RK1808 AI Compute Stick User manual

V1.4.1

Table of Contents

1 Overview

TB-RK1808 AI compute stick is Rockchip's Toybrick series. It is equipped with Rockchip's RK1808 neural network processor. It has low power consumption and high performance, and can be applied to various application fields of artificial intelligence. . The Host can obtain powerful deep learning and inference ability through the RK1808 AI compute stick. With the powerful computing power of the RK1808 AI compute stick, embedded devices can build AI algorithms on the edge of the network, making it easy for traditional embedded devices to upgrade artificial intelligence.

RK1808 AI compute stick can be used to assist inference calculation, and also supports independent AI compute function through secondary development.

2 Specification

Specification	
CPU	RK1808
Memory	1GB LPDDR
Storage	8GB EMMC
Interface	USB3.0　Type-A
Temperature	0℃~40℃
Size	82x31x13mm

2.1 Mini-system requirement

- Ubuntu 16.04 or Windows 7 x86_64 PC
- CPU intel core i3
- USB 3.0
- 2 GB RAM
- 4 GB Storage

3 Document description

Plug the RK1808 AI compute stick into the USB port of PC, and the usb device will be displayed on the PC. The directory structure and document description of the usb device is as follows:

Directory	Document	Description
doc	Rockchip_RK1808_AI_Compute_Stick_User_manual_EN.pdf	RK1808 AI compute stick user manual.
	Rockchip_RK1808_AI_Compute_Stick_User_manual_CN.pdf	
	Rockchip_RK1808_AI_Compute_Stick_Easy_Start_Demo_Guide_EN.pdf	The guide of Easy start yovov3 demo.
	Rockchip_RK1808_AI_Compute_Stick_Easy_Start_Demo_Guide_CN.pdf	
driver	ntb	USB ntd windows driver.
example	run_demo.bat	Run the batch program for yolov3 sample Windows with one click.
	run_demo.sh	Run the script for the yolov3 example with one click.
tool		Scripts and tools.

For more information, please refer to the official wiki: http://t.rock-chips.com/en/wiki.php?mod=view&pid=21

4 Getting start

This section describes an example of how the RK1808 AI compute stick can run mobilenet_v1 quickly on a Ubuntu 18.04 PC based on python 3.6 using RKNN-Toolkit.

4.1 Environment preparation

- A computer with ubuntu 18.04.
- RK1808 AI compute stick.
- Plug RK1808 AI compute stick into the USB port of PC，and input lsusb command to view the results (The line be marked in red , 2207:1800 is TB-RK1808S0's ID)：

1) Enter the following command:

```
lsusb
```

2) The execution result as shown below:

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 009: ID 2207:1800
```

4.2 RKNN-Toolkit installation

RKNN-Toolkit is a software development kit for users to perform model conversion, inference and performance evaluation base on python interface.

In this section, we will show how to install RKNN-Toolkit in a X86_64 PC with ubuntu 18.04 and python 3.6. perform the following command:

- Install Python3.6

```
sudo apt-get install python3.6
```

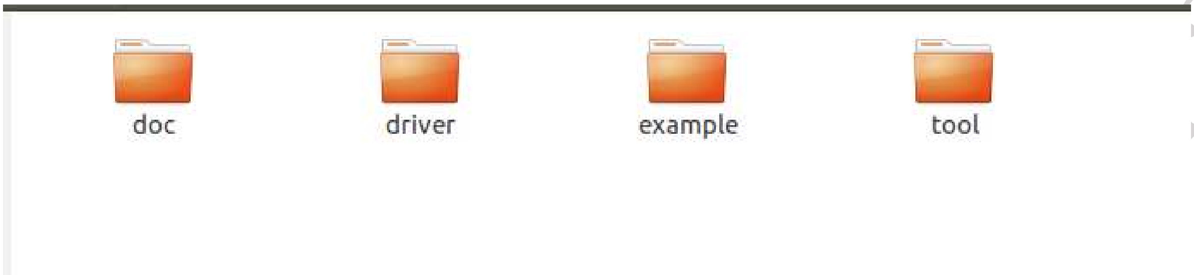
- Install opencv

```
sudo apt-get install -y    python3-opencv
```

- Install pip3

```
sudo apt-get install python3-pip
```

- Plug the RK1808 AI compute stick into the USB port of PC, and the usb device will be displayed on the PC. The root directory structure of the usb device is as follows:



- Perform the following steps：

1. In the current user directory, create an rknn directory and enter the rknn directory:

```
mkdir ~/rknn
cd ~/rknn
```

2. Install wget:

```
sudo apt-get install -y wget
```

3. Download RKNN-Toolkit from the official FTP file server

```
wget http://repo.rock-chips.com/python/rknn_toolkit-1.2.1-cp36-cp36m-linux_x86_64.whl
```

4. Install Python dependencies. The minimum version requirement for tensorflow is 1.11.0.Take installing 1.14.0 as an example:

```
pip3 install --user tensorflow==1.14.0
```

5. Install RKNN-Toolkit：

```
pip3 install --user rknn_toolkit-1.2.1-cp36-cp36m-linux_x86_64.whl
```

Notice: RKNN-Toolkit must match with python version, the default version for ubuntu18.04 is python3.6.

6. Check whether the rknn-toolkit has been successfully installed:

1) Enter the following command:

```
python3
```

2) Enter the following code in python3 running environment to import RKNN module.

```
from rknn.api import RKNN
```

(1) If RKNN has been imported successfully:

```
$ python3
>>> from rknn.api import RKNN
>>>
```

(2) Input quit(), quit from the Python, and Skip the following (3) steps.

(3) If report the following error after import RKNN, it was because the installation package for Tensorflow that RKNN relies on uses the SSE4.2 instruction set and the CPU can’t support those instruction set. So please change a computer which supports SSE4.2 instruction set.

```
$ python3
>>> from rknn.api import RKNN
2019-06-25 20:10:25.255397: F tensorflow/core/platform/cpu_feature_guard.cc:37] The TensorFlow library was compiled to use SSE4.2 instructions, but these aren't available on your machine.
```

4.3 Device access permission modification

1. Plug RK1808 AI compute stick into the USB port of PC, follow these steps to modify the USB device access permissions

   1) Back to U disk root directory, copy “tool/update_rk1808_ai_cs_rule.sh” to “~/rknn” directory. Modify update_rk1808_ai_cs_rule.sh with executable permission.

```
cp tool/update_rk1808_ai_cs_rule.sh ~/rknn -f
chmod +x ~/rknn/update_rk1808_ai_cs_rule.sh
```

   2) Back to the “~/rknn” directory, update_rk1808_ai_cs_rule.sh must be executed with root authority.

