1.3 Precautions before development

If you use image we provided.
User name： **jetbot**
Password： **yahboom**

All set passwords used in the system are **yahboom**.

Jetbot factory firmware opening opportunity to run through the system service self-starting APP program:

①If you want to temporarily turn off the app from the startup, execute the following command:
sudo systemctl stop jetbot_ start

②If you need to temporarily open the APP program that starts up and start, execute the following command:
sudo systemctl start jetbot_ start

③Or use the following command to manually run the APP program after closing the APP large program service process.
(This method can view the debugging information of the relevant output)
cd yahboom-jetbot/
sudo python3 yahboom- jetbot . pyc

④To permanently disable the function of the APP program that is turned on, execute the following command:
sudo systemctl disable jetbot_ start

⑤To permanently disable the function of the APP program that is turned on, execute the following command:
sudo systemctl enable jetbot start

⑥If you do not restart the APP program by restarting Jetbot, execute the following command:
sudo systemctl stop jetbot_ start
sudo systemctl start jetbot_ start

- **When we use APP control:**

1). Because the AI model is involved during booting, the time may be within 2-3 minutes. Please wait patiently for the boot start signal. If you have not successfully started after waiting for a long time, please try to restart Jetbot.

2). When you open the Jetbot and enter the function interface with the camera real-time screen for the first time, the camera will be initialized and the camera

driver will be started. After waiting for a short time, the image will be displayed. Before this, the image display frame is displayed as white without image, this status is normal.

3). When you first enter the automatic avoid function interface,Jetbot needs to wait for a short period of time to run the avoid model to the memory for the first time. Please wait patiently until the model is loaded.

4). When you first open the automatic follow function interface, Jetbot needs to wait for a short time to run the object model to the memory for the first time. Please wait patiently until the model is loaded.

5). color tracking may be affected by the brightness of the surrounding environment, in the dark environment may cause deviations in the recognized color, it is not hardware or software problems. In order to obtain better functional effects, please in a more light environment Use below.

6). Do not arbitrarily change the format of the configuration file config.txt file. You must modify the parameters according to the original file format to ensure that the configuration in the file is correctly read.

- **When we developed the JETBOT car by ourselves:**

①Before we develop other courses, we need to enter the following command to manually start the service and end the APP process to release system resources

sudo systemctl stop jetbot_start

②Confirm servo installation angle.

If WIFI is not configured, you can use "headless mode" to connect.

In Jupyter lab, run as show below code, reset the servo to the center position (the middle position of horizontal movement in left and right direction is 2100, and the middle position of vertical movement in vertical direction is 2048).

After program is run, then we can fix servo.

Import Yahboom official packaged serial bus servo drive library

Create and initialize a programmable servo control object

```
from servoserial import ServoSerial
import time
servo_device = ServoSerial()
```

Yahboom

Make servo rotate to the center position by asynchronous commands

```
servo_device.Servo_serial_control(1, 2100)
time.sleep(0.1)
servo_device.Servo_serial_control(2, 2048)
```

- **Camera abnormal processing and detection:**

When we may develop for a period of time due to some unconventional operations (such as not shut down and power off), the camera calls abnormally in Jupyter, and it will flashes.

When we may develop for a period of time due to some unconventional operations (such as not shut down and power off), the camera calls abnormally in Jupyter, and flashes.

Solution:

①Camera detection:

If you have display, enter following command in Jetbot command console:

nvgstcapture-1.0

If the camera is normal, the screen will be displayed. Press ctrl + C to exit and close the window.

②We need to end the Jupyter thread to clear the use cache, and then re-fresh Jupyter again to return work normally.

Input following command:

sudo systemctl stop jetbot_jupyter

Input following command:

sudo systemctl start jetbot_jupyter

Or you can use **top** command in Jetbot command console to find jupyter thread number.

Then, you can use following command to kill jupyter thread.

sudo kill -9 xxxx

After closing the thread, wait for the thread to start.

- **Course that need a handle**

We need to use the handle in the chapter of using the handle and the chapter of autopilot to complete the course.

- **When developing in Jupyter, the camera interface can only open one instance.**

The camera may not be called because the camera interface was not released because the thread that used the camera was completely closed. We can restart Jetbot to solve it.

sudo reboot

- **Features that need to be matched with the PTZ kit**

Face tracking, color tracking and Object follow(non-human) are also use vertical PTZ, but without a PTZ there will not be a direct impact on the function.

