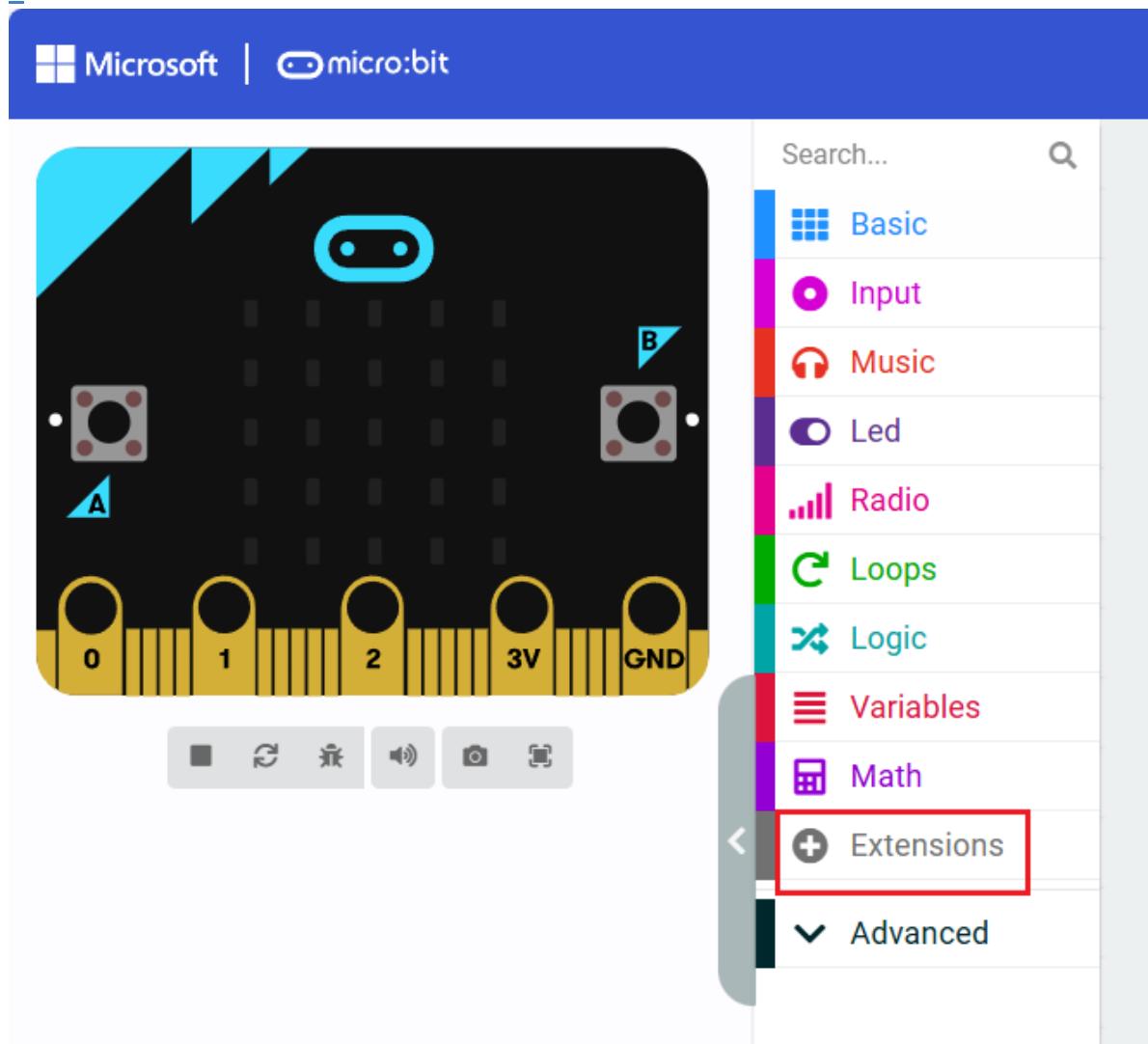
<https://makecode.microbit.org/#>

3.3.2 Create a new project



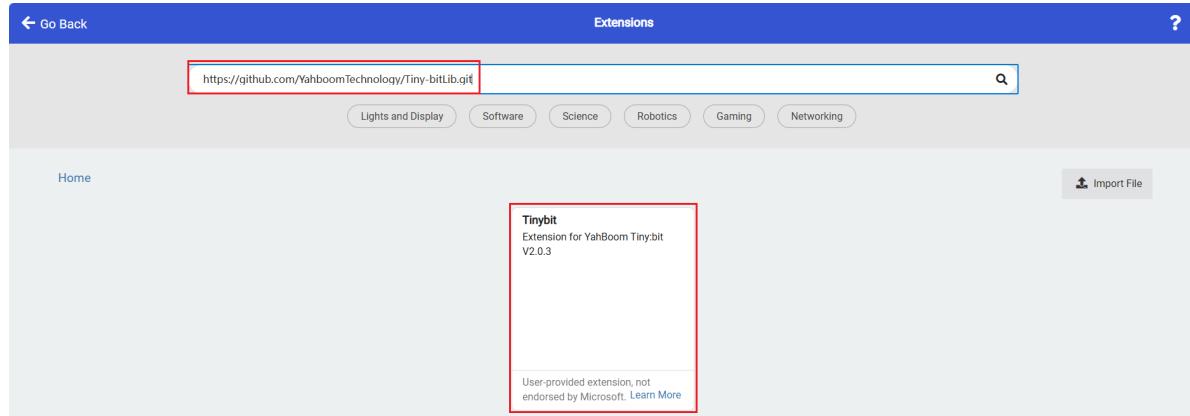
3.3.3 Add the camera building block

Website for the camera building block: <https://github.com/yahboomtechnology/ESP32-wifi-Microbit>



