

# Robotic Navigation and Exploration

## **Jetbot instruction**

Min-Chun Hu [anitahu@cs.nthu.edu.tw](mailto:anitahu@cs.nthu.edu.tw)  
CS, NTHU

# Outline

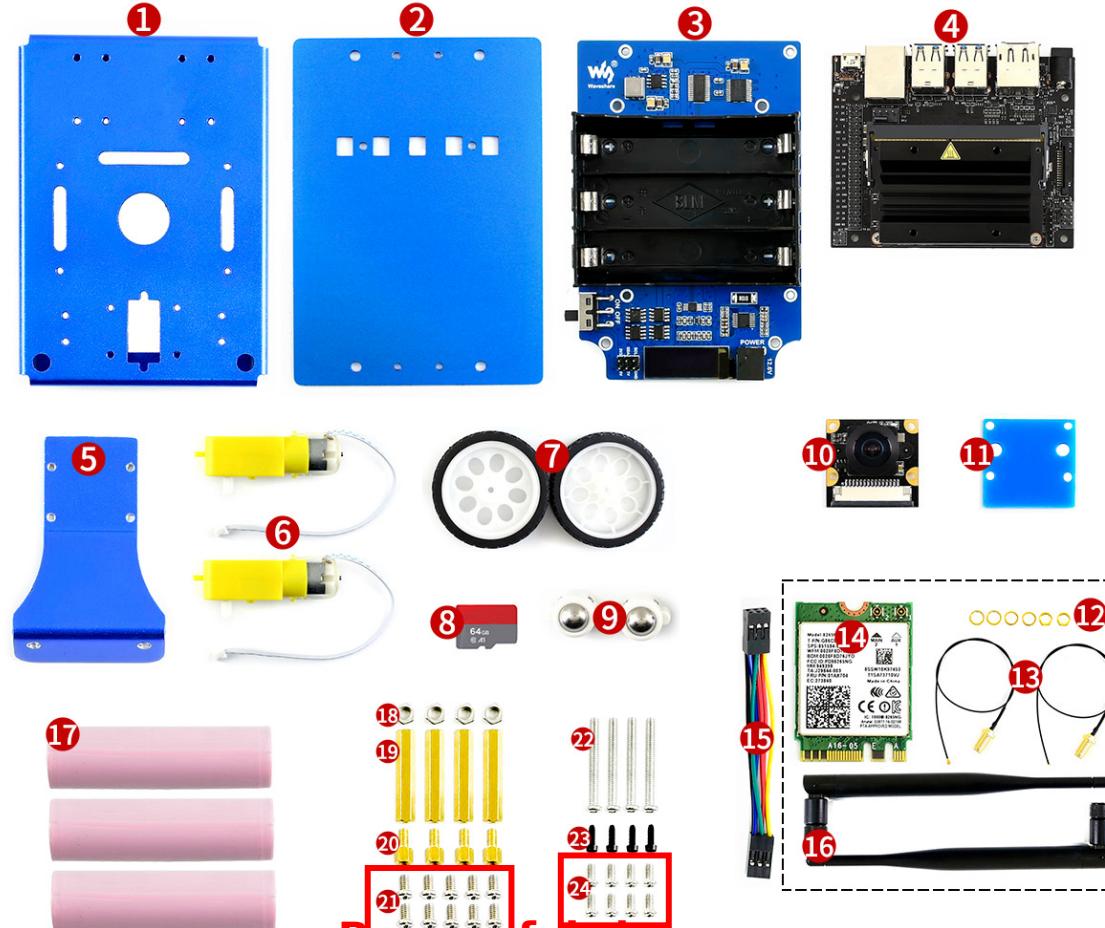
- 1. Hardware
- 2. Software
- 3. Test

# Hardware

# Assemble Manual

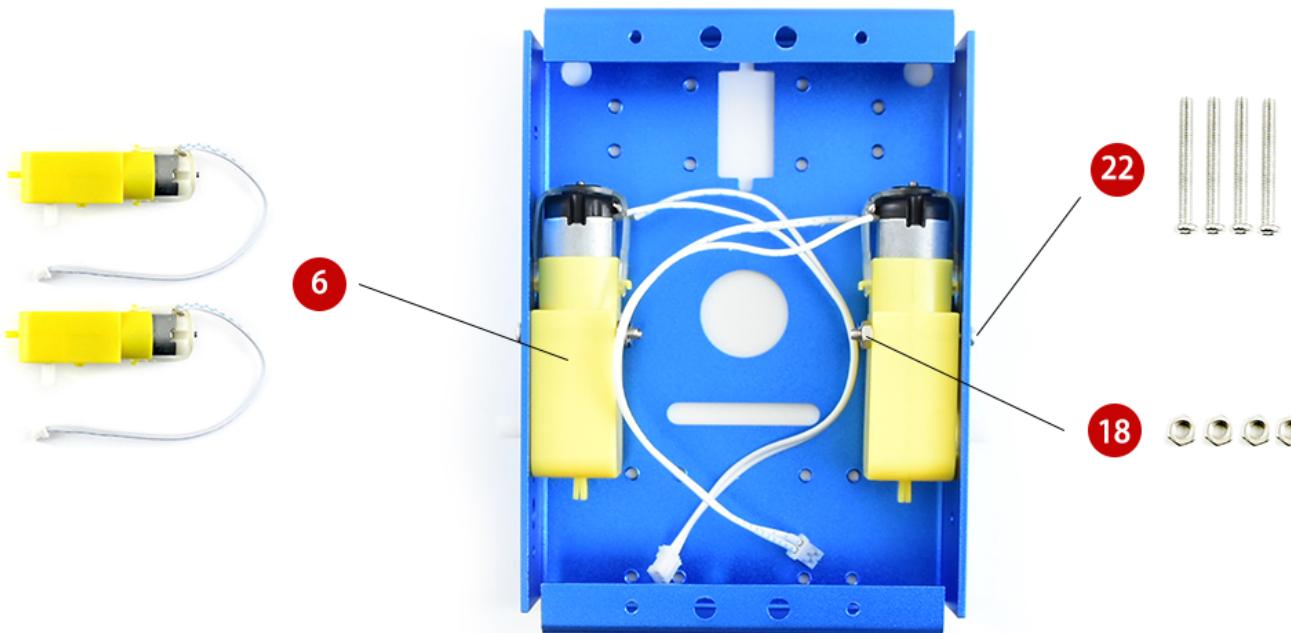
- [JetBot AI Kit Assemble Manual](#)
- Please based on this manual and take slides as reference.

