

# superkit with camera

---

## superkit with camera

1. Experiment preparation
2. Experimental installation
3. Microbit building block import and simple instructions
  - 3.1 Open the programming website
  - 3.2 New project
  - 3.3 Add camera building blocks
  - 3.4 Add Surerkit building blocks
  - 3.5 Introduction to the main building blocks
4. Experimental results

## 1. Experiment preparation

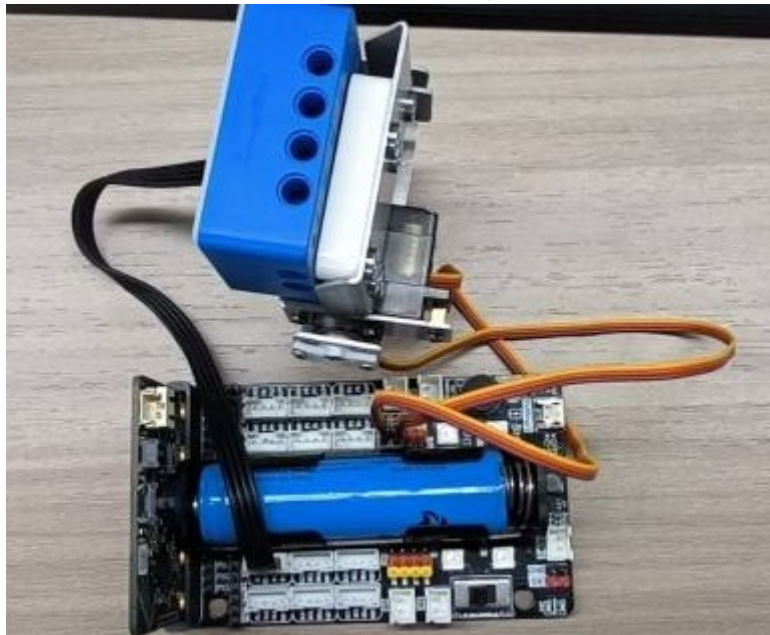
---

- superkit expansion board
- microbit
- wifi camera

## 2. Experimental installation

---

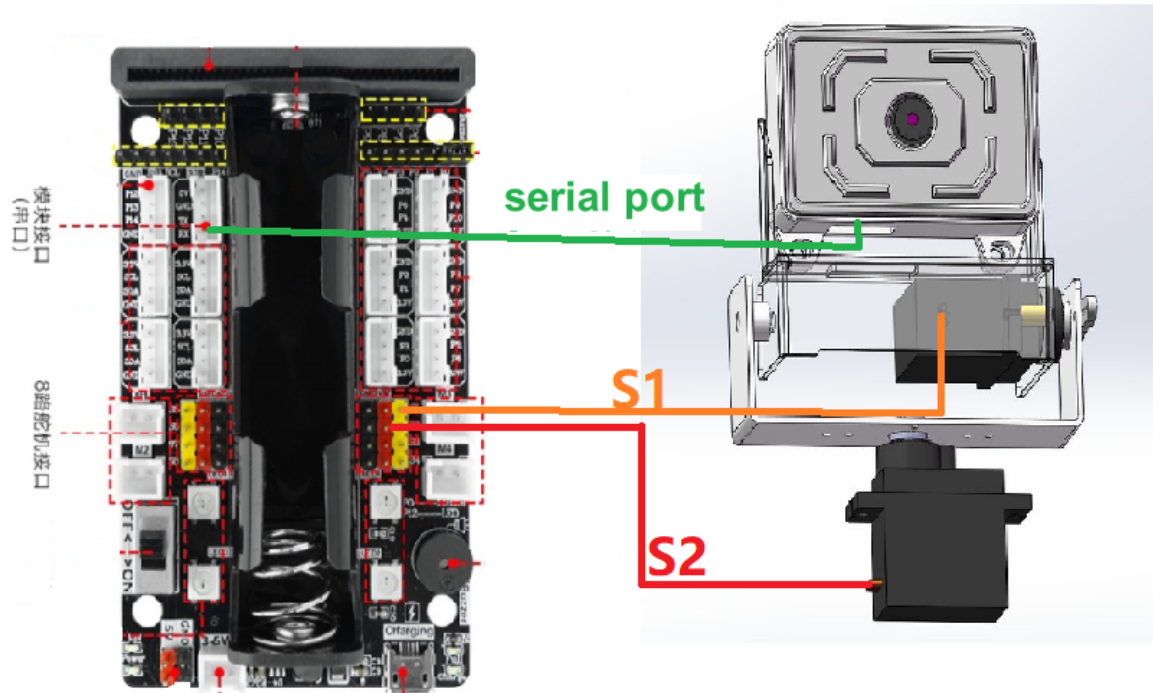
Connect the camera with gimbal and superkit, as shown below