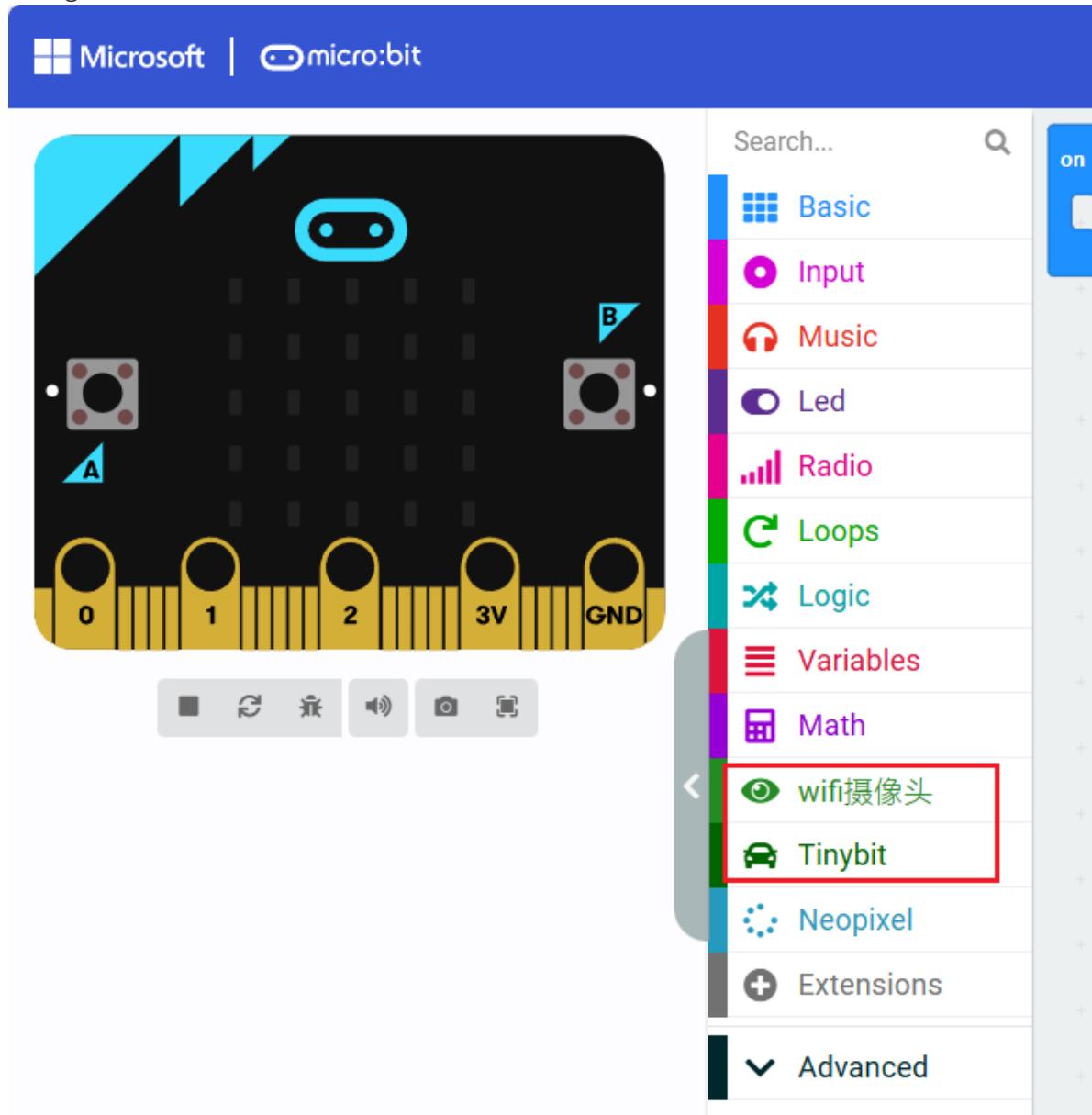
3.3.4 Add the car building blocks

The URL of the car building blocks: <https://github.com/YahboomTechnology/Tiny-bitLib.git>



3.3.5 Introduction to the main building blocks

After successfully introducing the above building blocks, the result will be displayed as shown in the figure below



- **Serial port initialization building block** This is used to define the pin for serial communication. When communicating with the wifi camera, the baud rate is 115200 by

default and cannot be changed

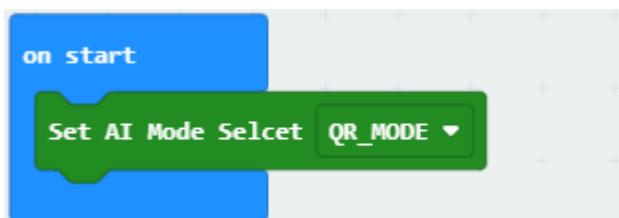


- **Set the size of the serial port receiving buffer** This block is used to define the size of a packet of data that can be accepted for transparent transmission, for example



This defines that the maximum size of a packet is 30 characters. Exceeding this will result in incomplete data reception. **This value cannot be less than 25, otherwise the IP information will also be incomplete**

- **Set AI mode** This building block is used to set the current AI visual gameplay mode, which can be set to different modes