# Components of Jetbot

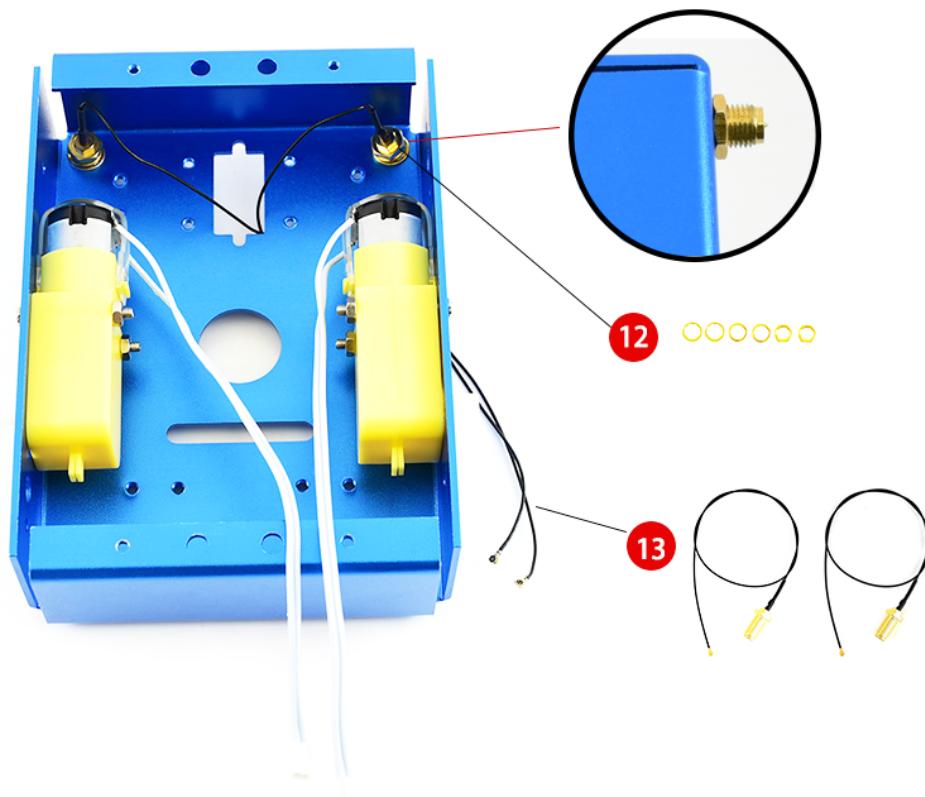


Be careful! these  
two are not the  
same !!