```
cd   ~/rknn
sudo ./update_rk1808_ai_cs_rule.sh
```

Notice: This step only need to be executed once during installation and does not need to be executed later.

2. After executing the script, use the lsusb command to view the device number of the TB-1808S0.

   1) Input the following command:

```
lsusb
```

   2) The execution result as shown below:

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 009: ID 2207:0018
```

Notice: 'Bus 003 Device 009', '003' and '009' represent bus and device number,

3. We can check the read and write permissions of the RK1808 AI compute stick.

   1) Input the following command:

```
ls -l /dev/bus/usb/003/009
```

Notice: “003/009”it's not fixed, please get right bus&device number by lsusb command according to your own situation.

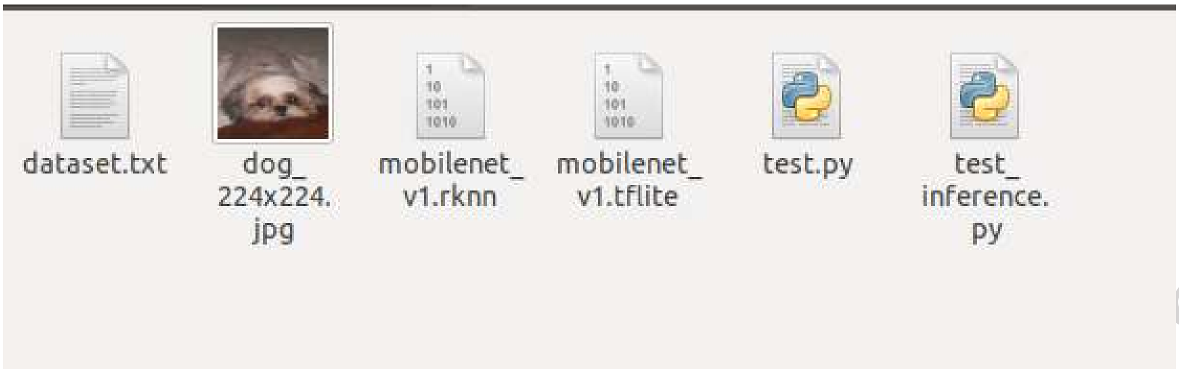
   2) The execution result as shown below (The correct read and write permissions are shown in the red mark):

```
crw-rw-rw- 1 root root 189, 264 6 月  14 16:02 /dev/bus/usb/003/009
```

4.4 Running demo on RK1808 AI compute stick

This section we will take mobilenet_v1 for example, to show how to use RK1808 AI compute stick. Mobilenet_v1 can realize feature extraction of an image and identification of the classification of the image.

The mobilenet_v1 demo directory structure and description are as follow：



- dataset.txt: a text file containing the test image path.
- dog_224x224.jpg: test image.
- mobilenet_v1.tflite: TensorFlow Lite model file.
- mobilenet_v1.rknn: rknn model file. Generated by non-rknn model（there is TensorFlow Lite model）through the rknn-toolkit convert.
- test.py: running script(include rknn model conversion script).
- test_inference.py: running script（only use to load the rknn model for inference）.

Running this demo on RK1808 AI compute stick follow below steps:

1. Download the mobilenet_v1.tar.gz from the official FTP file server and decompress it ：

```
wget http://repo.rock-chips.com/rk1808/mobilenet_v1.tar.gz
tar xvf mobilenet_v1.tar.gz
```

2. Enter the mobilenet_v1 directory and execute the test.py :

```
cd mobilenet_v1/
python3 test.py
```

3. Get the following results:

```
--> config model
done
--> Loading model
done
--> Building model
done
--> Export RKNN model
```

```
done
--> Init runtime environment
done
--> Running model
mobilenet_v1
-----TOP 5-----
[156]: 0.8837890625
[155]: 0.0677490234375
[188 205]: 0.00867462158203125
[188 205]: 0.00867462158203125
[263]: 0.0057525634765625

done
--> Begin evaluate model performance
========================================================================
                              Performance
========================================================================
Total Time(us): 7140
FPS: 140.06
========================================================================

done
```

According to the above execution results, TOP5 represents the results of the top 5 classifications predicted by the model, wherein [156] indicates the dog's label, and 0.8837890625 indicates the probability of predicting the label. It can be seen that the predicted result is the most likely for the dog. From the test picture (dog_224x224.jpg), we can see that the prediction result is accurate.

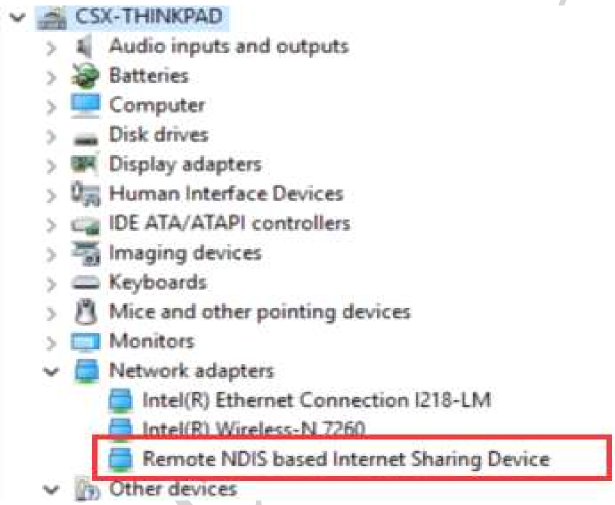
5 WEB configuration

RK1808 AI compute stick supports system configuration by web. How to access this configuration page and its main functions will be described below.

