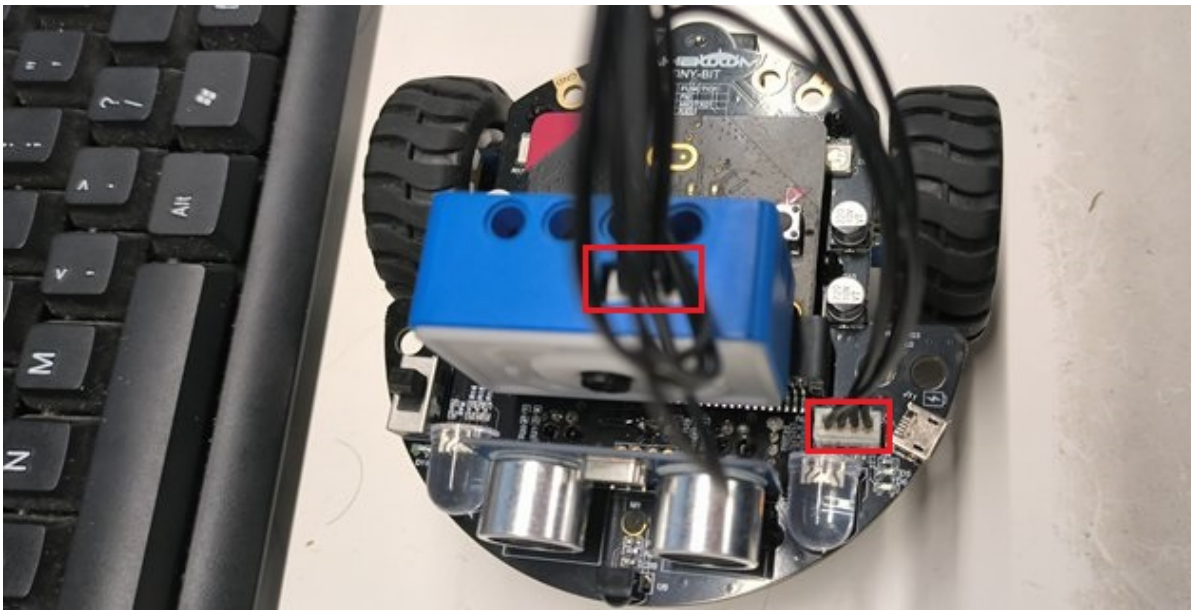
tinybit with camera

1. Experiment preparation

- tinybit car
  -microbit
- wifi camera

2. Experimental installation

Put the camera firmware on the car, as shown below

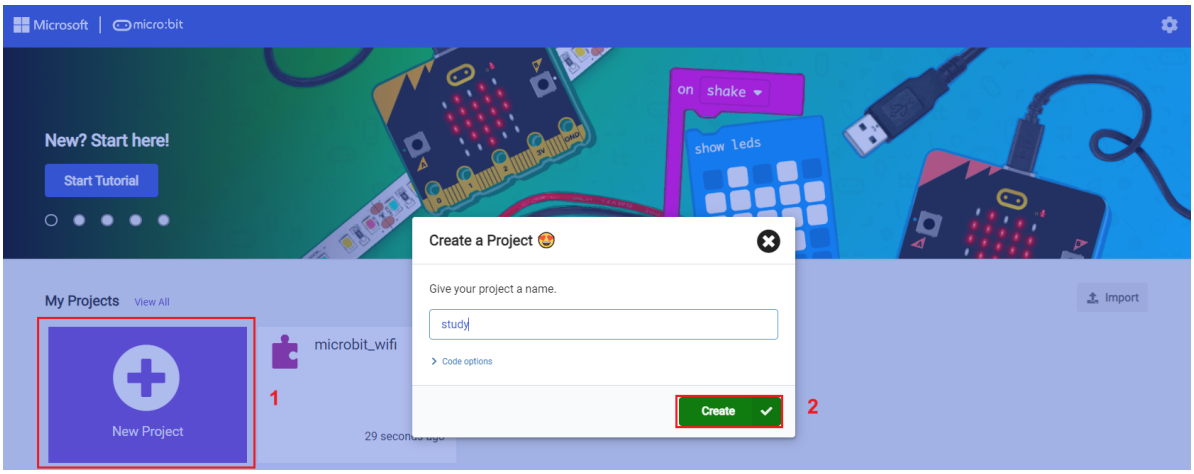


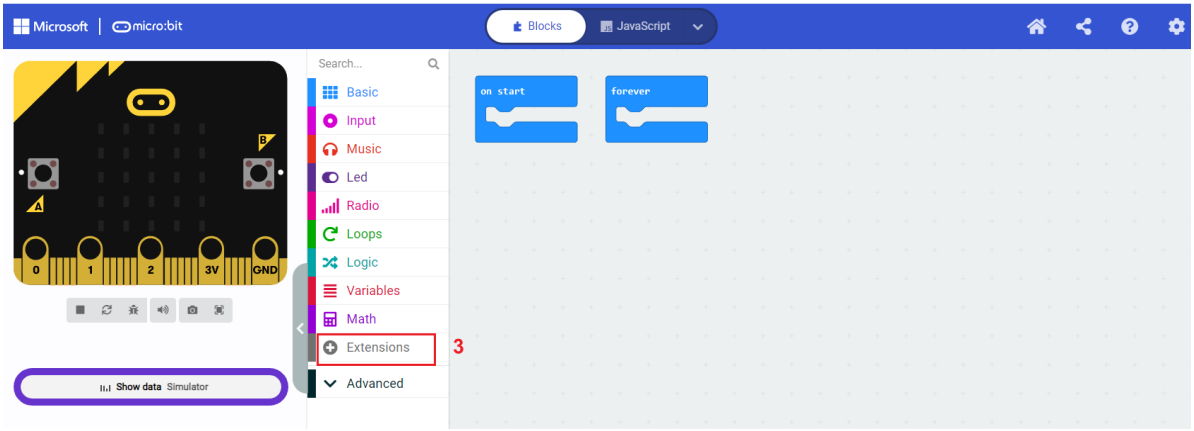
3. Microbit building block import and simple instructions