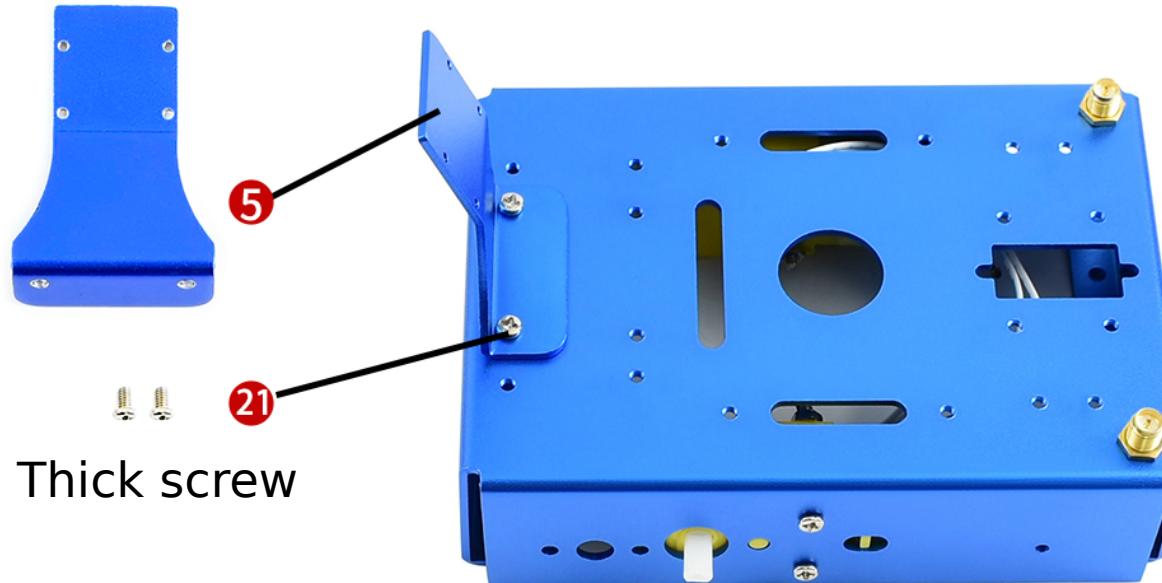
# Step 1: Motor



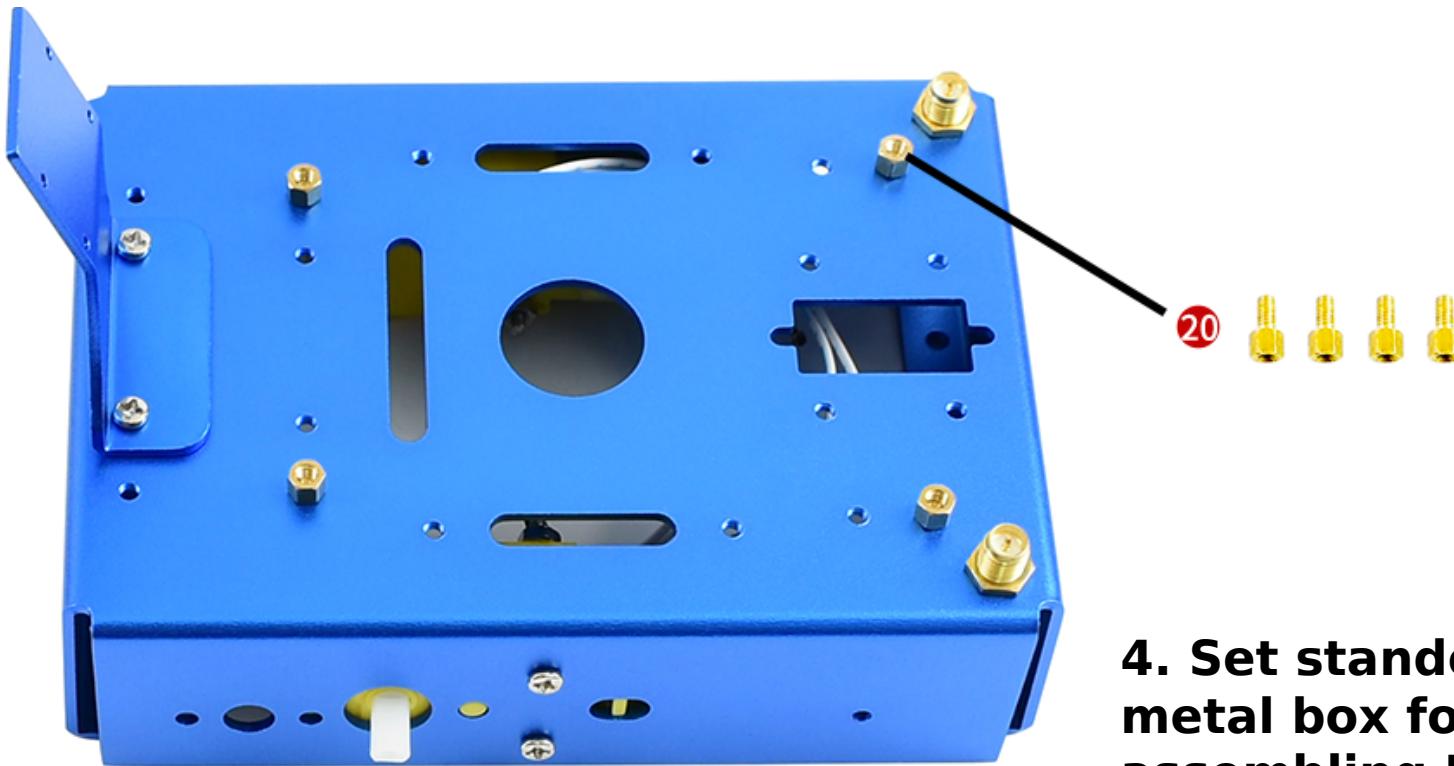
## Step 2: Antenna



## Step 3: Camera Stand

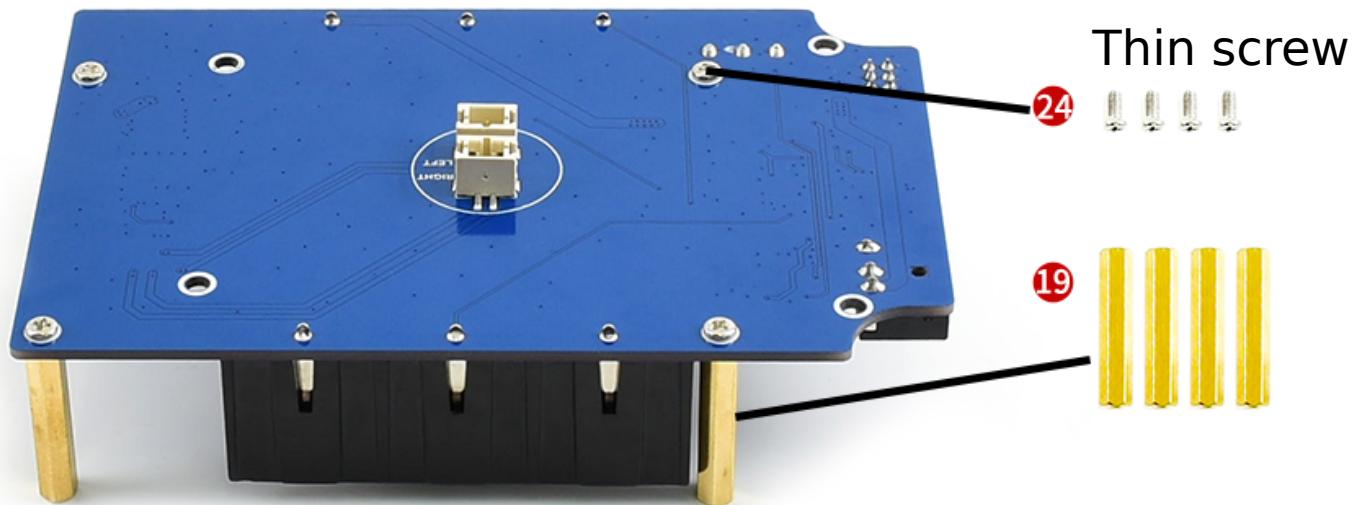


## Step 4: Extend board



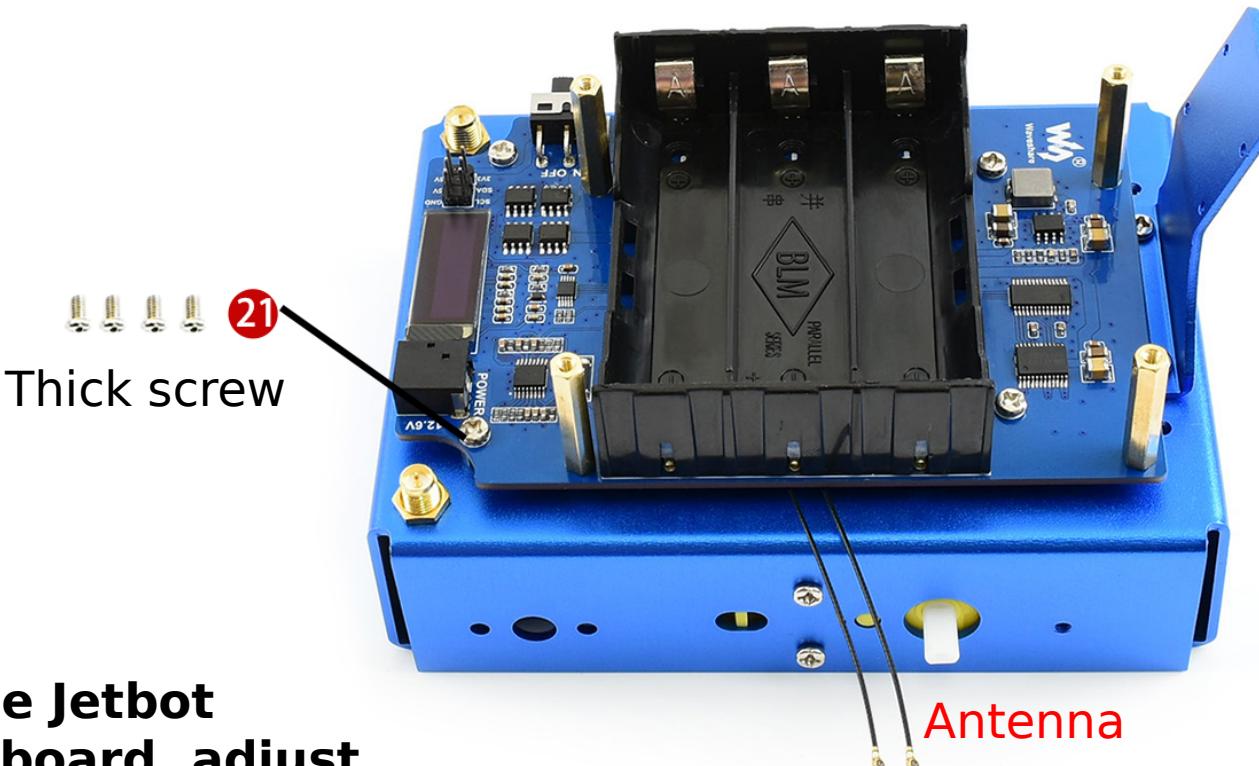
**4. Set standoffs on metal box for assembling JetBot expansion board**

