

Control buzzer and button

Control buzzer and button

1. Buzzer control
 - 1.1. Experimental goals
 - 1.2. Experiment preparation
 - 1.3. Experimental operations and phenomena
 - 1.3.1jetson motherboard/Raspberry Pi 4B
 - 1.3.2 Raspberry Pi 5
 - 1.4. Program source code
2. Button control
 - 2.1. Function of button K1
 - 2.2. Key K2 function
 - 2.3. RESET key function

1. Buzzer control

1.1. Experimental goals

Control the buzzer switch on the Transbot SE expansion board, and set the buzzer time to 100 milliseconds, 300 milliseconds, 1 second, etc.

1.2. Experiment preparation

The red box in the picture is the buzzer on the expansion board.

The buzzer on the Transbot SE expansion board is an active buzzer, so it is relatively simple to control. Please check the following functions.

Transbot_Lib library function needed to control the buzzer:

```
set_beep(on_time)
```

Parameter explanation: on_time=0: off, on_time=1: always beeping, on_time>=10: automatically turn off after beeping for xx milliseconds (on_time is a multiple of 10).

Return value: None.

1.3. Experimental operations and phenomena

1.3.1jetson motherboard/Raspberry Pi 4B

Please view the course video.

1.3.2 Raspberry Pi 5

Enter docker

Note: If you have a terminal that automatically starts docker, you can directly enter the temp directory in docker to view it. There is no need to manually start docker

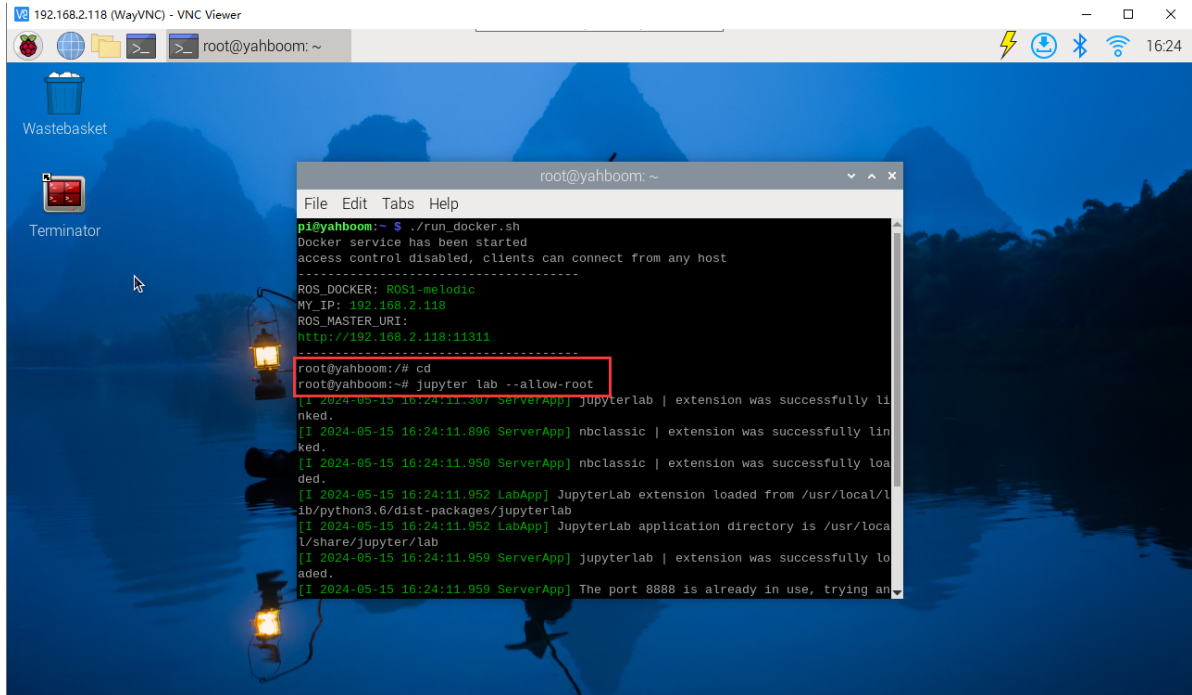
Start docker manually

```
./run_docker.sh
```