5.1 The Host Network Configuration

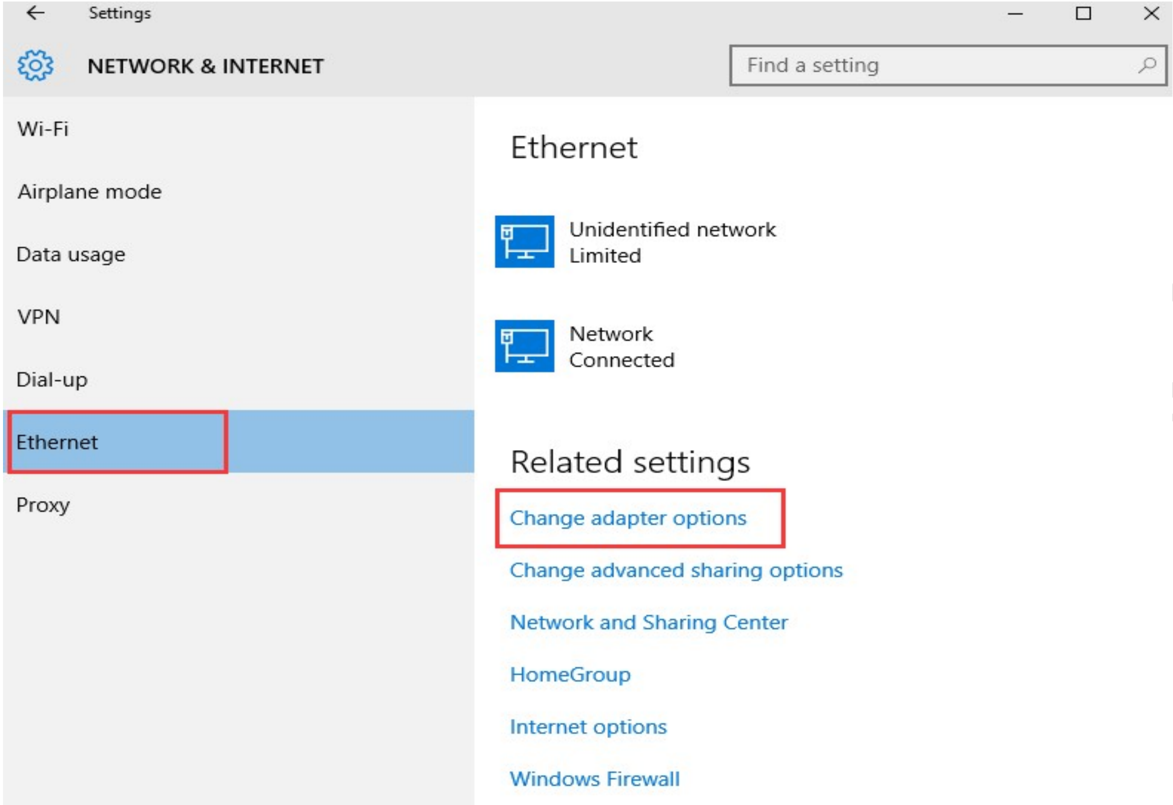
Windows 7/10 network configuration

1. Plug RK1808 AI compute stick。
2. Open the device manager，network adapter will display “Remote NDIS based Internet Sharing Device”.



Notice: Some computer may have unrecognized devices, uninstall unrecognized devices, and wait for the RNDIS driver to reinstall.

3. Click Windows network and Internet Settings，Click “Ethernet” -> “Change adapter options”.



Settings

NETWORK & INTERNET

Find a setting

Wi-Fi
Airplane mode
Data usage
VPN
Dial-up
Ethernet
Proxy

Ethernet

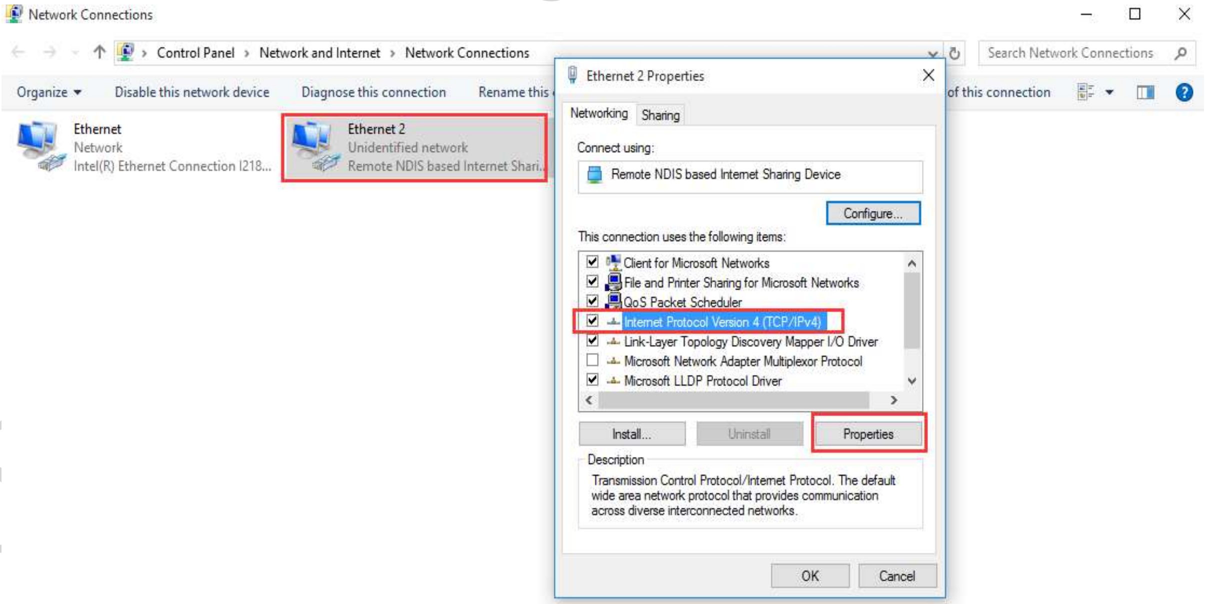
Unidentified network
Limited

Network
Connected

Related settings

Change adapter options
Change advanced sharing options
Network and Sharing Center
HomeGroup
Internet options
Windows Firewall

4. Right click "Ethernet *" (remote NDIS compatible device), -> "attribute" -> "Internet protocol version 4"->"attribute", and configure the IP address and mask as shown below, thus completing the network configuration.



Network Connections

Control Panel › Network and Internet › Network Connections

Search Network Connections

Organize ▾ Disable this network device Diagnose this connection Rename this ... of this connection

Ethernet
Network
Intel(R) Ethernet Connection I218...

Ethernet 2
Unidentified network
Remote NDIS based Internet Shari...

Ethernet 2 Properties

Networking Sharing

Connect using:

Remote NDIS based Internet Sharing Device

Configure...

This connection uses the following items:

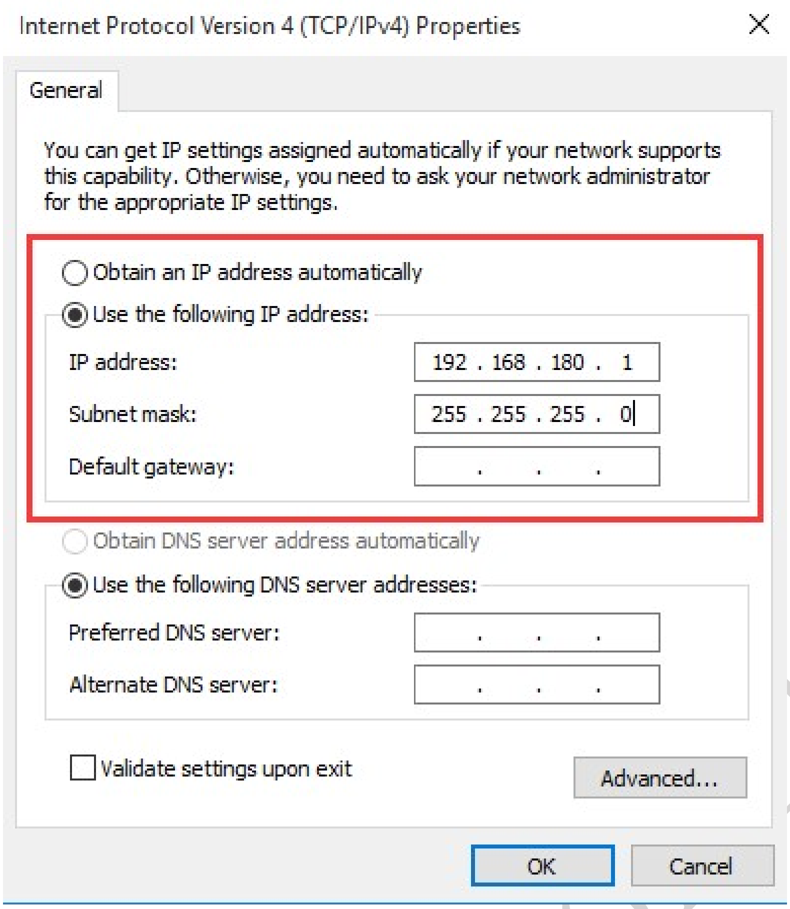
☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ QoS Packet Scheduler
☑ Internet Protocol Version 4 (TCP/IPv4)
☑ Link-Layer Topology Discovery Mapper I/O Driver
☐ Microsoft Network Adapter Multiplexor Protocol
☑ Microsoft LLDP Protocol Driver

Install... Uninstall Properties

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

OK Cancel



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 180 . 1

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

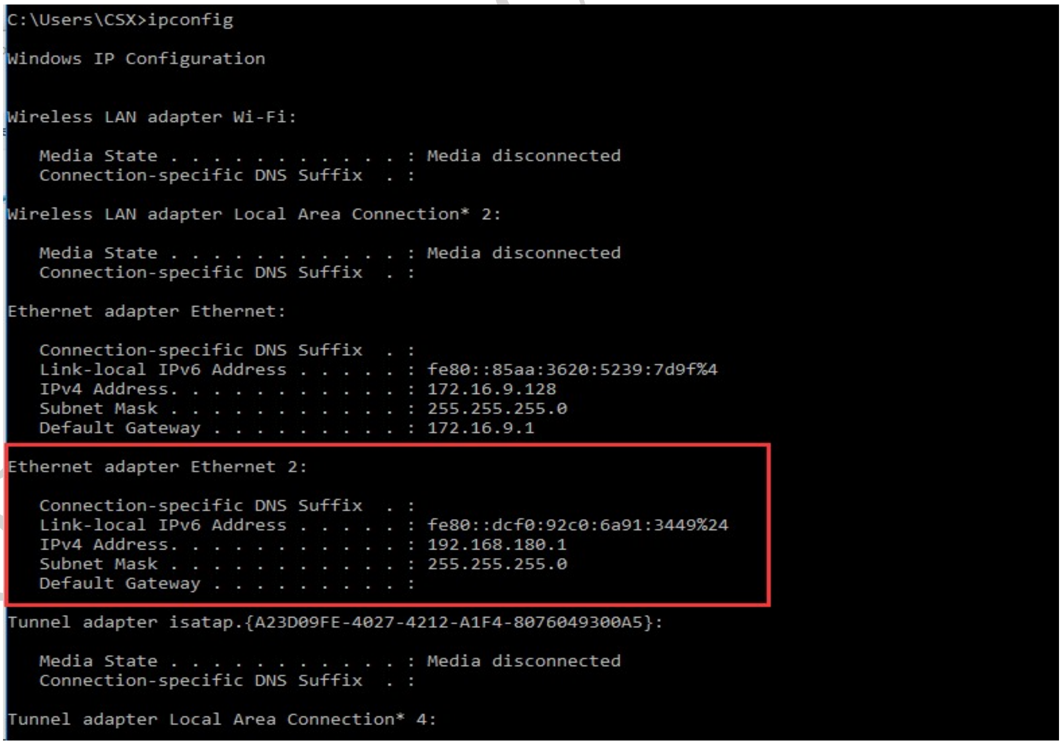
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

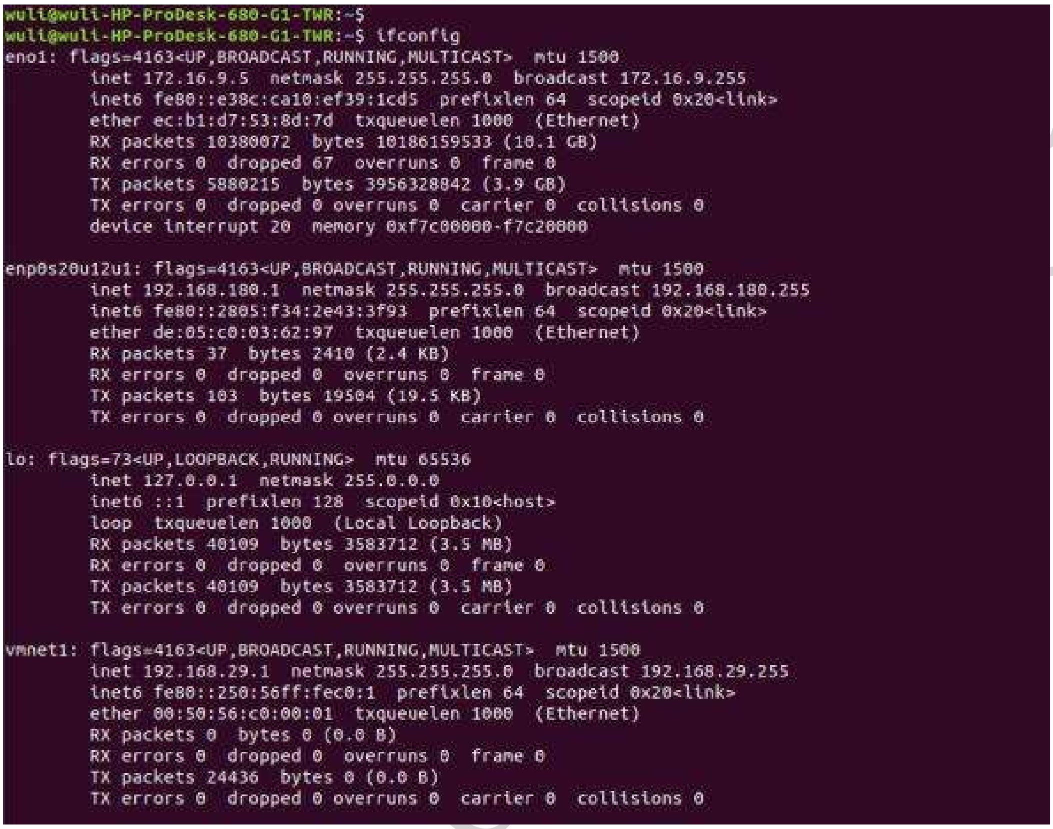
5. Check the IP address.



```
C:\Users\CSX>ipconfig

Windows IP Configuration


Wireless LAN adapter Wi-Fi:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::85aa:3620:5239:7d9f%4
   IPv4 Address. . . . . . . . . . . : 172.16.9.128
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.9.1

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::dcf0:92c0:6a91:3449%24
   IPv4 Address. . . . . . . . . . . : 192.168.180.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Tunnel adapter isatap.{A23D09FE-4027-4212-A1F4-8076049300A5}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 4:
```