## Step 5. Set standoffs on jetbot

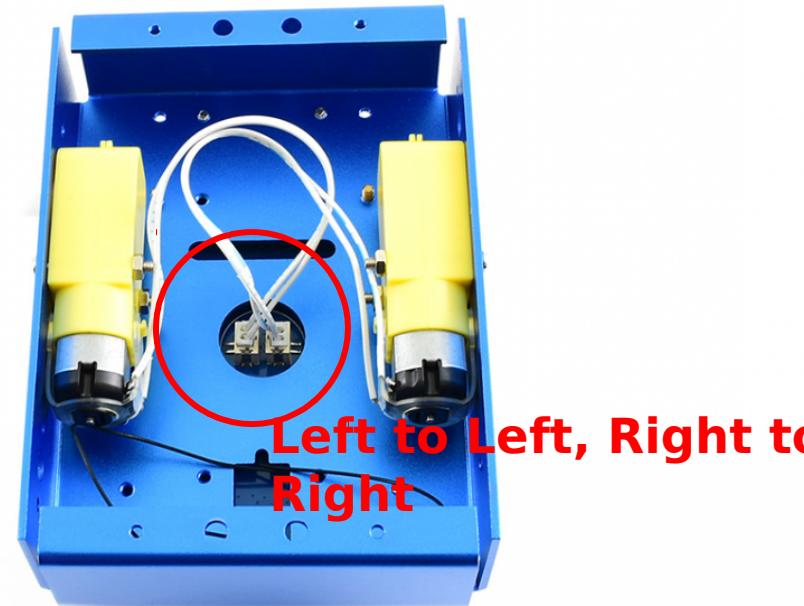


**5. Set standoffs on jetbot expansion board for assembling jetson nano developer kit**

## Step 6. Assemble Jetbot expansion board

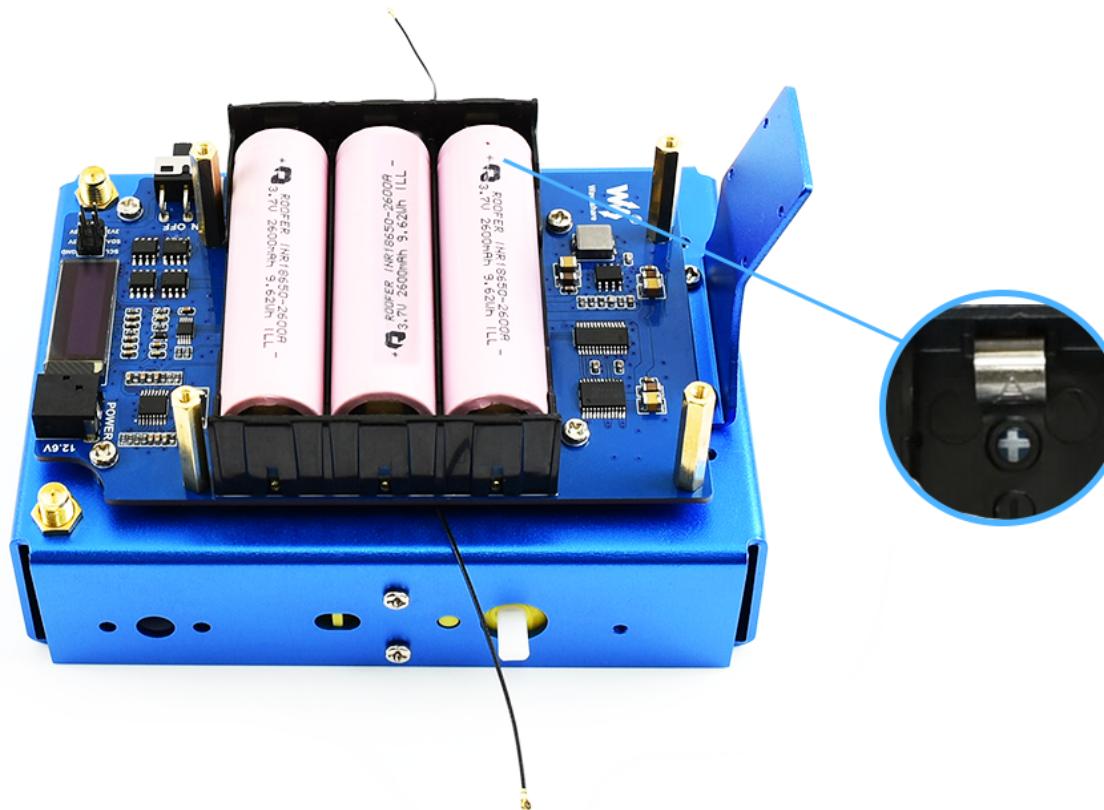


**6. Assemble Jetbot  
expansion board, adjust  
place of antennas**



Connect motors to jetbot expansion board,  
connect left motor to the left interface, and  
right to the right

