Control robot movement

1. Experimental goals

Control the forward and backward movement of Transbot-SE, turn left and right, and set the relevant parameters of the car.

2. Experiment preparation

The position of the red box in the picture below is the interface of motor MOTOR A on the left and motor B on the right. The motor interface has an anti-reverse connection function and can be connected to the motor using the Transbot SE motor cable.

The interface line sequence corresponding to the left and right motors is as shown in the figure below:

Transbot_Lib library function needed to control the motor of Transbot SE:

```
set_car_motion(velocity, angular)
```

Parameter explanation: Car motion control, this function will read the pulse data of the encoder to calculate the speed of the car movement. Velocity represents the speed of the car, the unit is m/s, the positive number is forward, the negative number is backward, the angular control represents the rotation speed of the car, the unit is rad/s, the positive number is left rotation, and the negative number is right rotation.

velocity=[-0.45, 0.45]

angular=[-2.00, 2.00]

Return value: None.

```
set_pid_param(kp, ki, kd, forever=False)
```

Parameter explanation: PID parameter control will affect the change of the car's movement speed controlled by the set_car_motion function. By default, no adjustment is required.

kp ki kd = [0, 10.00], decimals can be entered.

forever=True is permanently saved, =False is temporarily used.

Permanent storage is to write the data into the Flash of the microcontroller chip. The data will not be lost after restarting, and the writing time is long, so a delay time is added to avoid the problem of packet loss caused by the microcontroller. The temporary effect responds quickly and is effective once, and the data is no longer retained after restarting the single chip.

Return value: None.

```
set_speed_limit(line_limit, angular_limit, forever=False)
```

Parameter explanation: Set the minimum speed limit value (minimum value) of the car movement. It is not set by default.

For example: when the line speed limit value is set to 0.1, the next time the set_car_motion(0.05, 0) function is called, the microcontroller determines that 0.05 is less than 0.1, and automatically sets the input 0.05 to 0.1. If it is set_car_motion(-0.05, 0) , then -0.05 will be set to -0.1. The angular velocity limit value also has the same function.

line_limit=[0, 0.20], angular_limit=[0, 1.00]

forever=True is permanently saved, =False is temporarily used.

Return value: None.

```
set_imu_adjust(enable, forever=False)
```

Parameter explanation: Set the gyroscope to assist in adjusting the direction of the car's movement. After the setting is turned on, the set_car_motion function takes effect when the angular velocity is set to 0. When the car moves forward, due to the external force causing its direction of movement to deviate from the starting point, the gyroscope will adjust the car's forward direction to the original forward direction by adjusting the left and right speed of the motor. This function is only an auxiliary function and cannot guarantee that the car will move completely in a straight line. The same principle applies to retreating.

Due to the vibration of the car and the drift of the gyroscope itself, after turning on the gyroscope to adjust the direction, it cannot keep moving for a long time, otherwise the gyroscope will interfere with the movement of the car, causing one wheel to be faster and the other to be slower. Please turn this function off when testing for a long time.

enable=True turns on, =False turns off.

forever=True is permanently saved, =False is temporarily used.

Return value: None.

```
set_auto_report_state(enable, forever=False)
```

Parameter explanation: The microcontroller automatically returns the data status bit. The default is on. If set to off, it will affect the data reading function.

enable=True, the underlying expansion board will send data every 40 milliseconds. If enable=False, it will not be sent.

forever=True is permanently saved, =False is temporarily used.

Return value: None.

```
clear_auto_report_data()
```

Parameter explanation: Clear the cache data automatically sent by the microcontroller.

Return value: None.

```
reset_flash_value()
```

Parameter explanation: Reset the data saved in the car's flash and restore to factory default values. This function can also be achieved by pressing and holding the K2 key on the expansion board for about 10 seconds.

Return value: None.