- **Resolution of autonomous obstacle avoidance and object following**

In the applications of autonomous obstacle avoidance and object following, the models of the two applications of autonomous obstacle avoidance and object following are inconsistent. Do not change the resolution, because the input format of the neural network they trained is the format listed below:

obstacle avoidance model 224 224

object detection 300 300

When these two models are to be applied to one application at the same time, we can make the large resolution can be adjusted to a small resolution through the OpenCV resize method!

- **The problem of network connection failure.**

If the network connections are normal, it is usually the cause of addressing failure in the local area network. Repeat the re-connection operation several times, or use ping to ping IP each other.

About jetbot car, you can refer to [FAQ]---[Q1:If everything is configured, the phone still can't connect to Jetbot.]

- **Desktop switching**

If you need an HDMI screen for desktop operation, the Jetbot factory configuration image has three desktops: Unity desktop, GNOME desktop, and xfce desktop.

Unity is a smooth native desktop for Ubuntu. GNOME and xfce are third-party desktops.

NOME is mainly used for VNC remote desktop.

xfce desktop is mainly used for remote desktop that comes with Windows.

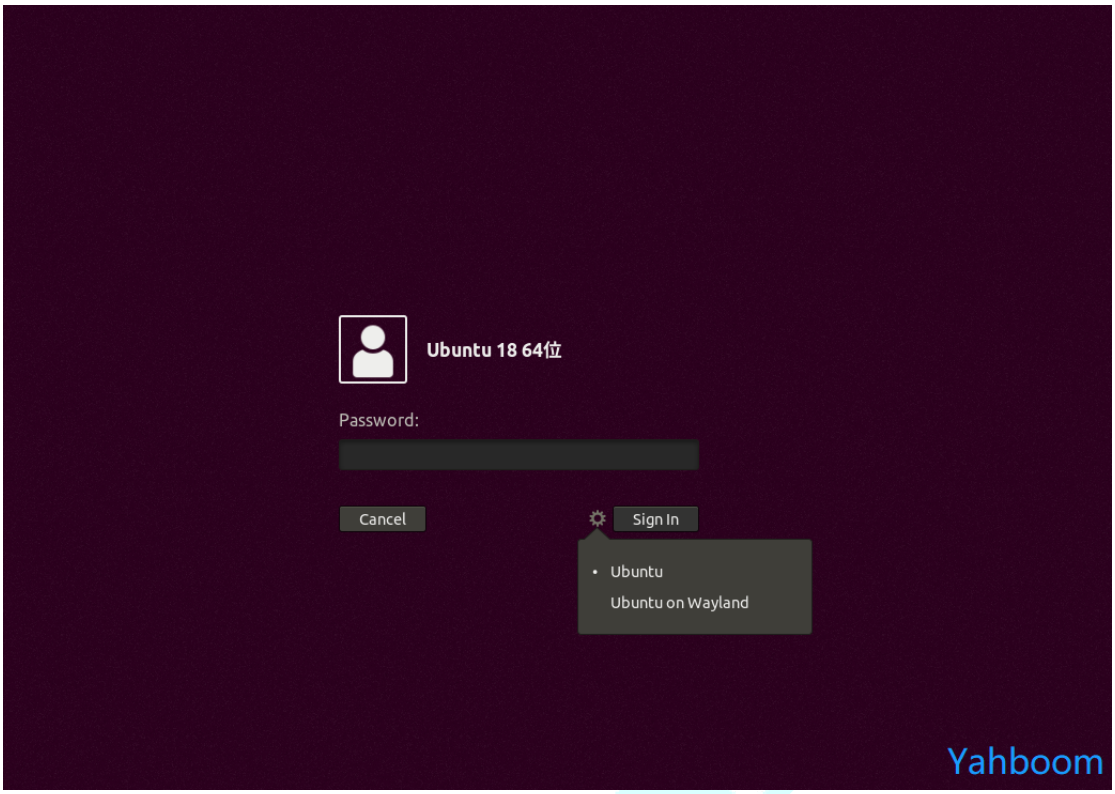
Considering that users may need VNC remote desktops to connect to Jetbot, the default factory login is the GNOME desktop, because the Unity desktop does not support VNC remote.

May be VNC remote desktop stuttering seriously affects the use. I generally use the Unity local desktop with higher fluency.

The remote desktop that comes with windows is connected to the xfce remote desktop.

If you want to switch desktops, first choose to log out of the account.

Then, you can choose the desktop you want to log in on the login interface.As shown below:



- **Object following---avoid version**

This function needs to adjust the camera to the direction angle of obstacle avoidance training to obtain good obstacle avoidance and following effects.