3.1 Open the programming website

https://makecode.microbit.org/#

3.2 New project

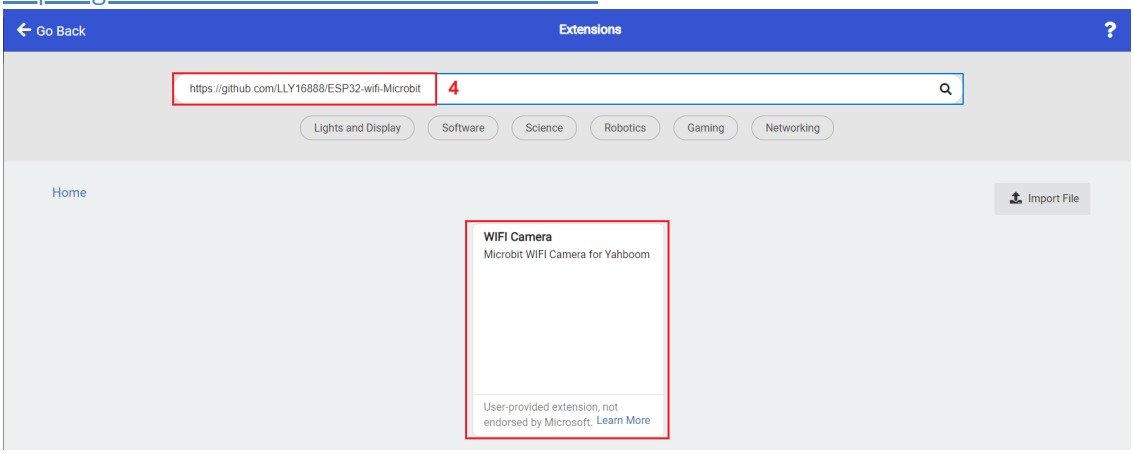




3.3 Add camera building blocks

URL of the building blocks: (Just choose one of them, they are the same)

1. https://github.com/yahboomtechnology/ESP32-wifi-Microbit
2. https://github.com/LLY16888/ESP32-wifi-Microbit