Wiring instructions:

superkit	wifi camera
Servo interface S1	Vertical servo
Servo interface S2	Horizontal servo
Serial interface	Serial port for wifi camera

## Wiring diagram

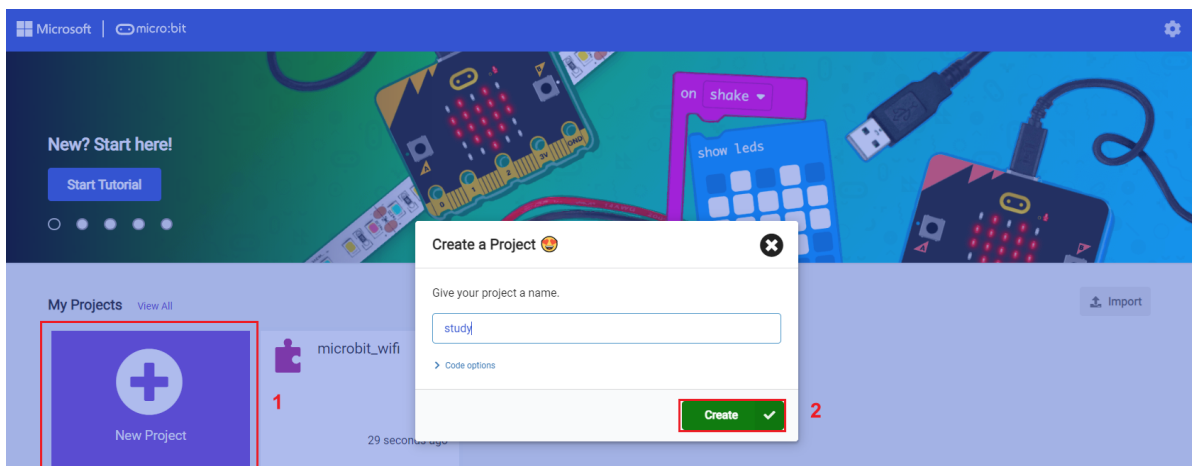


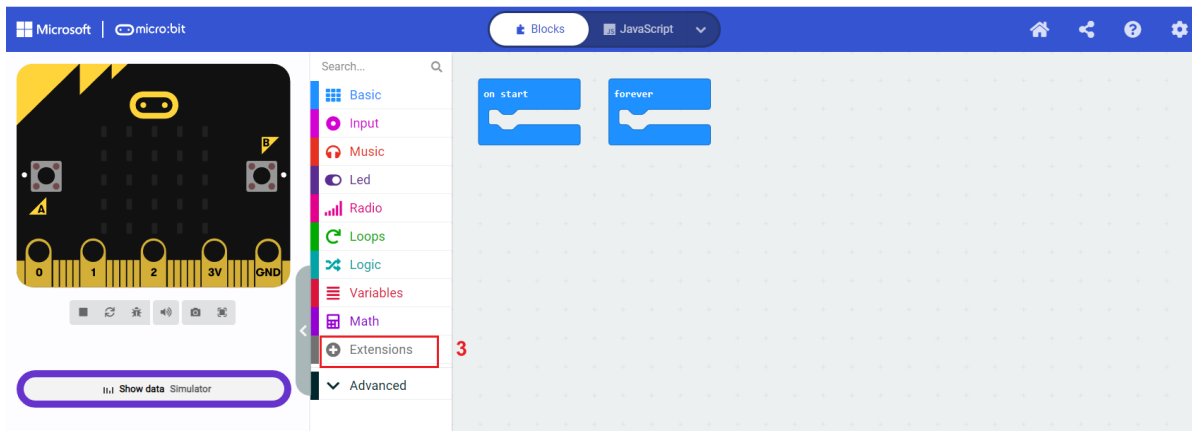
## 3. Microbit building block import and simple instructions