## Step 7: Install battery

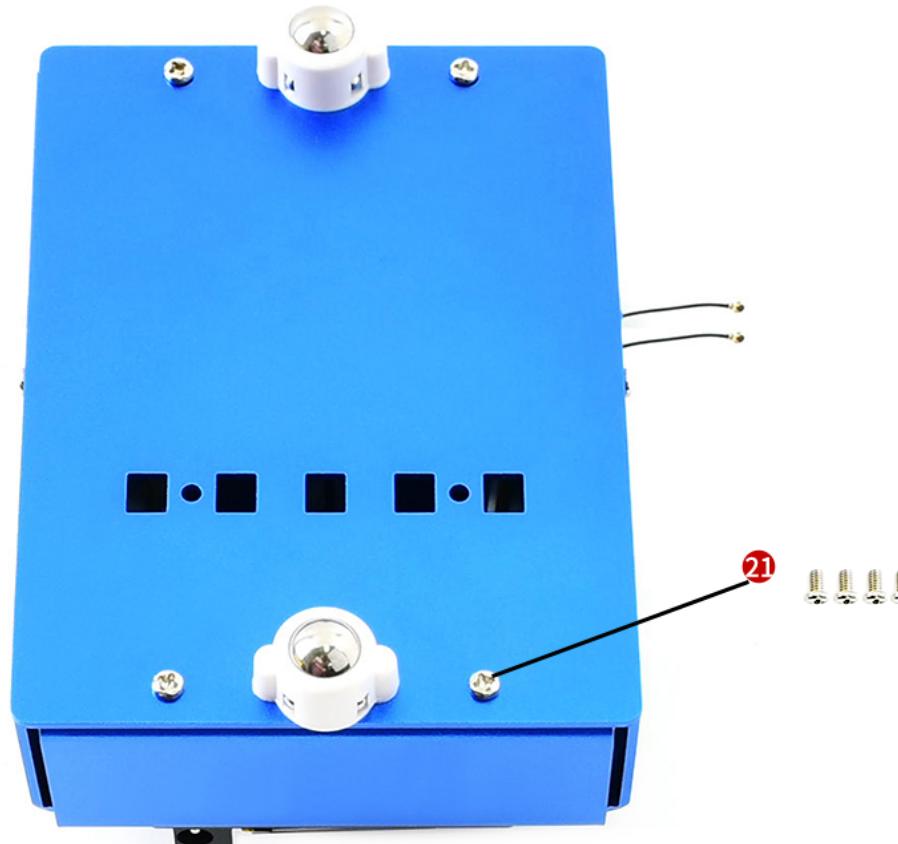


## Step 8: Install universal wheel

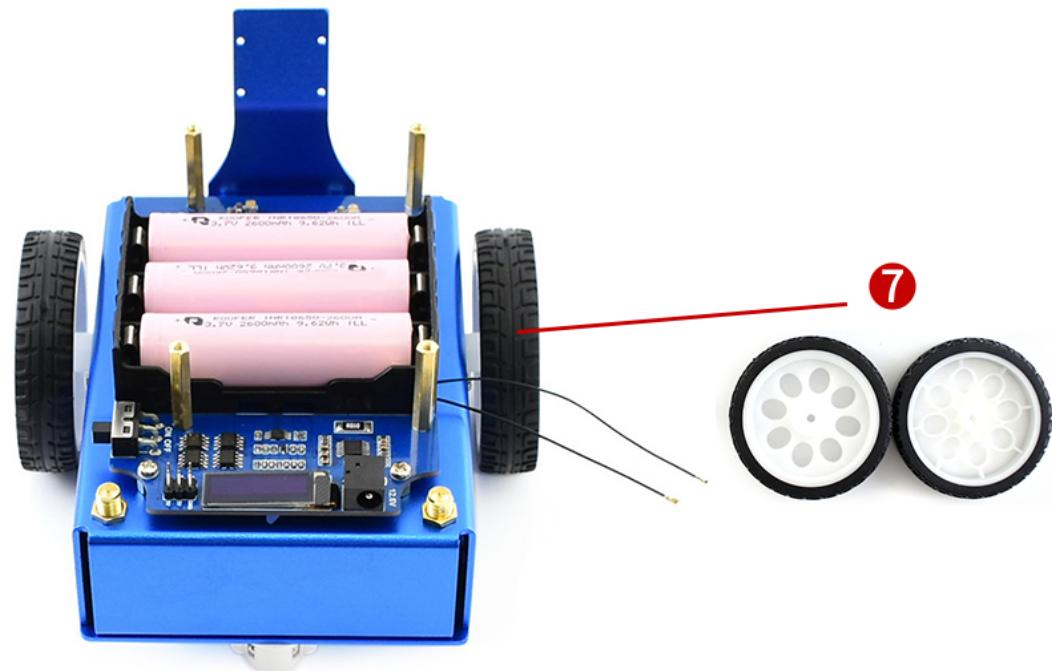


**Turn out screws from Omni-direction wheels, then fixing the wheels to metal bottom board**

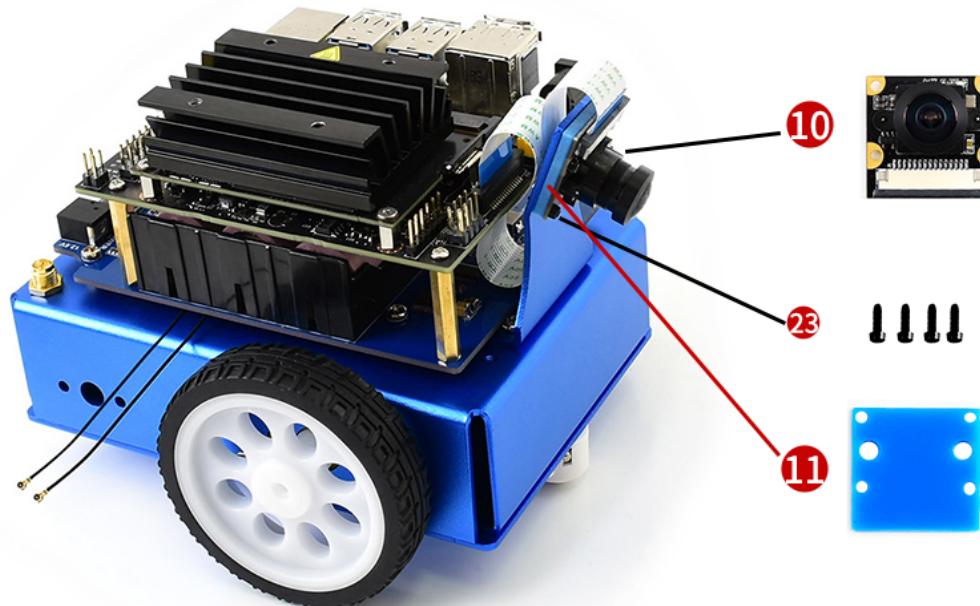
## Step 9: Assemble metal box



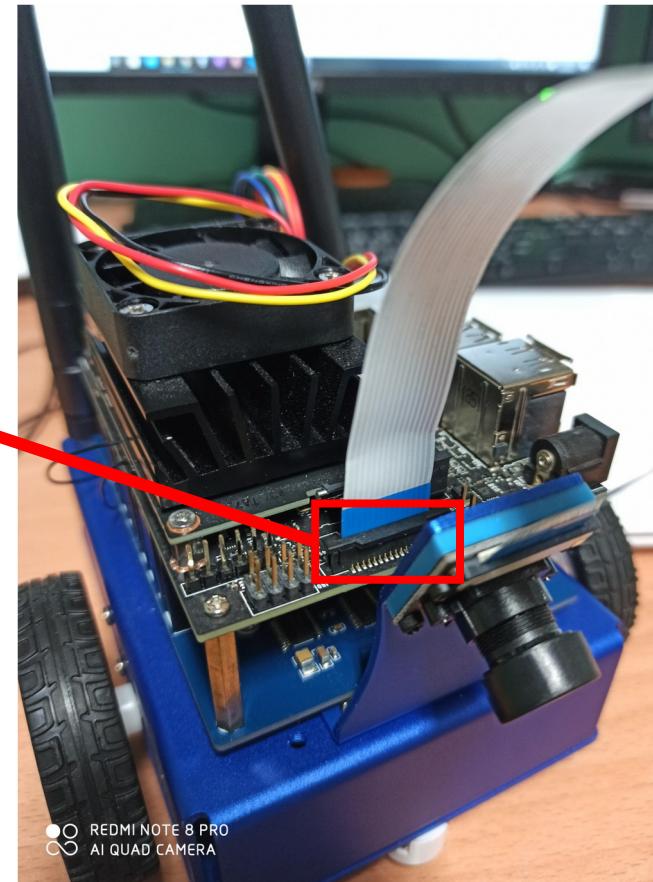
## Step 10: Assemble wheels



## Step 11 : Install camera

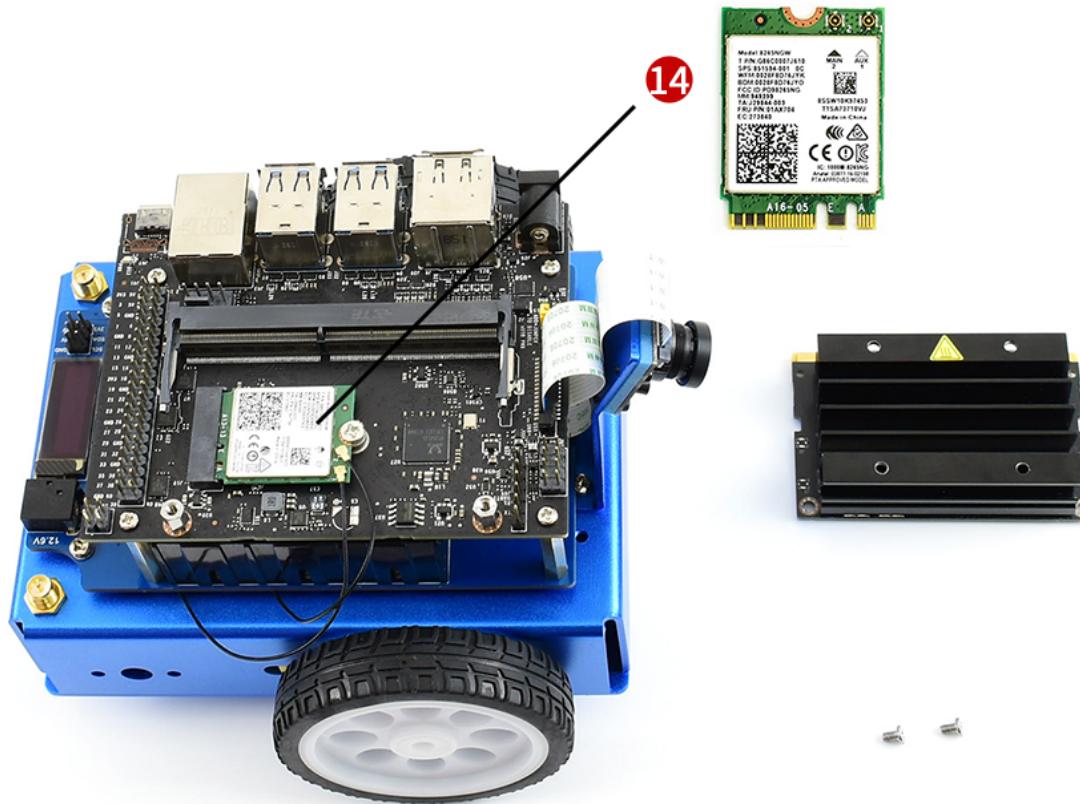


- 1.Pull it up
- 2.Tuck the Camera Cable  
into the slot.