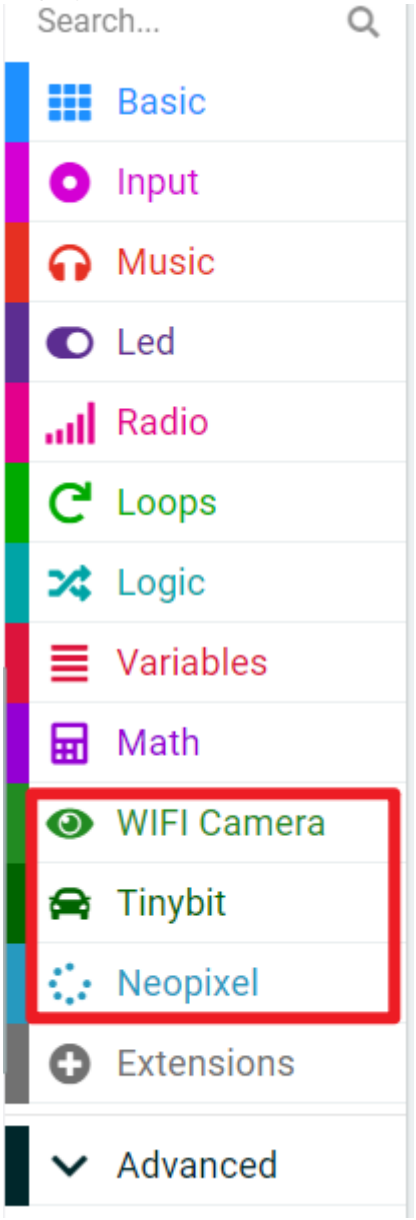


3.4 Add the building blocks of the car

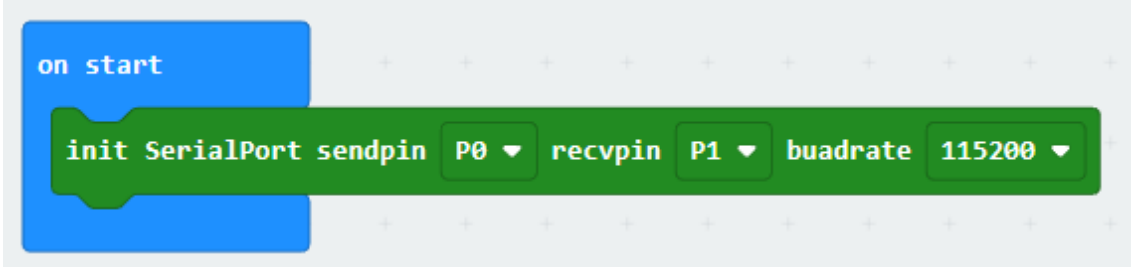
The URL of the car building blocks: https://github.com/YahboomTechnology/Tiny-bitLib.git

3.5 Introduction to the main building blocks

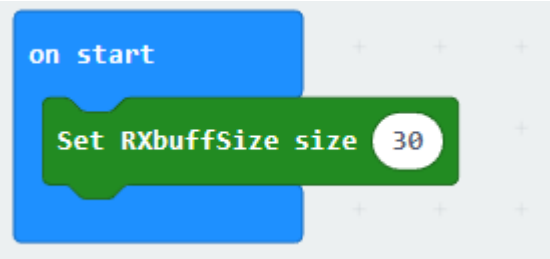
After the above building blocks are successfully introduced, the results as shown below will be displayed.



- **Serial port initialization building block** This is used to define the pins for serial port communication and communication with wifi cameras. The default baud rate is 115200 and cannot be changed.

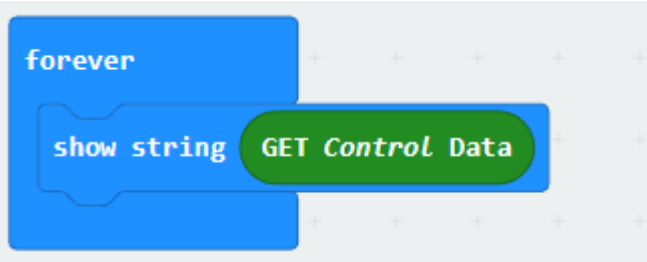


- **Set the size of the serial port receiving buffer** This building block is used to define the size of a packet of data that can be transmitted transparently, such as

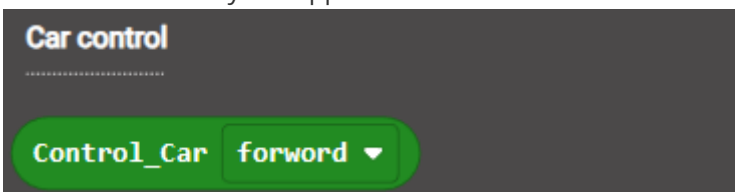


This defines the maximum size of a packet to be 30 characters. Exceeding it will result in incomplete data reception. **This value cannot be less than 25, otherwise the IP information will also be incomplete**