Run jupyter lab program

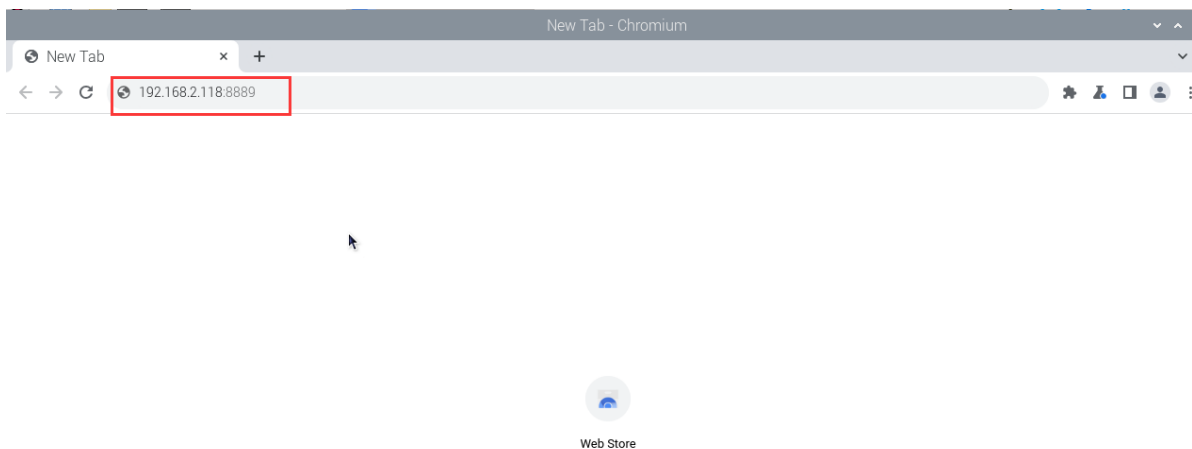
```
cd
```

```
jupyter lab --allow-root
```

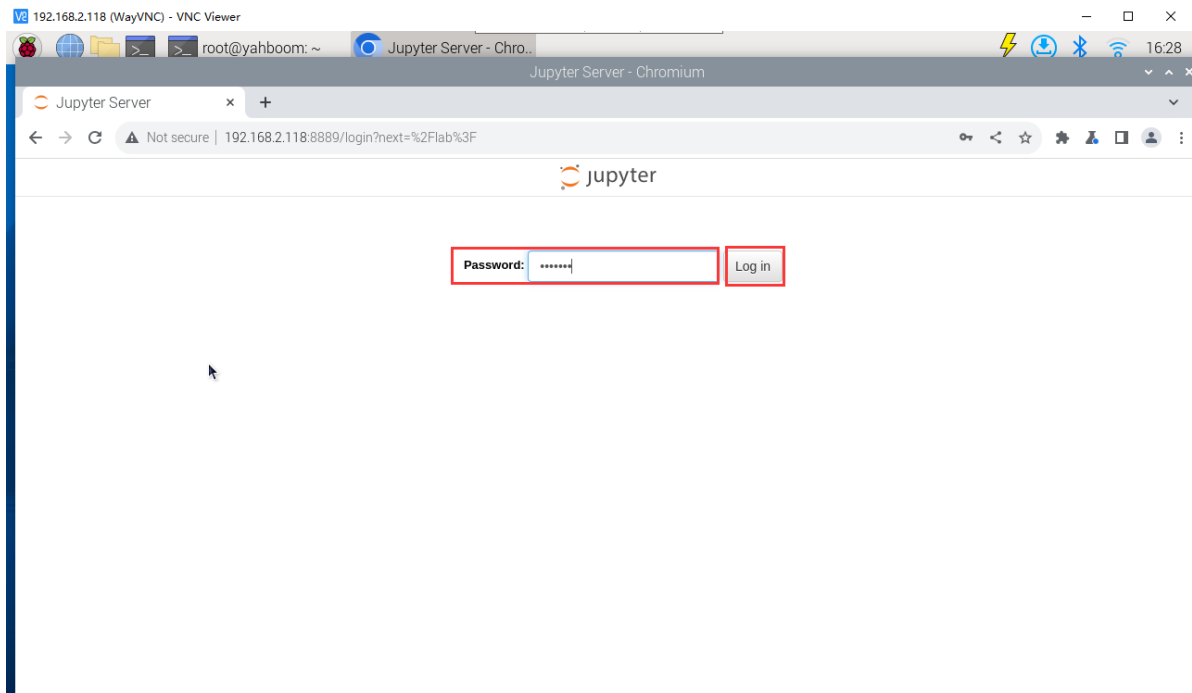


Taking the current IP address 192.168.2.118 as an example, open the browser of Raspberry Pi 5 or enter in the browser of your computer