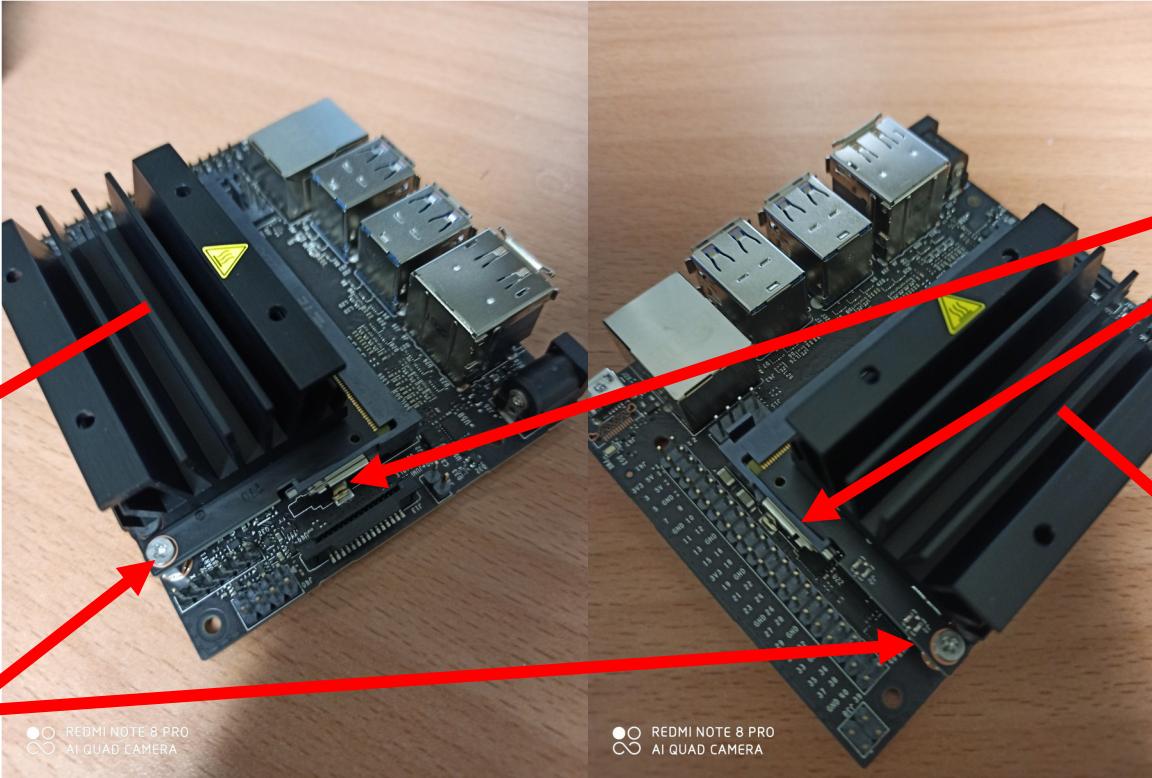
# Step 11: Assemble wireless-AC 8265

Detail on next slide.



**3.Pull it out**

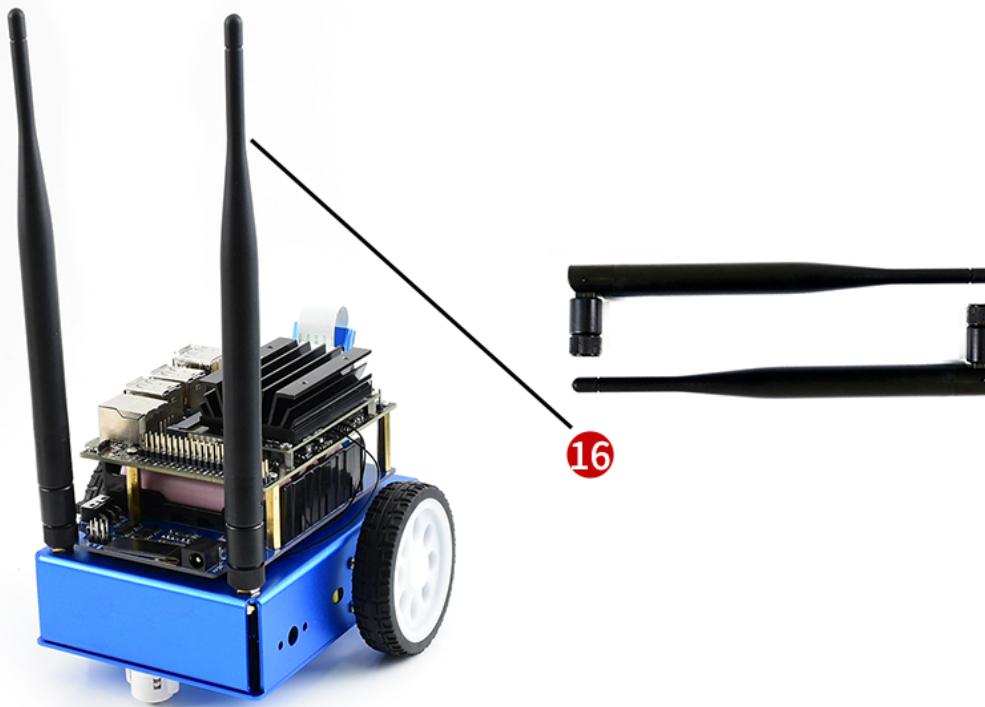
**1. Turn out  
screws both  
side**



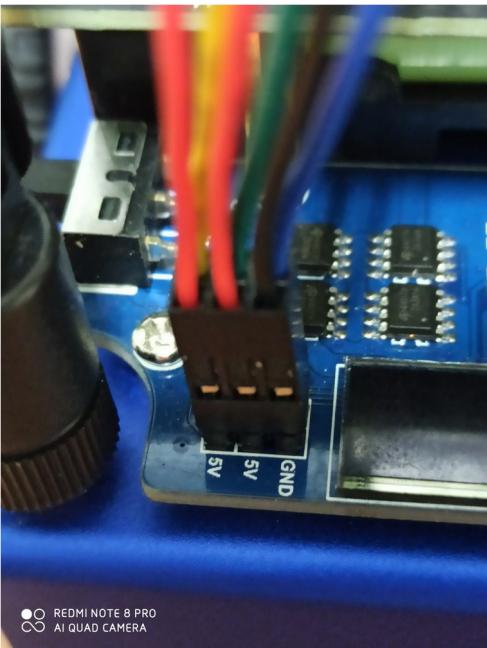
**2. Hold it  
open**

**3. Pull it out**

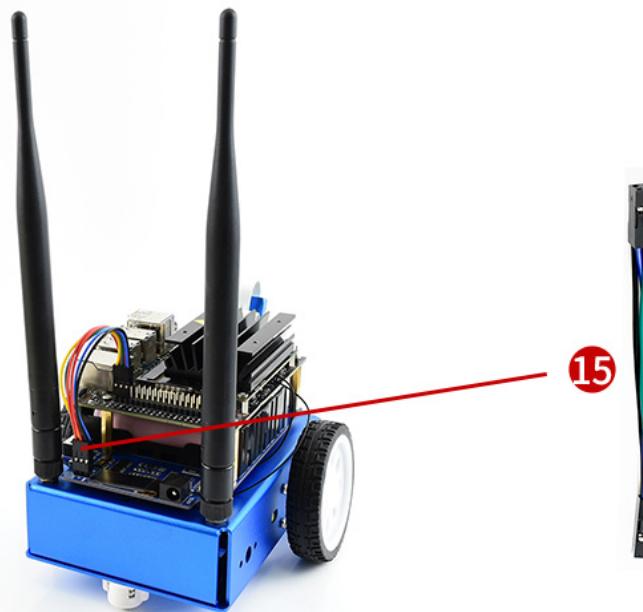
## Step 13: Connect antennas



## Step 14: connect the 8PIN cable



Please be careful on the  
order!