### 3.1 Open the programming website

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

### 3.2 New project

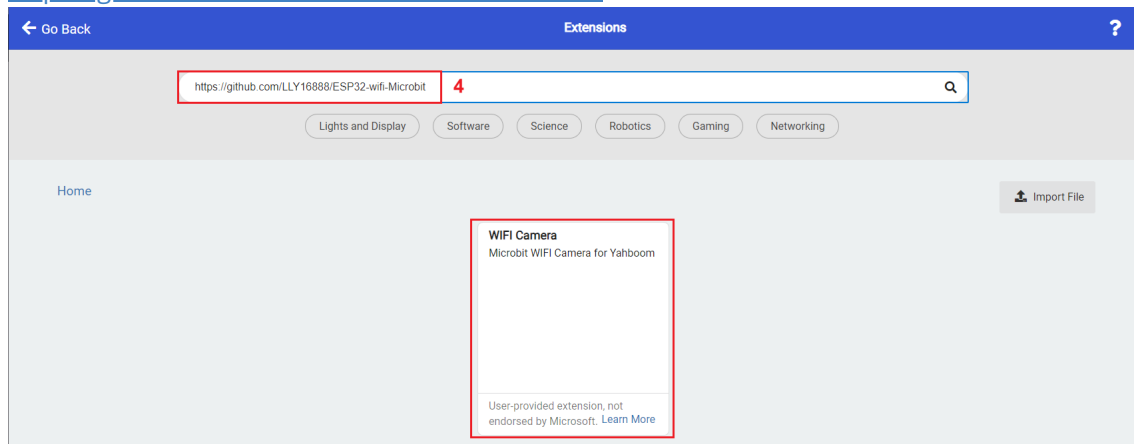




### 3.3 Add camera building blocks

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>

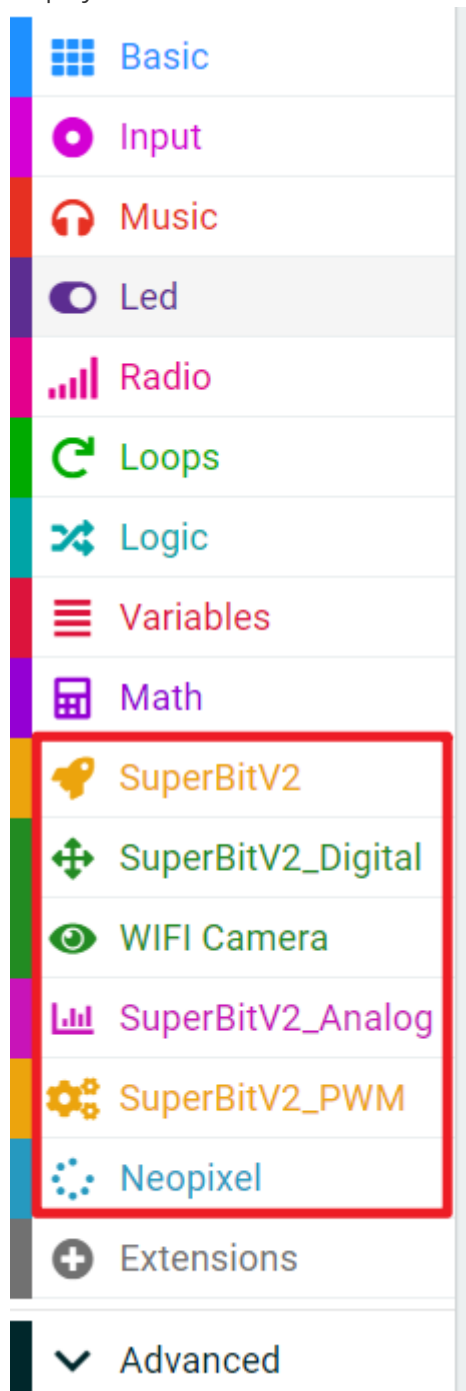


### 3.4 Add Surerkit building blocks

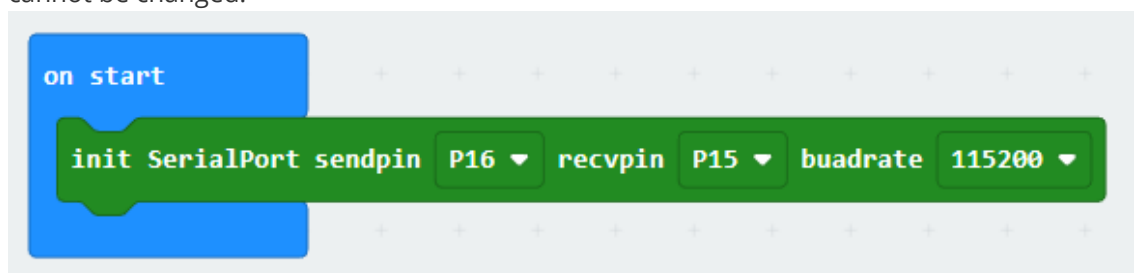
URL of Surerkit building blocks: <https://github.com/YahboomTechnology/SuperBitLibV2.git>

### 3.5 Introduction to the main building blocks

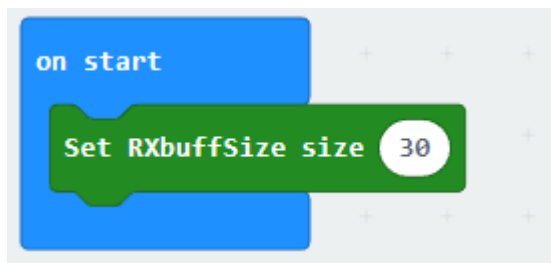
After the above building blocks are successfully introduced, the results as shown below will be displayed.



- **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.

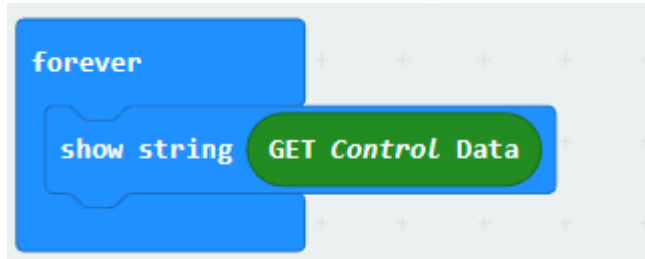


- **Set the size of the serial port receiving buffer** This building block is used to define the size of a packet of data that can be transmitted transparently, such as



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



- **The building block of the servo** This building block is mainly used to receive the instructions sent by the app and then transmit them to superkit for servo control.