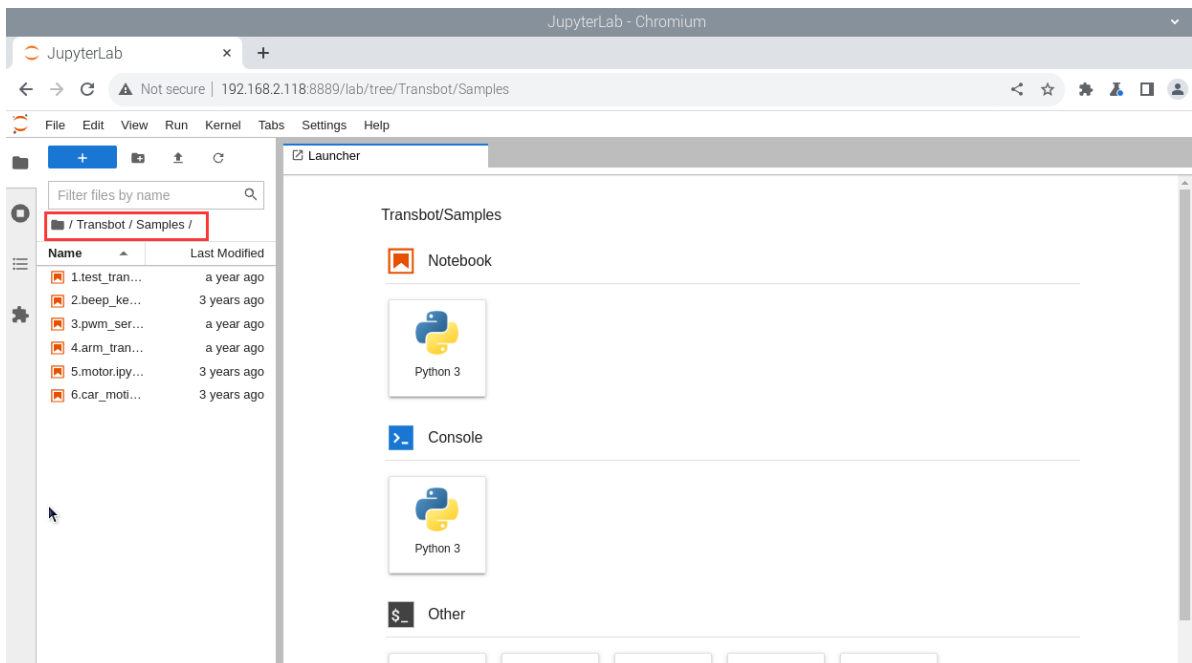
```
192.168.2.118:8889
```



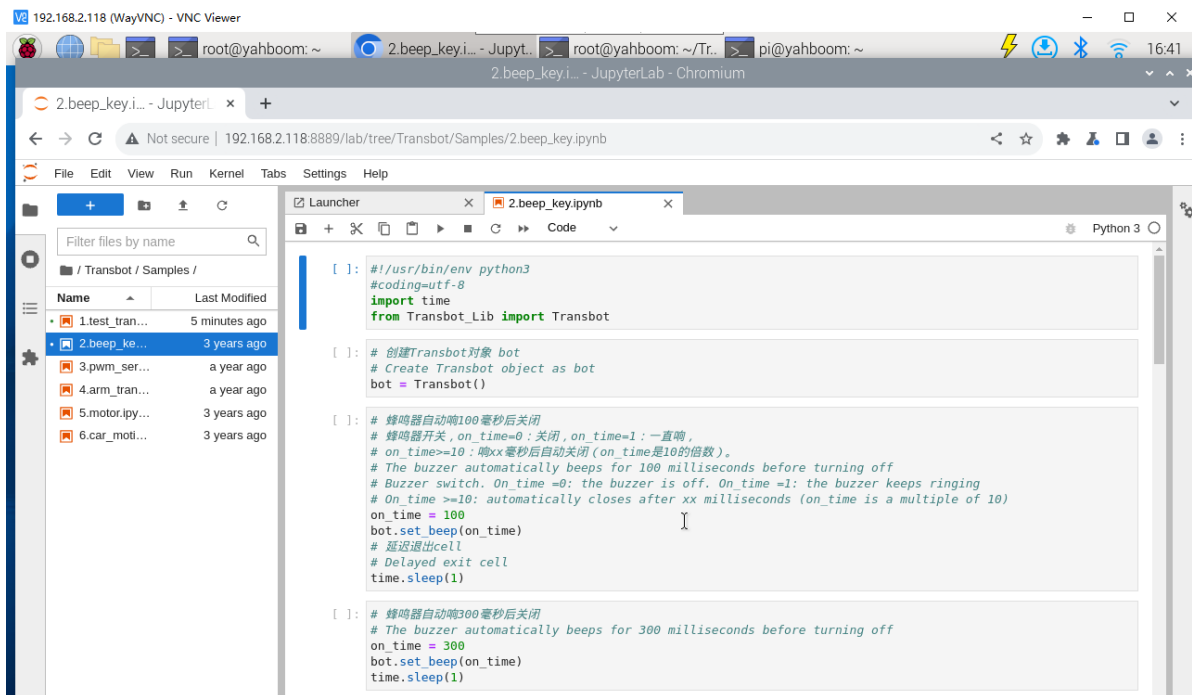
Enter the jupyter lab login interface after pressing Enter, enter the password yahboom, and then click login