Jetson Nano	Expansion Board
5V	5V
5V	5V
GND	GND
3V3	3V3
3	SDA
5	SCL

# Software

# AI Kit(OS & test reference)

- [JetBot AIKit](#)
- Please download OS [image](#) here.

# Software Setup for JetBot

- **Step 1 - Flash JetBot image onto SD card**
  - Download the expandable JetBot SD card image [jetbot\\_image\\_v0p3p2.zip](#)
  - The above expandable image should work on any SD card greater than or equal to 32GB. If you run into issues please try the archived [63GB](#) images.
- Insert an SD card into your desktop machine
- Using [Etcher](#), select the [jetbot\\_image\\_v0p3p2.zip](#) image and flash it onto the SD card
- Remove the SD card from your desktop machine

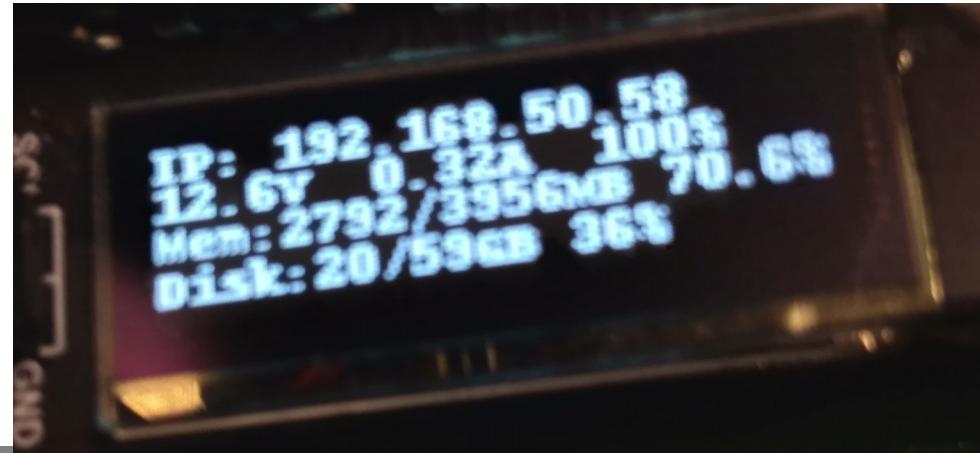
# Software Setup for JetBot

- **Step 2. Startup Jetson Nano Developer Kit**
- Connect HDMI display, keyboard and Mouse to Jetson Nano Developer Kit.

# Software Setup for JetBot

- **Step 3. Connect Jetbot to WIFI**
- All the examples use WIFI, we need to connect JetBot to WIFI firstly.
  - Start Jetson nano Developer Kit, default user name and password of Jetbot are both **jetbot**
  - Click Network icon on top-right of Desktop and connect WIFI
  - Power off. Then assemble Jetbot
  - Start Jetson nano again. After booting, Ubuntu will auto-connect WIFI, IP address is also displayed on OLED

**Jetbot** screen



# Software Setup for JetBot

- **Step 4. Access JetBot via Web**
  - After networking. You can remove peripherals and power adapter.
  - Turn Power switch of Jetbot into On
  - After booting, IP address of OLED can be displayed on OLED
  - Navigate to [http://<jetbot\\_ip\\_address>:8888](http://<jetbot_ip_address>:8888) from your desktop's web browser

# Software Setup for JetBot

- **Step 5. Install the latest software**
- The JetBot GitHub repository may contain software that is newer than that pre-installed on the SD card image. To install the latest software:
  - Access Jetbot by going to `http://<jetbot_ip_address>:8888`
  - Launch a new terminal. Default user name and password are both **jetbot**

```
git clone https://github.com/waveshare/jetbot
cd jetbot
sudo python3 setup.py install
cd
sudo apt-get install rsync
rsync -r jetbot/notebooks ~/Notebooks
```

# Software Setup for JetBot

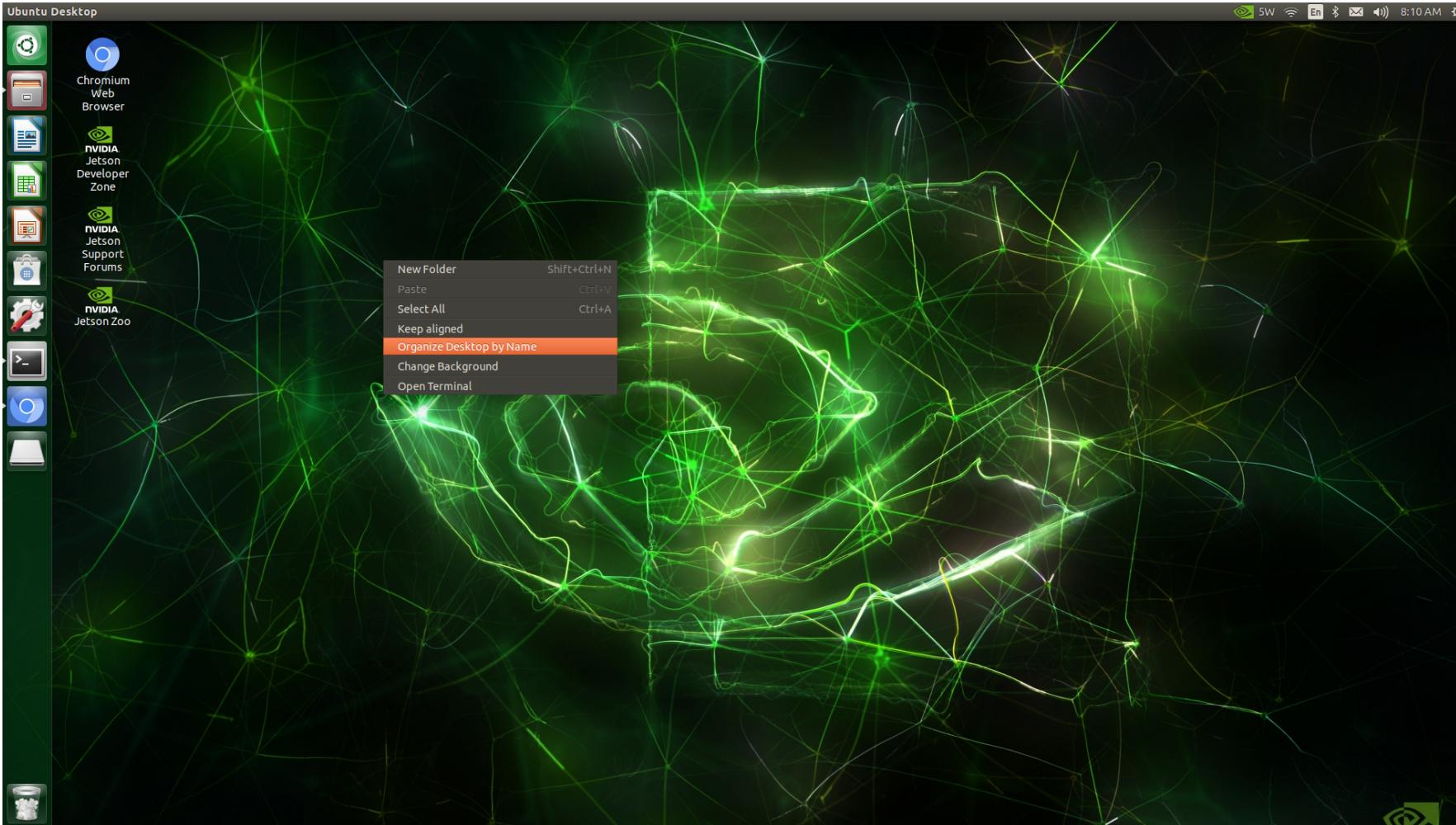
- **Step 6. Configure power mode**
- To ensure that the Jetson Nano doesn't draw more current than the battery pack can supply, place the Jetson Nano in 5W mode by calling the following command
- You need to launch a new Terminal and enter following commands to select 5W power mode

```
sudo nvpmode1 -m1
```