The following functions all return data, and the data needs to be read after create_receive_threading() is started normally:

```
get_accelerometer_data()
```

Parameter explanation: Obtain three-axis accelerometer data

Return value: a_x, a_y, a_z

```
get_gyroscope_data()
```

Parameter explanation: Obtain gyroscope three-axis data

Return value: g_x, g_y, g_z

```
get_motion_data()
```

Parameter explanation: Get the car speed, return linear speed v, angular speed a

Return value: v, a

```
get_motion_pid()
```

Parameter explanation: Get the motion PID parameters of the car, and read error returns [-1, -1, -1]

Return value: kp, ki, kd

```
get_imu_state()
```

Parameter explanation: Get the status of gyroscope-assisted car movement. Returning True means turning it on, returning False means turning it off. If the reading fails, it returns -1.

Return value: Gyroscope adjustment status

3. Experimental results

Note: If you modify part of the configuration in this course and set it to be permanently saved, it will cause abnormal situations in the future (for example, set the automatic return data status enable=False and permanently save it to True, resulting in the inability to read the data normally). Please call the reset_flash_value() function, or press and hold the K2 key on the expansion board for about 10 seconds to restore the factory configuration.

3.1jetson motherboard/Raspberry Pi 4B

Please view the course video.

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

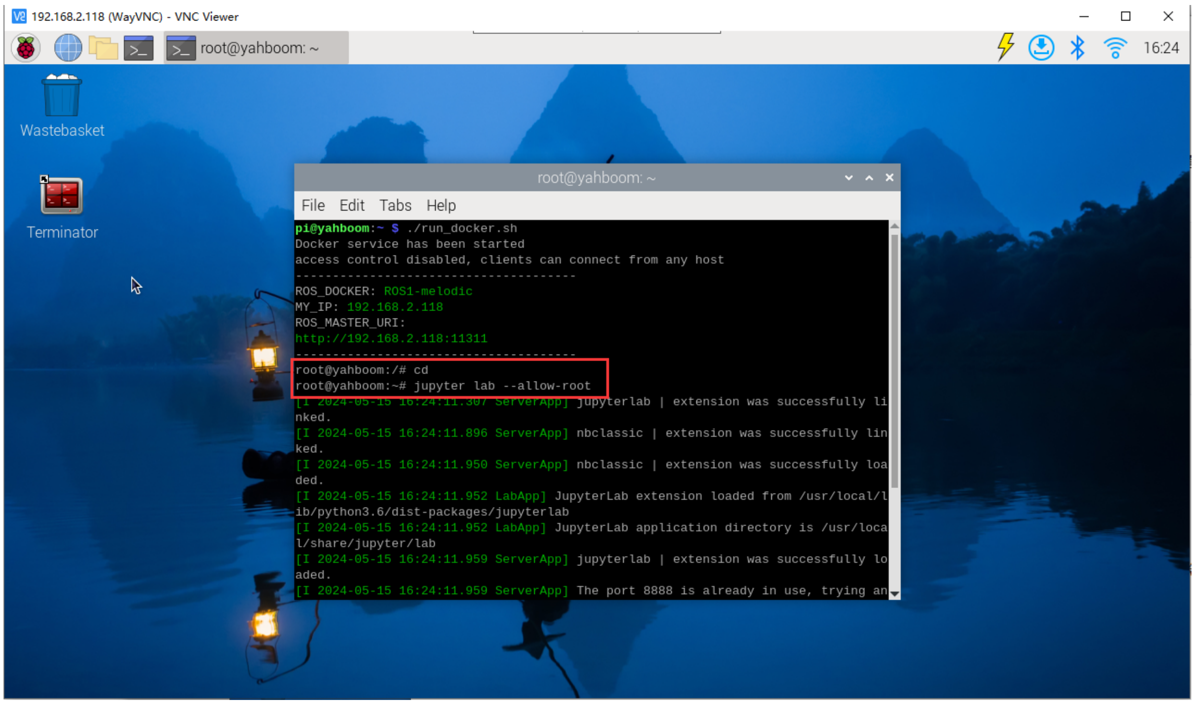
Start docker manually

```
./run_docker.sh
```