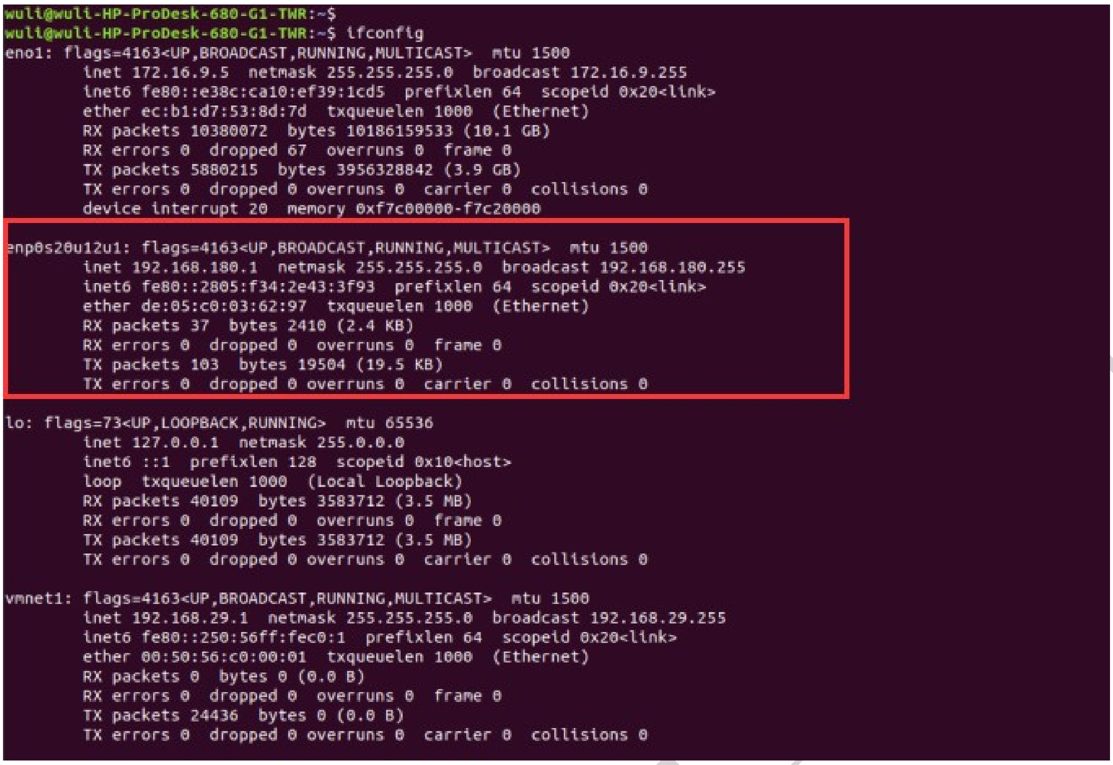
Linux network configuration (Take Ubuntu 18.04 as an example).

1. Before inserting RK1808 AI compute stick, open the terminal and enter the command: ifconfig.



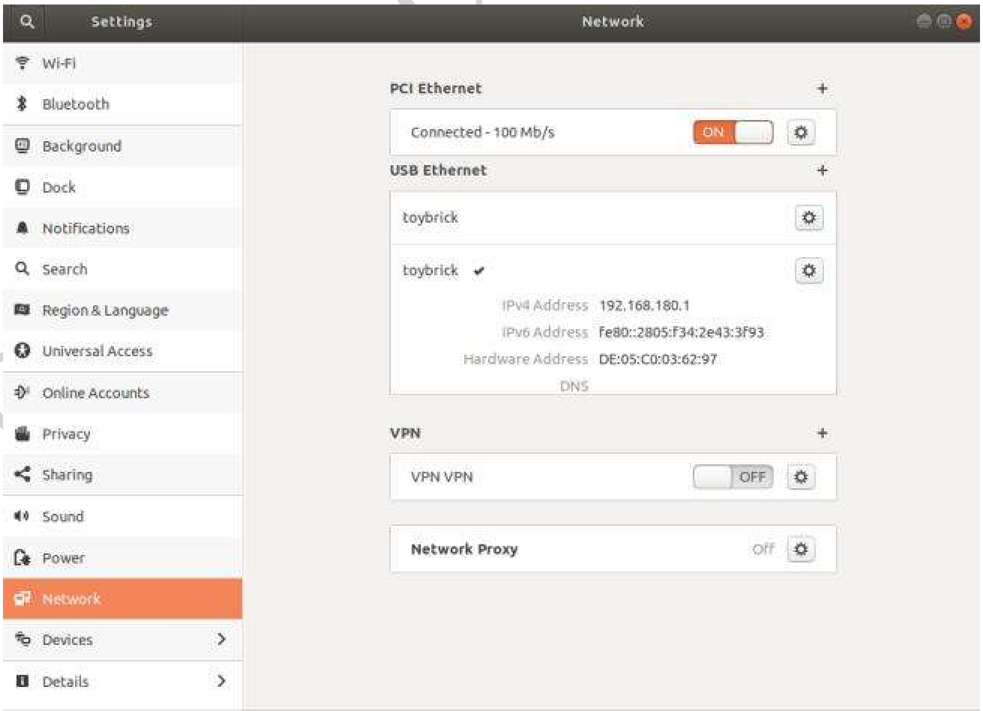
```
wuli@wuli-HP-ProDesk-680-G1-TWR:~$
wuli@wuli-HP-ProDesk-680-G1-TWR:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.9.5  netmask 255.255.255.0  broadcast 172.16.9.255
        inet6 fe80::e38c:ca10:ef39:1cd5  prefixlen 64  scopeid 0x20<link>
        ether ec:b1:d7:53:8d:7d  txqueuelen 1000  (Ethernet)
        RX packets 10380072  bytes 10186159533 (10.1 GB)
        RX errors 0  dropped 67  overruns 0  frame 0
        TX packets 5880215  bytes 3956328842 (3.9 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf7c00000-f7c20000

enp0s20u12u1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.180.1  netmask 255.255.255.0  broadcast 192.168.180.255
        inet6 fe80::2805:f34:2e43:3f93  prefixlen 64  scopeid 0x20<link>
        ether de:05:c0:03:62:97  txqueuelen 1000  (Ethernet)
        RX packets 37  bytes 2410 (2.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 103  bytes 19504 (19.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 40109  bytes 3583712 (3.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40109  bytes 3583712 (3.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vmnet1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.29.1  netmask 255.255.255.0  broadcast 192.168.29.255
        inet6 fe80::250:56ff:fec0:1  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:c0:00:01  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24436  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. Then insert RK1808 AI compute stick and enter the command ifconfig again, you can find an extra with red box in the picture.