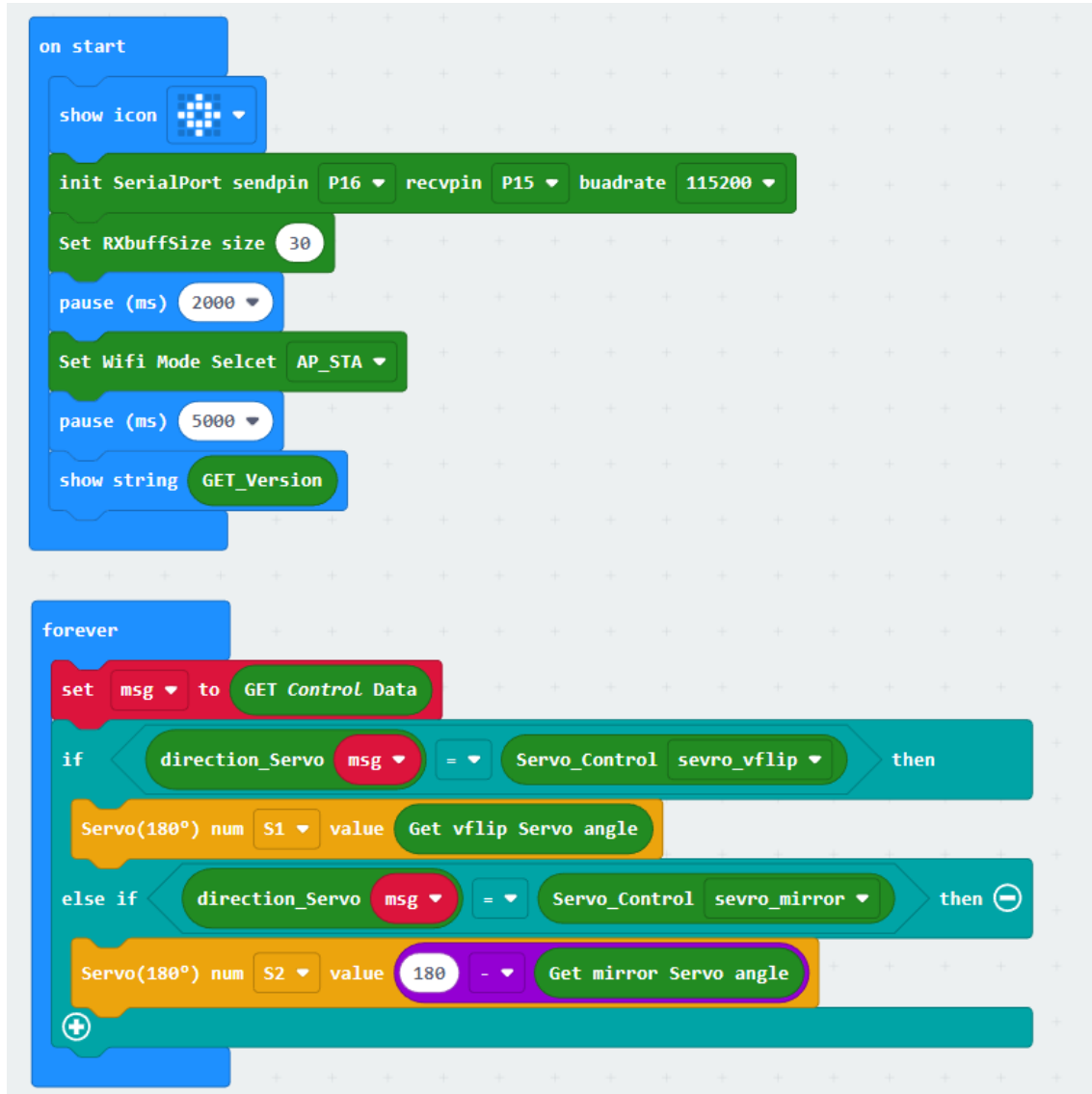


- You can know the function of other building blocks by looking at their names. How to use them can be found in the source code provided in this tutorial, which will not be explained in this tutorial.

#### **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 the app and connect. The initial display is an icon. After the wifi is successfully started, its firmware version number will be displayed.
2. If you want to choose dual-mode coexistence or spontaneous hotspot mode (the experiment defaults to dual mode), you can connect to its wifi (the default wifi name is Yahboom\_ESP32\_WIFI, no password), and then change the app's ip address to 192.168 .4.1



3. (Not necessary) You can query the STA mode IP of the wifi module by yourself, display it on the screen, record the IP, or connect it with the mobile app\*\* This method must turn on the STA mode or dual mode, and turn on the superkit Just wait 5 seconds to query the IP. There is no need to put it in an infinite loop, which will affect the use of the control function\*\*
4. By clicking on the interface on the app, you can control the servo gimbal up, down, left, and other operations, and the real-time image of the camera is also displayed on the app page.

#### Horizontal screen