Run jupyter lab program

```
cd
```

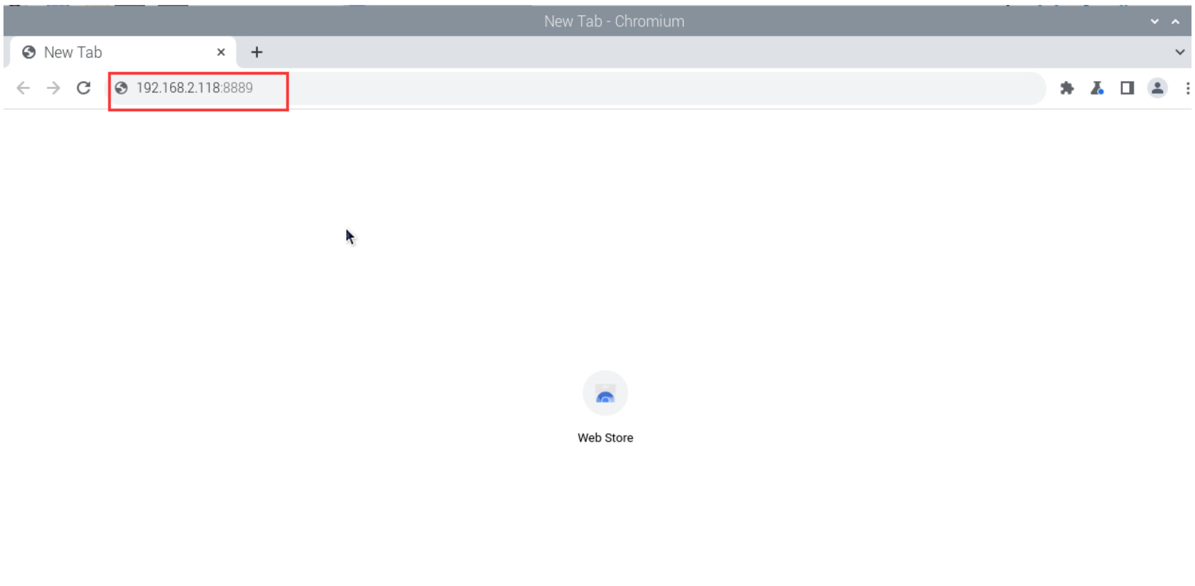
```
jupyter lab --allow-root
```



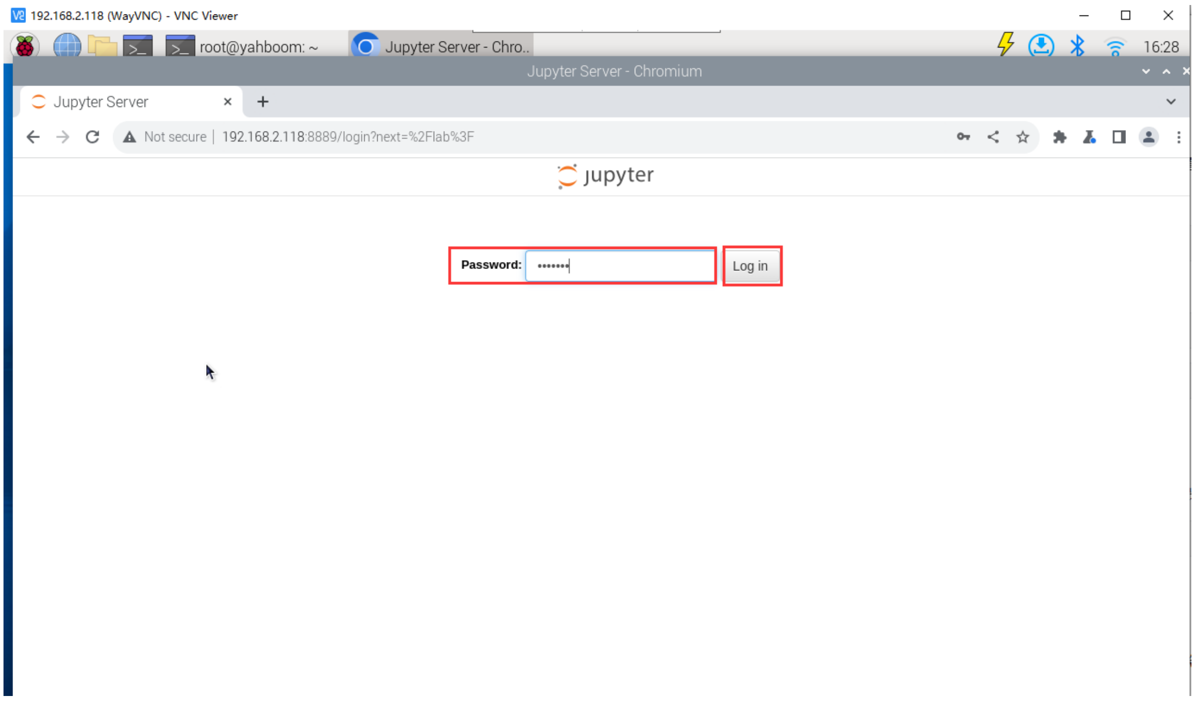
```
pi@yahboom:~ $ ./run_docker.sh
Docker service has been started
access control disabled, clients can connect from any host
-------------------------------------
ROS_DOCKER: ROS1-melodic
MY_IP: 192.168.2.118
ROS_MASTER_URI:
http://192.168.2.118:11311
-------------------------------------
root@yahboom:/# cd
root@yahboom:~# jupyter lab --allow-root
[I 2024-05-15 16:24:11.307 ServerApp] jupyterlab | extension was successfully li
nked.
[I 2024-05-15 16:24:11.896 ServerApp] nbclassic | extension was successfully lin
ked.
[I 2024-05-15 16:24:11.950 ServerApp] nbclassic | extension was successfully loa
ded.
[I 2024-05-15 16:24:11.952 LabApp] JupyterLab extension loaded from /usr/local/l
ib/python3.6/dist-packages/jupyterlab
[I 2024-05-15 16:24:11.952 LabApp] JupyterLab application directory is /usr/loca
l/share/jupyter/lab
[I 2024-05-15 16:24:11.959 ServerApp] jupyterlab | extension was successfully lo
aded.
[I 2024-05-15 16:24:11.959 ServerApp] The port 8888 is already in use, trying an
```