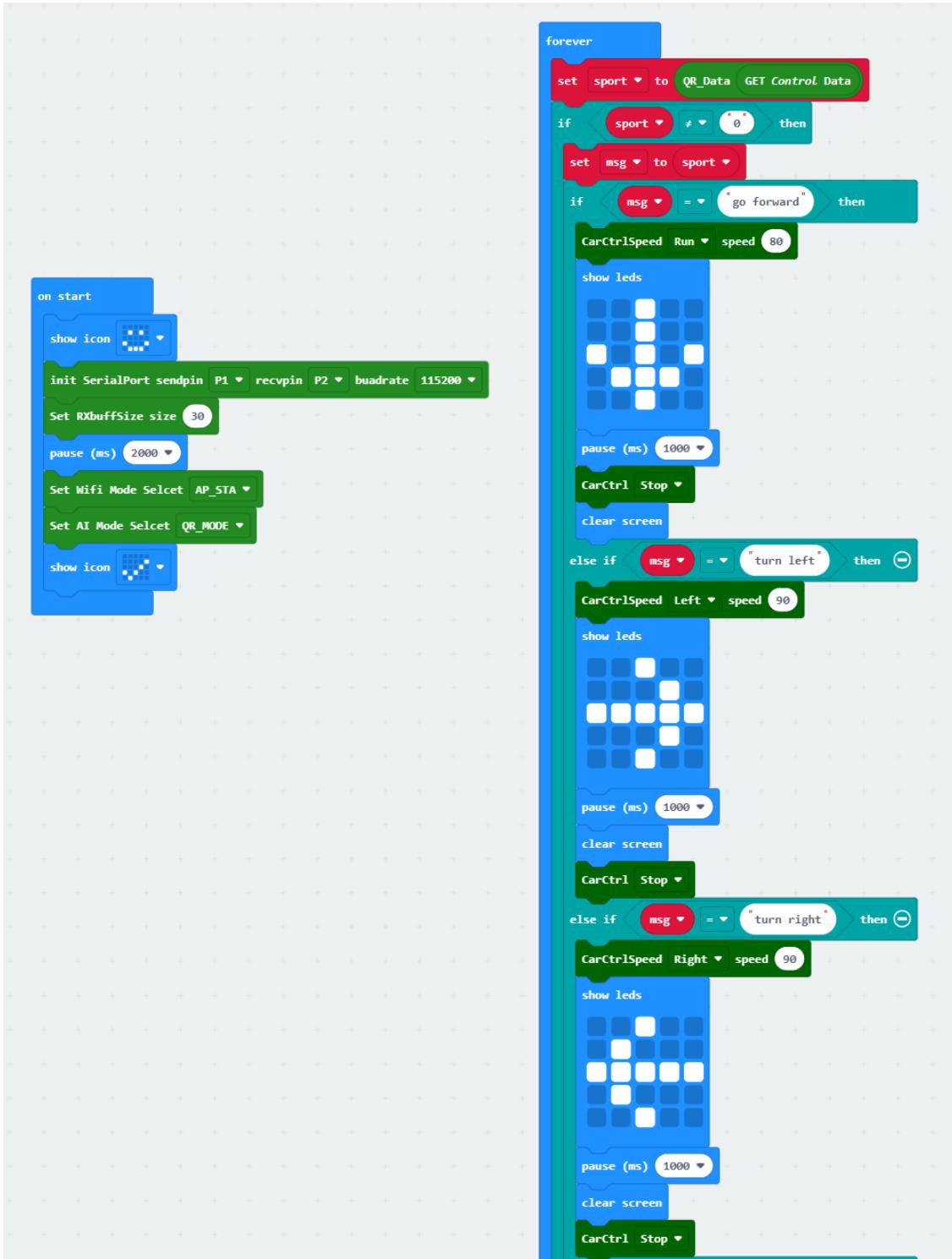
- **Building block for obtaining transparent data** This building block is mainly used to obtain information sent by the host computer to microbit, and transmits information through the wifi camera as an intermediate



- **Method to open the source code provided by the tutorial**

1. Open the URL <https://makecode.microbit.org/#> in the browser
2. Then drag the microbit-QR_Sport.hex file provided in this experiment into the browser that opens the URL, and the program diagram of the source code of this project will be