- Check if mode is correct

```
sudo nvpmode1 -q
```

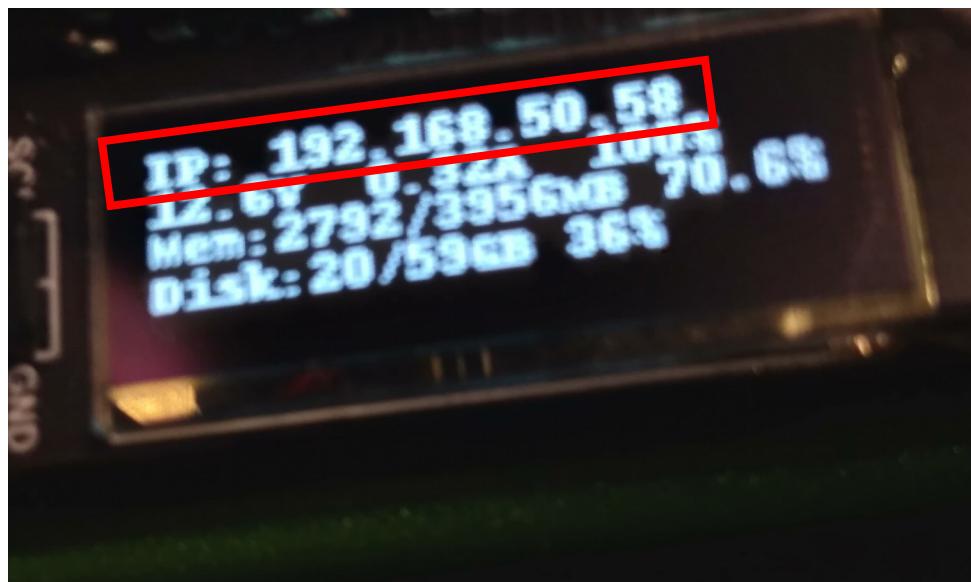
# After install successfully



# Test

# Test for JetBot

- Access jetbot by going to **http://<jetbot\_ip\_address>:8888**, navigate to **~/Notebooks/basic\_motion/**
- Open **basic\_motion.ipynb** file and following the notebook.



# basic\_motion.ipynb

The screenshot shows a Jupyter Notebook interface with the title "basic\_motion.ipynb". The left sidebar displays a file tree with a single item: "basic\_motion.ipynb" (Last Modified 12 days ago). The main content area is titled "Basic Motion". It includes a welcome message, an introduction to controlling JetBot, and sections on importing the Robot class and commanding the robot. The notebook contains several code cells, some of which show TensorFlow deprecation warnings. A warning is present about moving the robot, and a reminder to check wiring. The bottom status bar indicates "No Kernel! | Dead".

Welcome to JetBot's browser based programming interface! This document is called a *Jupyter Notebook*, which combines text, code, and graphic display all in one! Pretty neat, huh? If you're unfamiliar with *Jupyter* we suggest clicking the [Help](#) drop down menu in the top toolbar. This has useful references for programming with *Jupyter*.

In this notebook, we'll cover the basics of controlling JetBot.

### Importing the Robot class

To get started programming JetBot, we'll need to import the `Robot` class. This class allows us to easily control the robot's motors! This is contained in the `jetbot` package.

If you're new to Python, a *package* is essentially a folder containing code files. These code files are called *modules*.

To import the `Robot` class, highlight the cell below and press `ctrl + enter` or the `play` icon above. This will execute the code contained in the cell

```
[1]: from jetbot import Robot
```

```
WARNING:tensorflow:From /usr/lib/python3.6/dist-packages/graphsurgeon/_utils.py:2: The name tf.NodeDef is deprecated. Please use tf.compat.v1.NodeDef instead.
```

```
WARNING:tensorflow:From /usr/lib/python3.6/dist-packages/graphsurgeon/DynamicGraph.py:4: The name tf.GraphDef is deprecated. Please use tf.compat.v1.GraphDef instead.
```

Now that we've imported the `Robot` class we can initialize the class *instance* as follows.

```
[2]: robot = Robot()
```

### Commanding the robot

Now that we've created our `Robot` instance we named "robot", we can use this instance to control the robot. To make the robot spin counterclockwise at 30% of its max speed we can call the following

WARNING: This next command will make the robot move! Please make sure the robot has clearance.

```
[38]: robot.right(speed=0.3)
```

Cool, you should see the robot spin counterclockwise!

If your robot didn't turn left, that means one of the motors is wired backwards! Try powering down your robot and swapping the terminals that the `red` and `black` cables of the incorrect motor.

REMINDER: Always be careful to check your wiring, and don't change the wiring on a running system!

Now, to stop the robot you can call the `stop` method.

Mode: Command No Kernel! | Dead

Ln 1, Col 1 basic\_motion.ipynb

# Test 1. Turn around

## Commanding the robot

Now that we've created our `Robot` instance we named "robot", we can use this instance to control the robot. To m:

WARNING: This next command will make the robot move! Please make sure the robot has clearance.

```
]: robot.right(speed=0.3)
```

Cool, you should see the robot spin counterclockwise!

If your robot didn't turn left, that means one of the motors is wired backwards! Try powering down your rob

REMINDER: Always be careful to check your wiring, and don't change the wiring on a running system!

Now, to stop the robot you can call the `stop` method.

```
]: robot.stop()
```

Maybe we only want to run the robot for a set period of time. For that, we can use the Python `time` package.

- Example :

[https://drive.google.com/file/d/189kP-VIYnnSmgCOuvkb4Uz\\_819IICYhx/view?fbclid=IwAR0PWQEovM53mjmX0jXtb2TQCPNH3eZpX2unAPoWy9OIZX3q4-pAmPEn94g](https://drive.google.com/file/d/189kP-VIYnnSmgCOuvkb4Uz_819IICYhx/view?fbclid=IwAR0PWQEovM53mjmX0jXtb2TQCPNH3eZpX2unAPoWy9OIZX3q4-pAmPEn94g)

## Test 2. Forward

The `link` function that we created

```
19]: robot.forward(0.3)
      time.sleep(2.0)
      robot.stop()
```

You should see the sliders respond to

- Example :

[https://drive.google.com/file/d/1894LCZIn3Zgbv23I6I2ArT\\_hle5N81A/view?fbclid=IwAR2CUzhVOqntXG2e9ANtHkJERhfZZCZtDBYVIdcD9FvsydSgWQpUD7b0N64](https://drive.google.com/file/d/1894LCZIn3Zgbv23I6I2ArT_hle5N81A/view?fbclid=IwAR2CUzhVOqntXG2e9ANtHkJERhfZZCZtDBYVIdcD9FvsydSgWQpUD7b0N64)