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

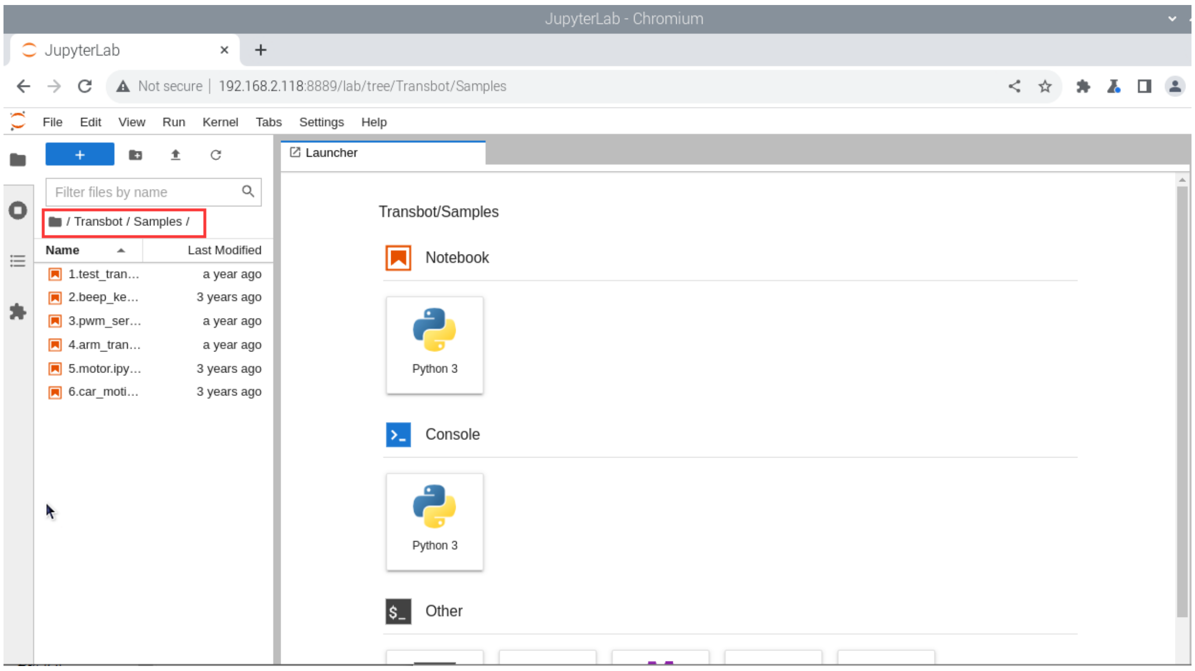
```
192.168.2.118:8889
```



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

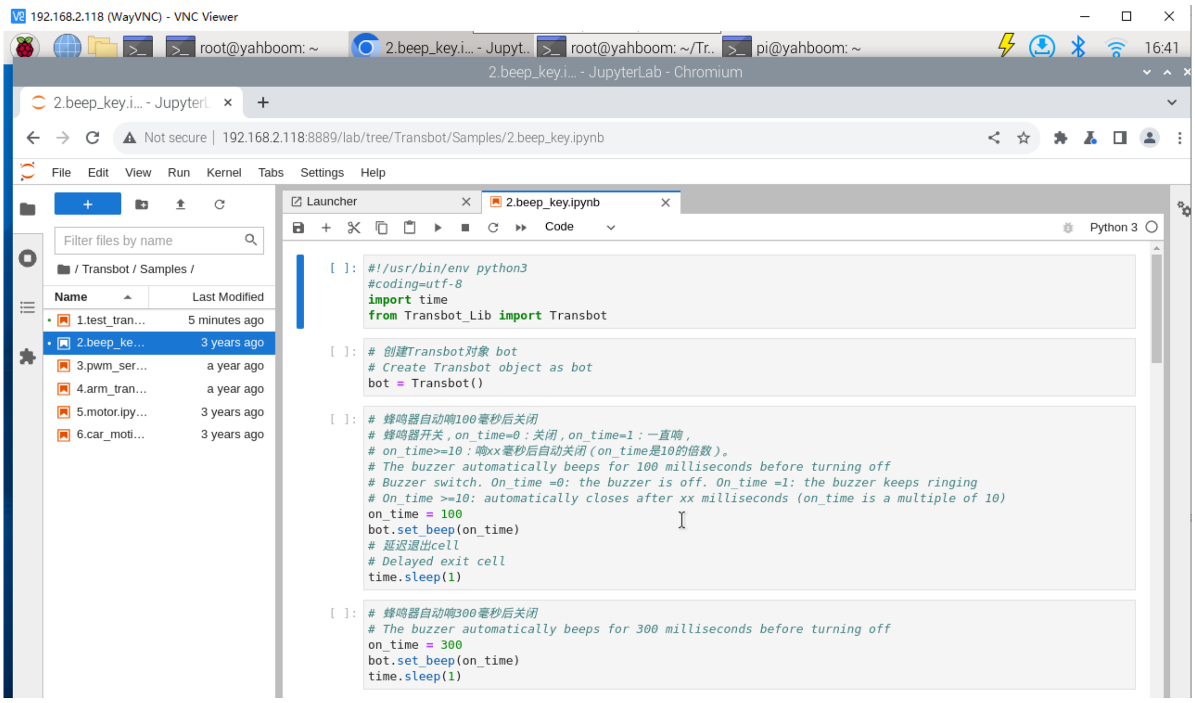

192.168.2.118 (WayVNC) - VNC Viewer
root@yahboom: ~
Jupyter Server - Chro..
16:28
Jupyter Server - Chromium
Jupyter Server
Not secure
192.168.2.118:8889/login?next=%2Flab%3F
jupyter
Password:
Log in

Enter the Transbot/Samples directory



Name	Last Modified
1.test_tran...	a year ago
2.beep_ke...	3 years ago
3.pwm_ser...	a year ago
4.arm_tran...	a year ago
5.motor.ipy...	3 years ago
6.car_moti...	3 years ago

Double-click the code you want to run


192.168.2.118 (WayVNC) - VNC Viewer
2.beep_key.i... - JupyterLab - Chromium
192.168.2.118:8889/lab/tree/Transbot/Samples/2.beep_key.ipynb
Launcher
2.beep_key.ipynb
Python 3