automatically opened

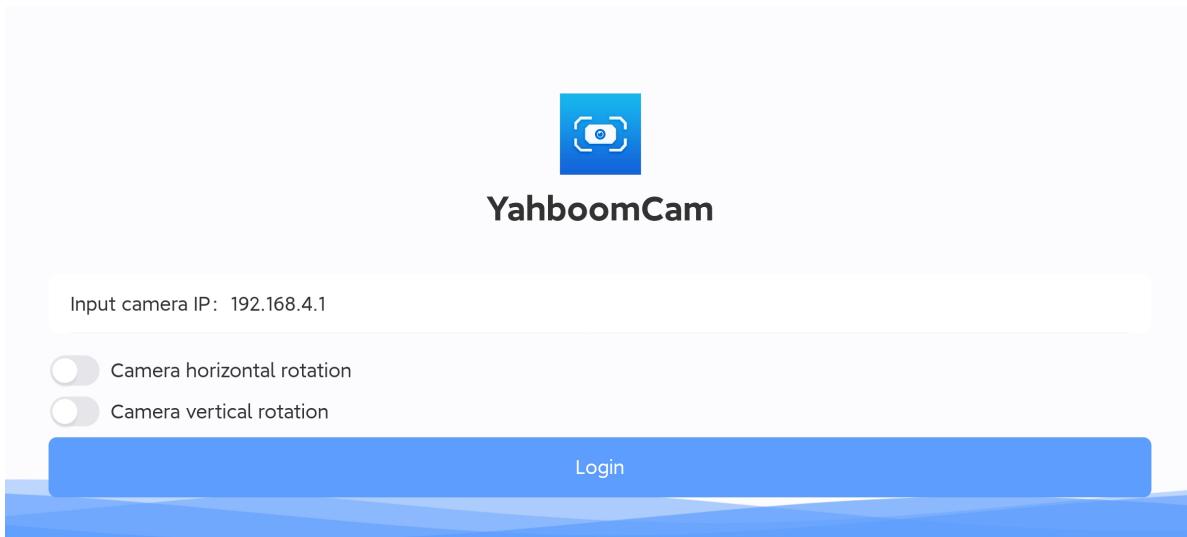


3.4 Experimental results

1. Turn on the switch of the car, wait for the pattern of the corresponding mode setting to be displayed on the microbit, and then you can connect the APP. After successful startup, the camera module will generate a hotspot. The default IP address of the hotspot AP mode is 192.168.4.1 (this tutorial recommends using a simple self-heating hotspot AP mode for connection. To connect to WIFI using STA mode, please see the extended content of STA mode connection below)
2. Download the APP: For Android, please use the browser to scan the following QR code to download and install YahboomCam. For Apple, please use the camera to scan the QR code to enter the App Store to download and install, or search for "YahboomCam" in the App Store to find "YahboomCam".



3. Use your mobile phone to connect to the hotspot (the default hotspot name is Yahboom_ESP32_WIFI, without a password). Open YahboomCam, enter the camera IP and fill in the default IP address 192.168.4.1 displayed by microbit, and then click Login.



4. The page after login is as follows, and you can see the camera screen.



Aim the camera at the QR code to be recognized. After recognition, the car will perform corresponding actions according to the instructions, and the instruction icon will be displayed on the microbit mainboard at the same time.

When the forward instruction is recognized, the forward arrow will be displayed on the microbit mainboard, and the car will move forward for one second at the same time; when the backward instruction is recognized, the backward arrow will be displayed on the microbit mainboard, and the car will move back for one second at the same time; when the left instruction is recognized, the left arrow will be displayed on the microbit mainboard, and the car will move left for one

second at the same time; when the right instruction is recognized, the right arrow will be displayed on the microbit mainboard, and the car will move right for one second at the same time.

Note: You need to open the screen to start recognition. If the screen is closed in the middle, the recognition will also be closed.

QR code image:



go forward



go back



turn left

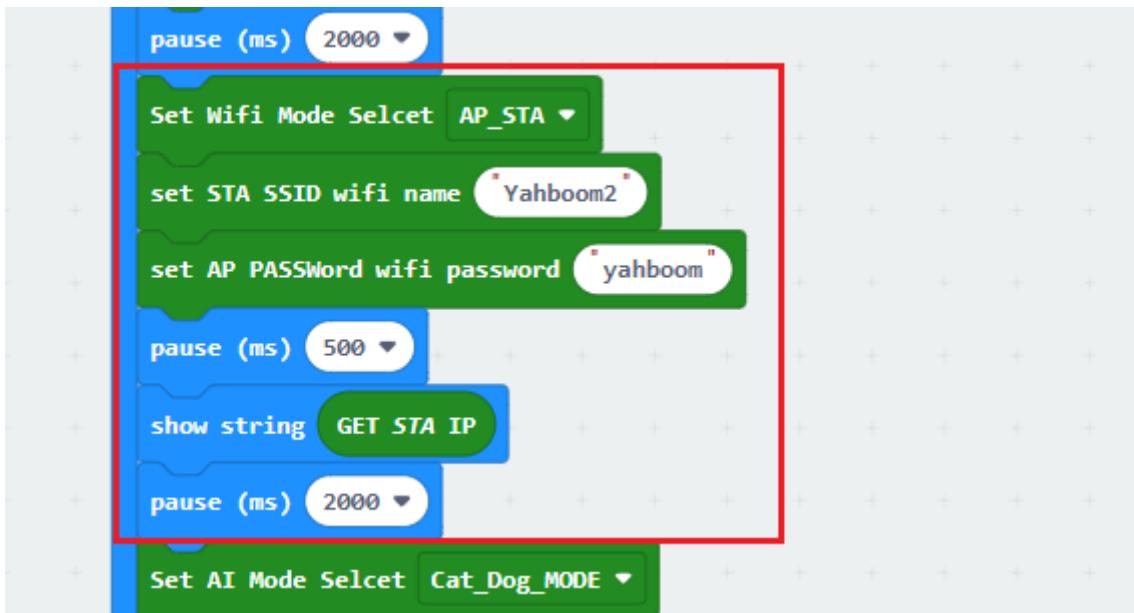


turn right

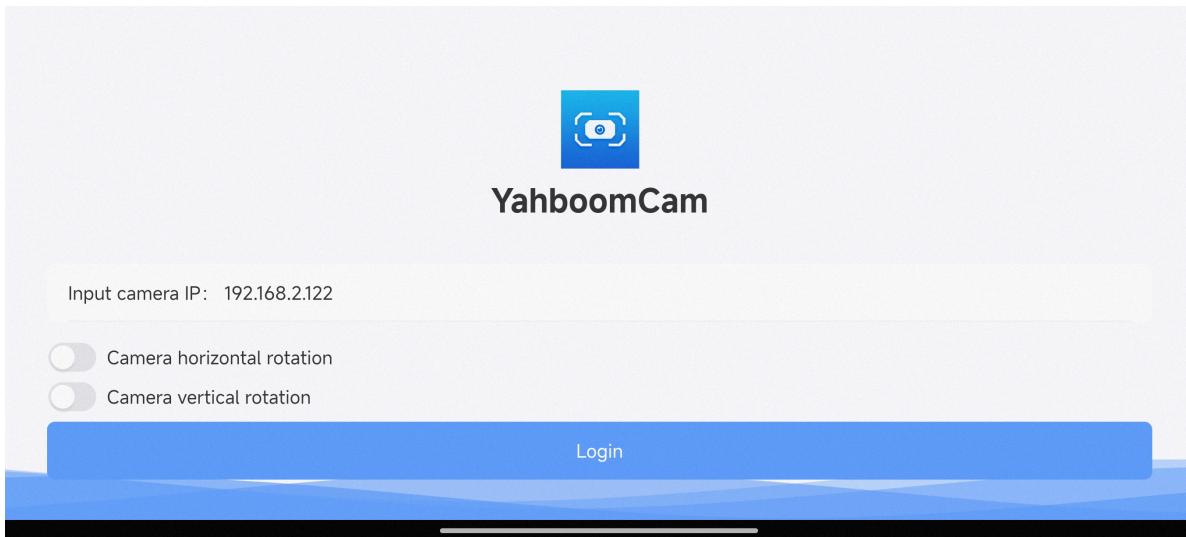
3.5 Extension: STA mode connection

1. Change the source code, drag the .hex file provided in this experiment into the browser that opens the URL, set the WiFi name and password to connect, and the WiFi name and password must be filled in correctly according to the WiFi you want to connect to, otherwise there is no way to correctly connect to the WiFi and display the IP address. If the connection is successful, the IP address of sta_ip:192.168.x.x will be displayed. If the connection is

unsuccessful, sta_ip:null will be displayed. You need to check whether the WiFi name and address filled in the program are correct.



2. Connect your phone to the WIFI set above, and make sure that the phone and the robot are connected to the same WiFi. Open YahboomCam, enter the camera IP and fill in the sta_ip address displayed by microbit (pay attention to the content displayed by microbit, and after seeing 1, 9, and 2 floating continuously, the IP address is being displayed), and then click Login. (The IP in the picture is only an example, the specific IP should be based on the microbit display)



3. After logging in, you can see the following page and the camera screen.