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



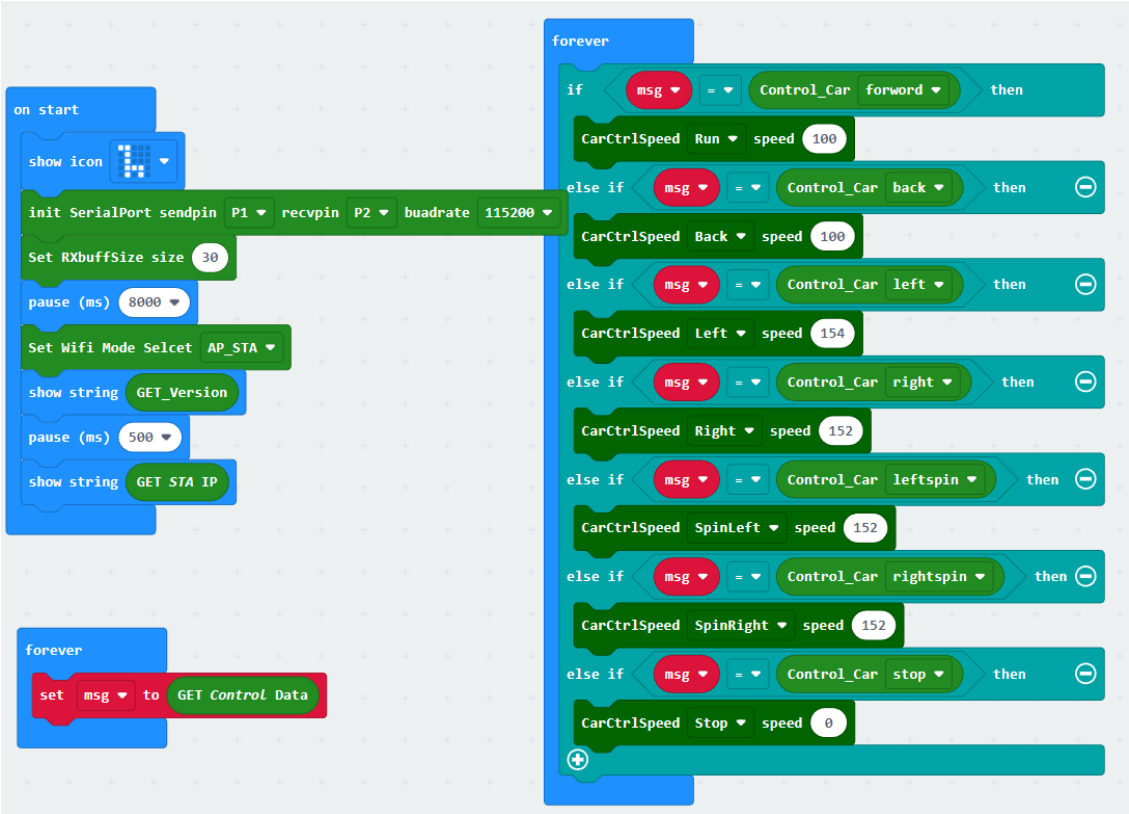
- **Building block controlled by the car** This building block is mainly used to receive the instructions sent by the app and transmit them to the car.



- You can know the function of other building blocks by looking at their names. How to use them can be found in the source code provided in this tutorial, which will not be explained in this tutorial.
  Open the source code method provided by the tutorial