Name	Last Modified
1.test_tran...	5 minutes ago
2.beep_ke...	3 years ago
3.pwm_ser...	a year ago
4.arm_tran...	a year ago
5.motor.ipy...	3 years ago
6.car_moti...	3 years ago

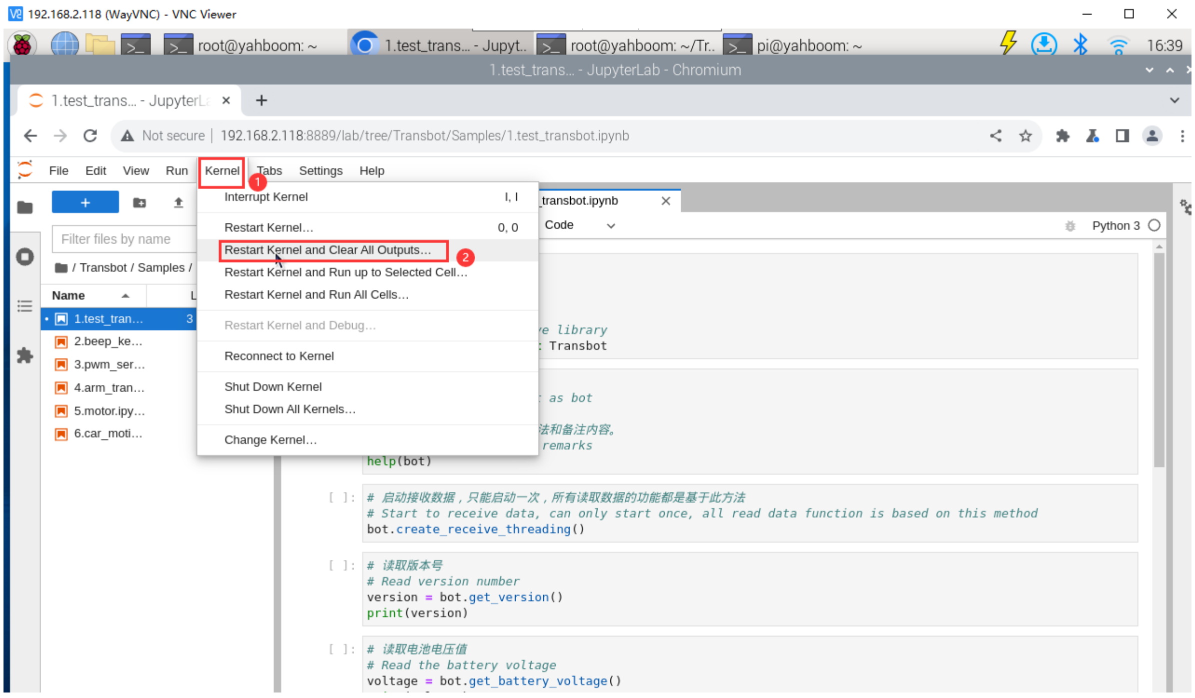
```python
#!/usr/bin/env python3
#coding=utf-8
import time
from Transbot_Lib import Transbot
```

```python
# 创建Transbot对象 bot
# Create Transbot object as bot
bot = Transbot()
```

```python
# 蜂鸣器自动响100毫秒后关闭
# 蜂鸣器开关，on_time=0：关闭，on_time=1：一直响，
# on_time>=10：响xx毫秒后自动关闭（on_time是10的倍数）。
# The buzzer automatically beeps for 100 milliseconds before turning off
# Buzzer switch. On_time =0: the buzzer is off. On_time =1: the buzzer keeps ringing
# On_time >=10: automatically closes after xx milliseconds (on_time is a multiple of 10)
on_time = 100
bot.set_beep(on_time)
# 延迟退出cell
# Delayed exit cell
time.sleep(1)
```

```python
# 蜂鸣器自动响300毫秒后关闭
# The buzzer automatically beeps for 300 milliseconds before turning off
on_time = 300
bot.set_beep(on_time)
time.sleep(1)
```

Restart the kernel and clear all output before running


Kernel
1
Interrupt Kernel I, I
Restart Kernel… 0, 0
Restart Kernel and Clear All Outputs…
2
Restart Kernel and Run up to Selected Cell…
Restart Kernel and Run All Cells…
Restart Kernel and Debug…
Reconnect to Kernel
Shut Down Kernel
Shut Down All Kernels…
Change Kernel…