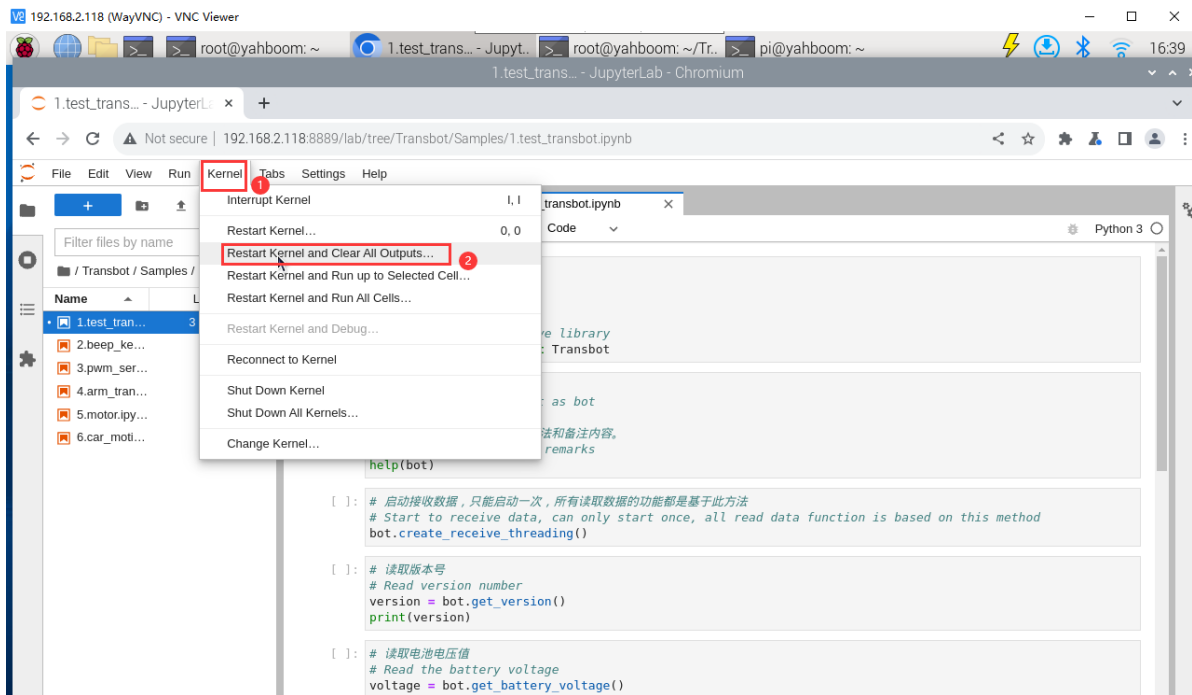
Enter the Transbot/Samples directory



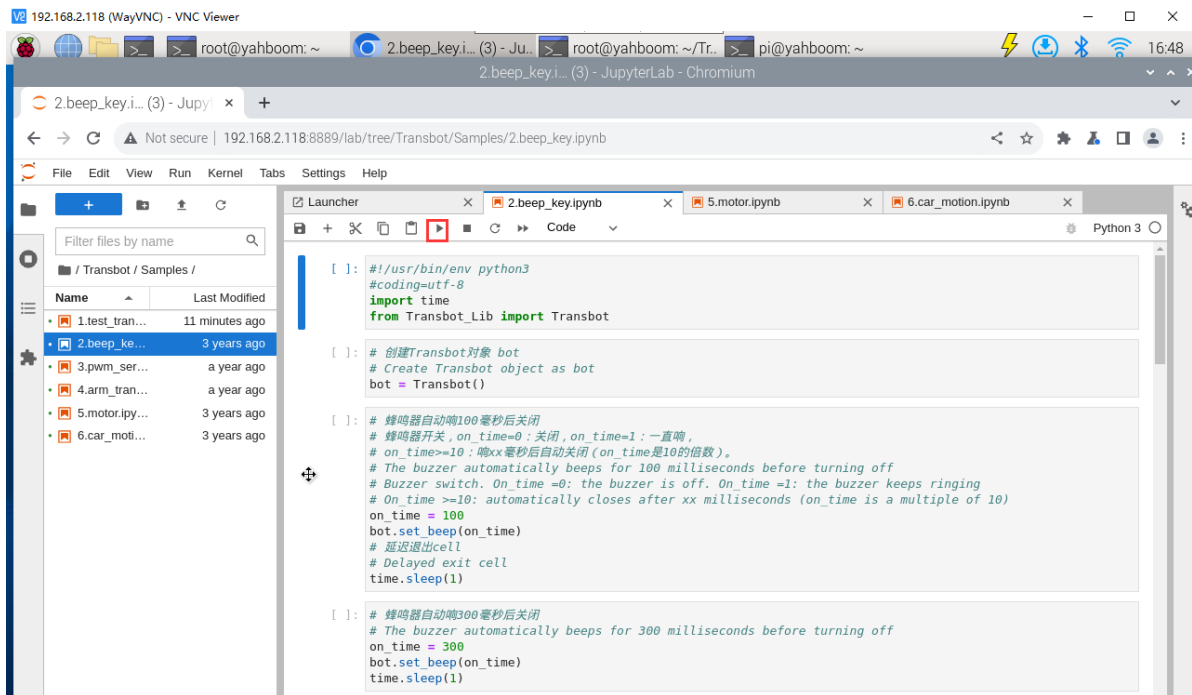
Double-click the code you want to run



Restart the kernel and clear all output before running



Click on the first code block, then click the run button to start running one by one



Turn on the power of the Transbot SE robot, and open Jetson Nano or the browser of the remote computer to enter the Jupyter lab editor.

Reference code path: Transbot/Samples/2.beep_key.ipynb

2. Button control

The blue boxes in the picture above correspond to the positions of the three buttons.

2.1. Function of button K1

Button K1 is connected to the Jetson Nano mainboard and is managed by Jetson Nano, so it has no corresponding function on the expansion board. Because the Raspberry Pi is in docker, the button K1 function is not available. By default, after the Transbot SE program is started, long press the K1 button to enter the network configuration mode. You can use the APP to generate a QR code for the camera to scan and connect to the network. For specific functions and uses, please see the content of [Camera WiFi Network Configuration Tutorial].

With the large program closed, you can customize the function of button K1 on the Jetson Nano. Button K1 is connected to the physical pin 11 of the Jetson Nano, and the BCM number is 17. Raspberry Pi does not have this usage.

2.2. Key K2 function

The buzzer will sound once every time the button K2 function is pressed.

After long pressing button K2 for about 10 seconds, the siren will sound for one second, indicating that the microcontroller is restored to factory settings, that is, the data area in Flash is restored to default.

Button K2 is only connected to the microcontroller on the expansion board. The function of this button has been fixed and cannot be customized.

2.3. RESET key function

The reset button can restart the microcontroller on the expansion board.