```
wuli@wuli-HP-ProDesk-680-G1-TWR:~$
wuli@wuli-HP-ProDesk-680-G1-TWR:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.9.5  netmask 255.255.255.0  broadcast 172.16.9.255
        inet6 fe80::e38c:ca10:ef39:1cd5  prefixlen 64  scopeid 0x20<link>
        ether ec:b1:d7:53:8d:7d  txqueuelen 1000  (Ethernet)
        RX packets 10380072  bytes 10186159533 (10.1 GB)
        RX errors 0  dropped 67  overruns 0  frame 0
        TX packets 5880215  bytes 3956328842 (3.9 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf7c00000-f7c20000

enp0s20u12u1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.180.1  netmask 255.255.255.0  broadcast 192.168.180.255
        inet6 fe80::2805:f34:2e43:3f93  prefixlen 64  scopeid 0x20<link>
        ether de:05:c0:03:62:97  txqueuelen 1000  (Ethernet)
        RX packets 37  bytes 2410 (2.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 103  bytes 19504 (19.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 40109  bytes 3583712 (3.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40109  bytes 3583712 (3.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vmnet1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.29.1  netmask 255.255.255.0  broadcast 192.168.29.255
        inet6 fe80::250:56ff:fec0:1  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:c0:00:01  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24436  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

3. Click the system Settings, select the network, find the new device with the same Mac address in step 2, and click the Settings icon in the upper right corner.



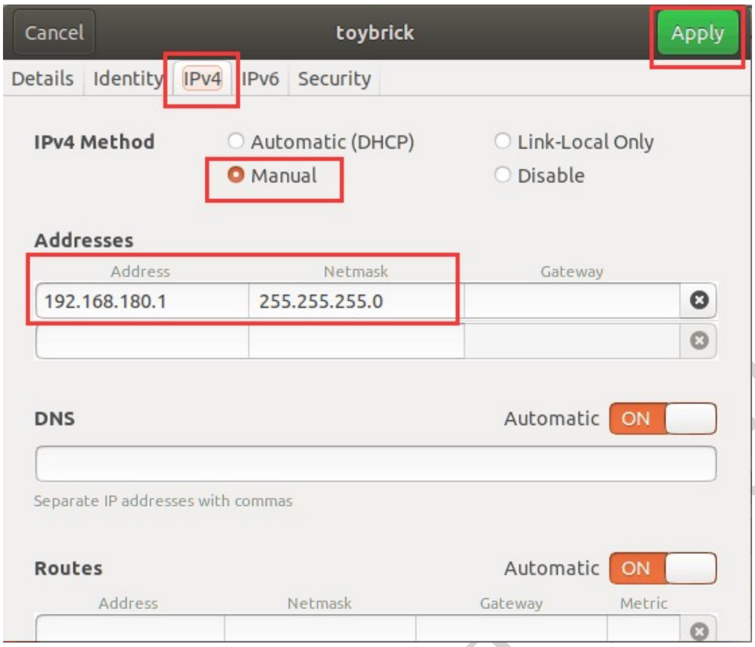
PCI Ethernet
Connected - 100 Mb/s ON

USB Ethernet
toybrick
toybrick ✓
IPv4 Address 192.168.180.1
IPv6 Address fe80::2805:f34:2e43:3f93
Hardware Address DE:05:C0:03:62:97
DNS

VPN
VPN VPN OFF

Network Proxy Off

4. According to the figure below, select ipv4 ,select manually ,add a new IP -- 192.168.180.1 255.255.255.0.Finally, Click apply to complete the network .



IPv4 Method: ○ Automatic (DHCP) ○ Link-Local Only ● Manual ○ Disable

Addresses

Address	Netmask	Gateway
192.168.180.1	255.255.255.0	

DNS Automatic ON

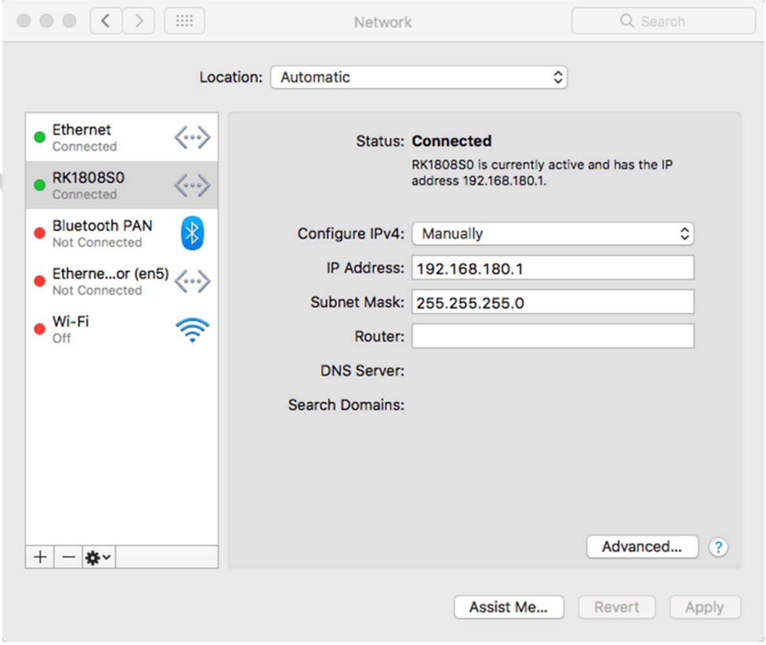
Separate IP addresses with commas

Routes Automatic ON

MAC OS network configuration：

1. Open "System Preference ", and choose " Network ".

2. Choose RK1808S0 tab, and config the ip address.



Location: Automatic

Status: Connected

RK1808S0 is currently active and has the IP address 192.168.180.1.