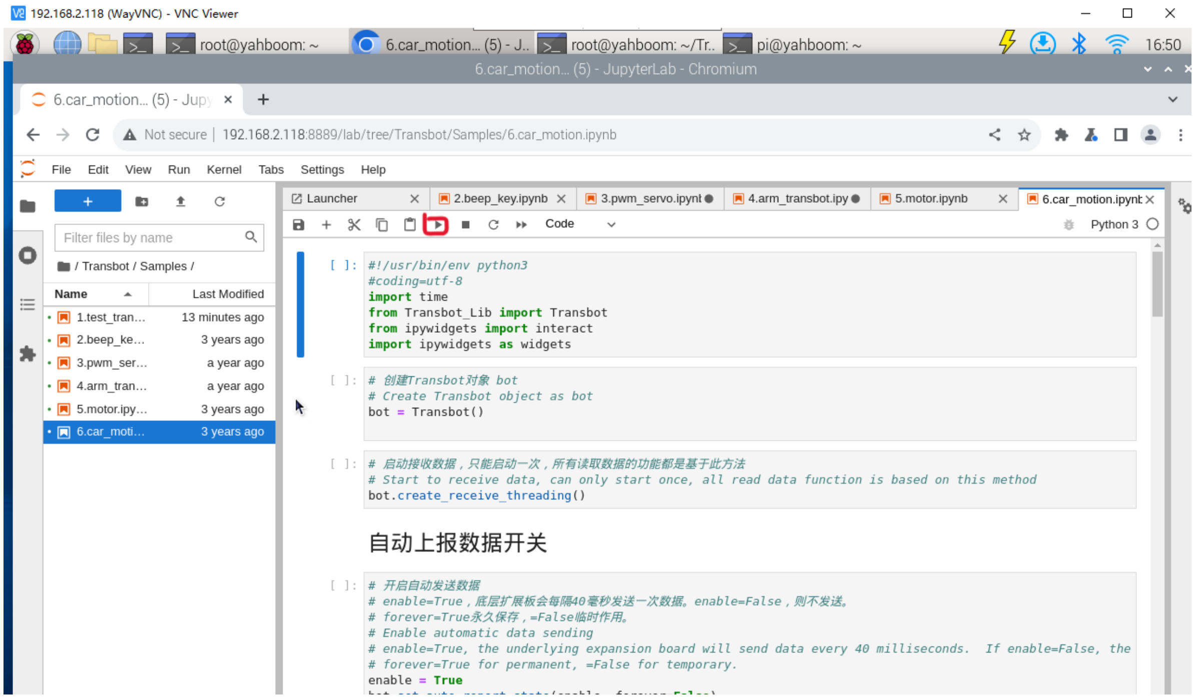
```
help(bot)
```

```
# 启动接收数据，只能启动一次，所有读取数据的功能都是基于此方法
# Start to receive data, can only start once, all read data function is based on this method
bot.create_receive_threading()
```

```
# 读取版本号
# Read version number
version = bot.get_version()
print(version)
```

```
# 读取电池电压值
# Read the battery voltage
voltage = bot.get_battery_voltage()
```

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



```
#!/usr/bin/env python3
#coding=utf-8
import time
from Transbot_Lib import Transbot
from ipywidgets import interact
import ipywidgets as widgets
```

```
# 创建Transbot对象 bot
# Create Transbot object as bot
bot = Transbot()
```

```
# 启动接收数据，只能启动一次，所有读取数据的功能都是基于此方法
# Start to receive data, can only start once, all read data function is based on this method
bot.create_receive_threading()
```

自动上报数据开关

```
# 开启自动发送数据
# enable=True，底层扩展板会每隔40毫秒发送一次数据。enable=False，则不发送。
# forever=True永久保存，=False临时作用。
# Enable automatic data sending
# enable=True, the underlying expansion board will send data every 40 milliseconds.  If enable=False, the
# forever=True for permanent, =False for temporary.
enable = True
```

4. Program source code

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/6.car_motion.ipynb