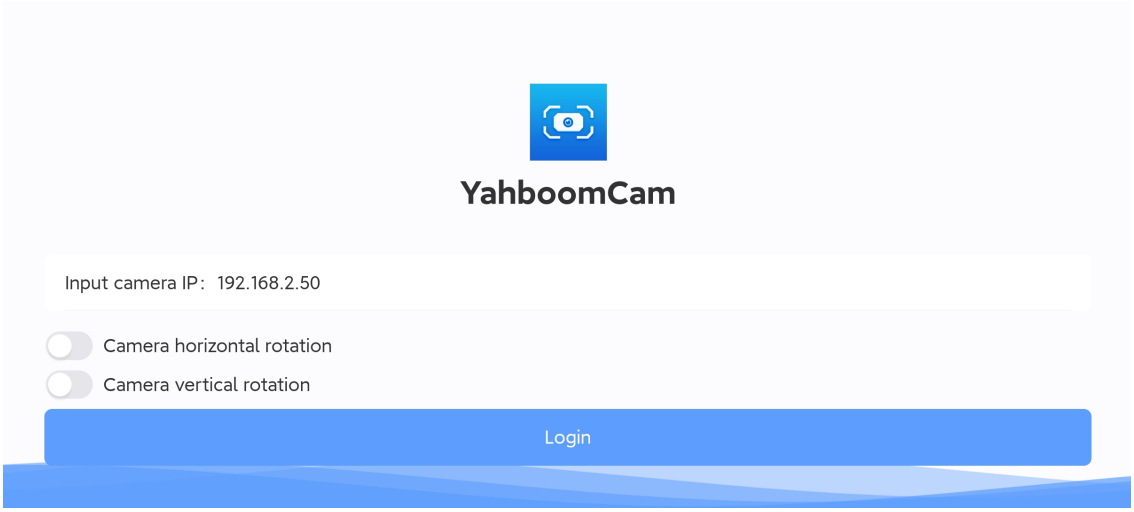
1. Open the URL https://makecode.microbit.org/# in the browser
2. Then drag the hex file provided in this experiment into the browser that opens the URL, and it will open automatically.

3. Program diagram of the source code of this project

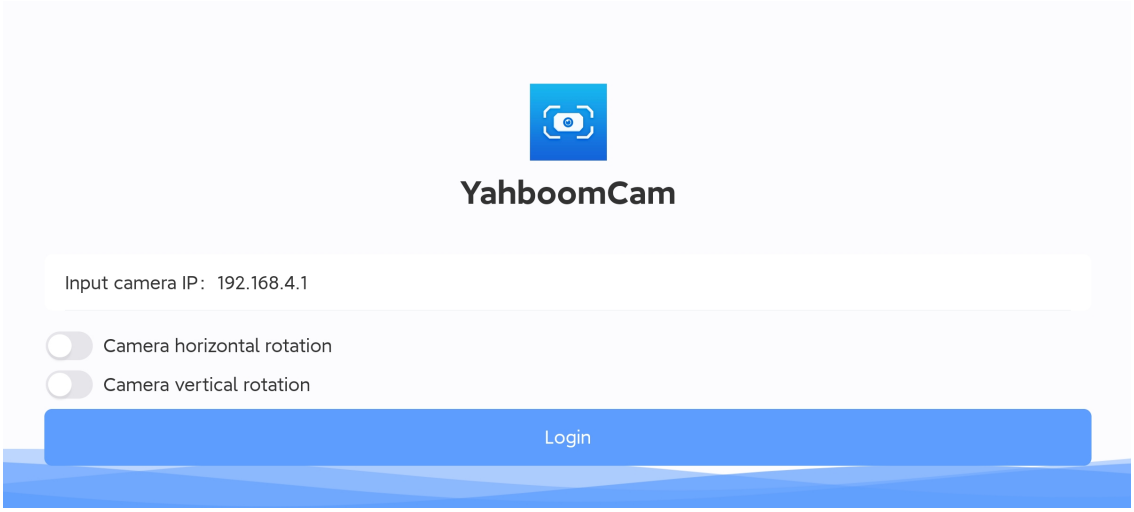


4. Experimental results

1. After powering on, wait until the microbit screen no longer displays anything before you can control and connect the app. The initial display is an icon. After the wifi is successfully started, its firmware version number will be displayed, and finally it will be displayed. IP address to connect to wifi.
2. Connect to the ip address displayed on the microbit through the app. For example, the ip address displayed on the microbit is 192.168.2.50. Then configure the connection interface as shown in the figure.



3. If you choose dual-mode coexistence (the experiment defaults to this mode), you can connect to its wifi (the default wifi name is Yahboom_ESP32_WIFI, no password), and then the ip address in the second step of the app can be changed. into 192.168.4.1



4. By clicking on the interface on the app, you can remotely control the car, move forward, backward, turn left, turn right, etc. (the car cannot be controlled by the steering gear), and the real-time picture of the camera is also displayed on the app page.
   Horizontal screen


```
Note: Every time you restart the app, you need to click the exit button in the upper right corner, then exit and reconfigure the IP address information before logging in.
```