Configure IPv4: Manually

IP Address: 192.168.180.1

Subnet Mask: 255.255.255.0

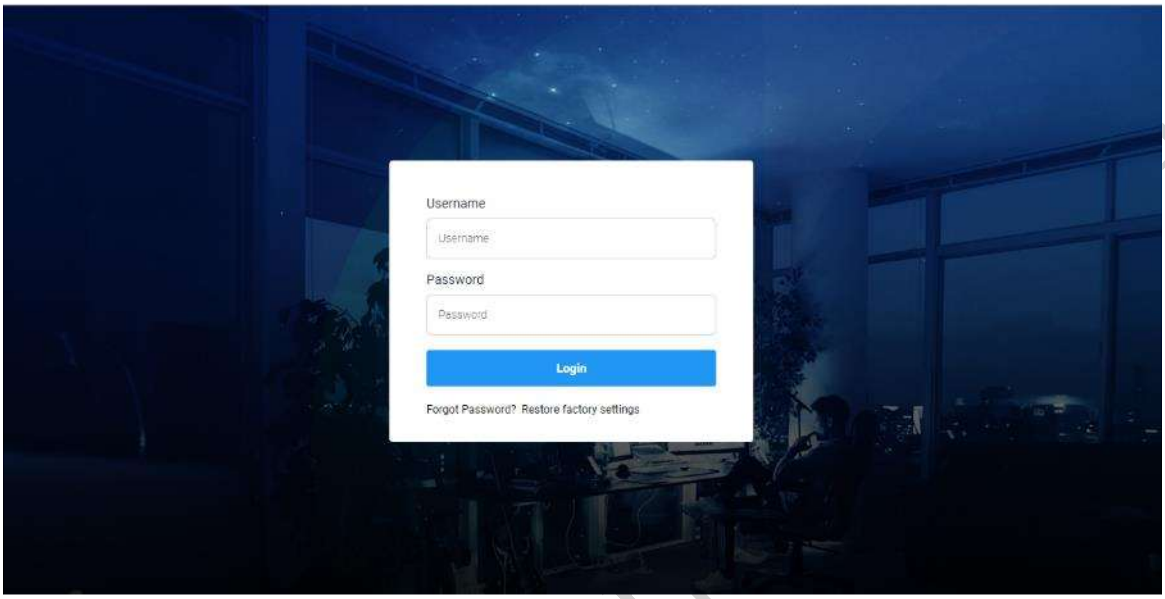
Router:

DNS Server:

Search Domains:

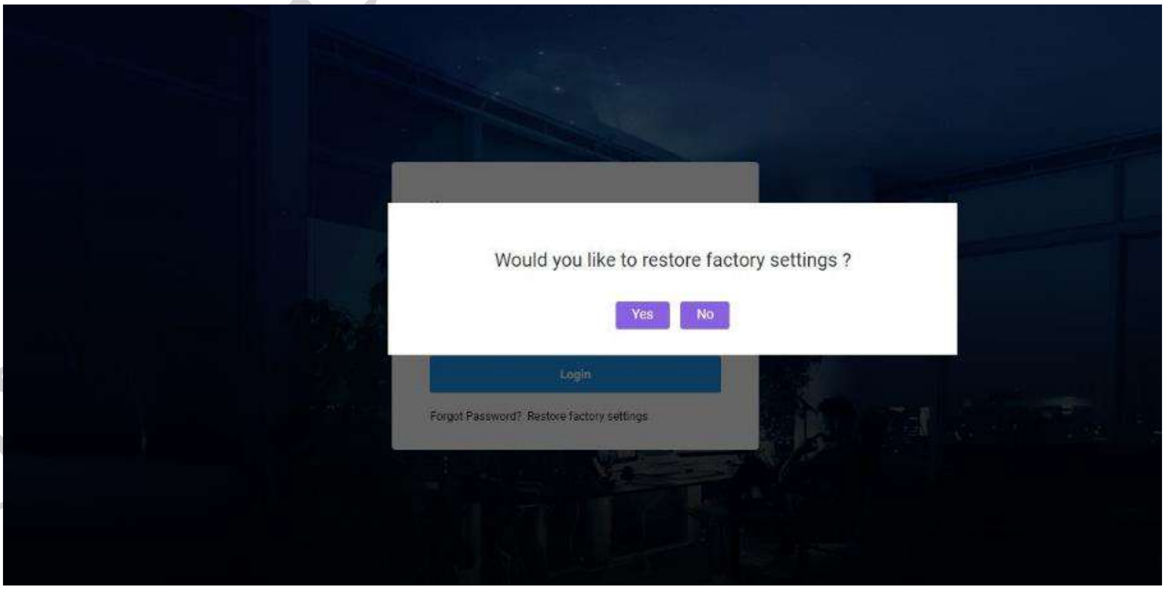
5.2 WEB Login

1. After Network configuration is completed, you can input http://192.168.180.8 in the browser. And then into RK1808 AI compute stick Login page.

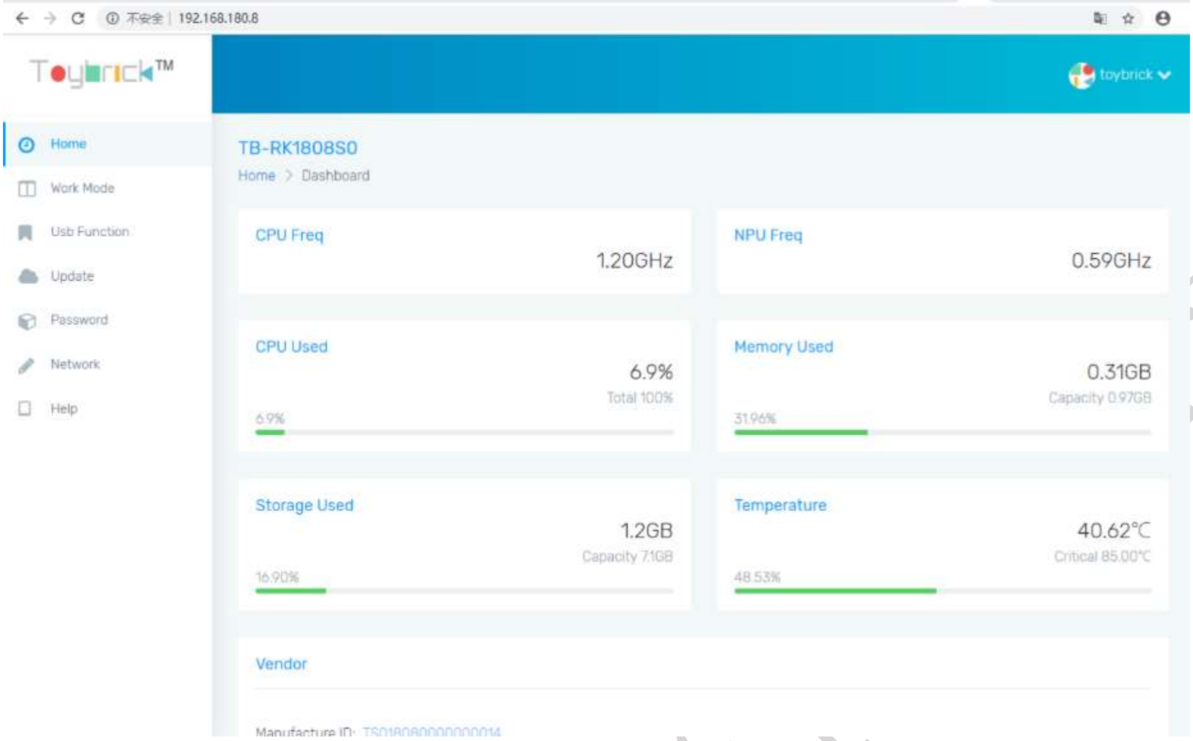


If you forget the password, please click “Forget Password”.

Notice: It will not only reset the password, but also restore the RK1808 AI compute stick to factory Settings. Please be careful


Would you like to restore factory settings ?

2. Enter username and password to login (default username and password are both toybrick), and then enter the Home page.



TB-RK1808S0

Home > Dashboard

CPU Freq
1.20GHz

NPU Freq
0.59GHz

CPU Used
6.9%
Total 100%
6.9%

Memory Used
0.31GB
Capacity 0.97GB
31.96%

Storage Used
1.2GB
Capacity 7.1GB
16.90%

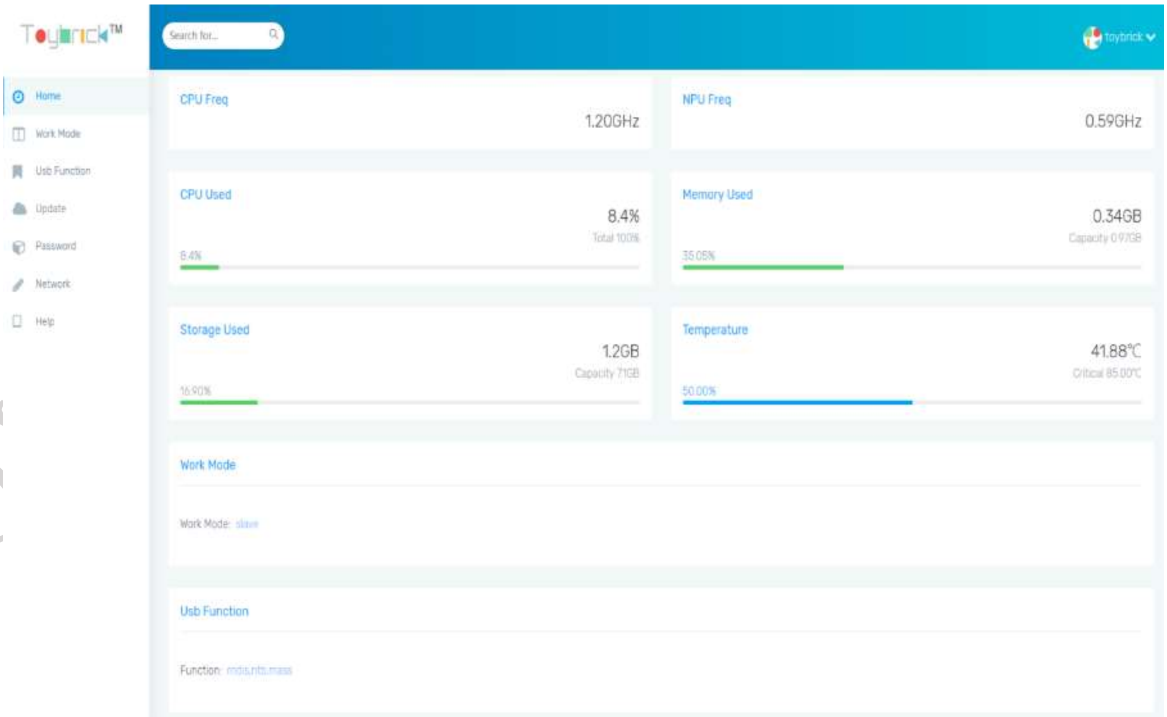
Temperature
40.62℃
Critical 85.00℃
48.53%

Vendor

Manufacture ID: TS01808000000014

5.3 Introduction of Main Functions of WEB Pages

1. Home：Displays the main system information and configuration information for RK1808 AI compute stick.



CPU Freq
1.20GHz

NPU Freq
0.59GHz

CPU Used
8.4%
Total 100%
8.4%

Memory Used
0.34GB
Capacity 0.97GB
35.05%

Storage Used
1.2GB
Capacity 7.1GB
16.90%

Temperature
41.88℃
Critical 85.00℃
50.00%

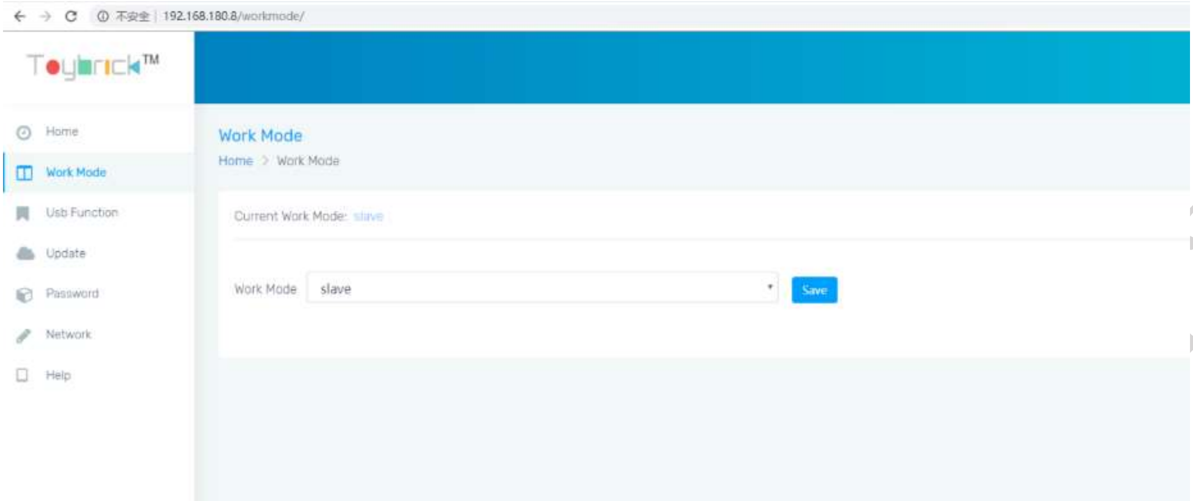
Work Mode

Work Mode: slave

Usb Function

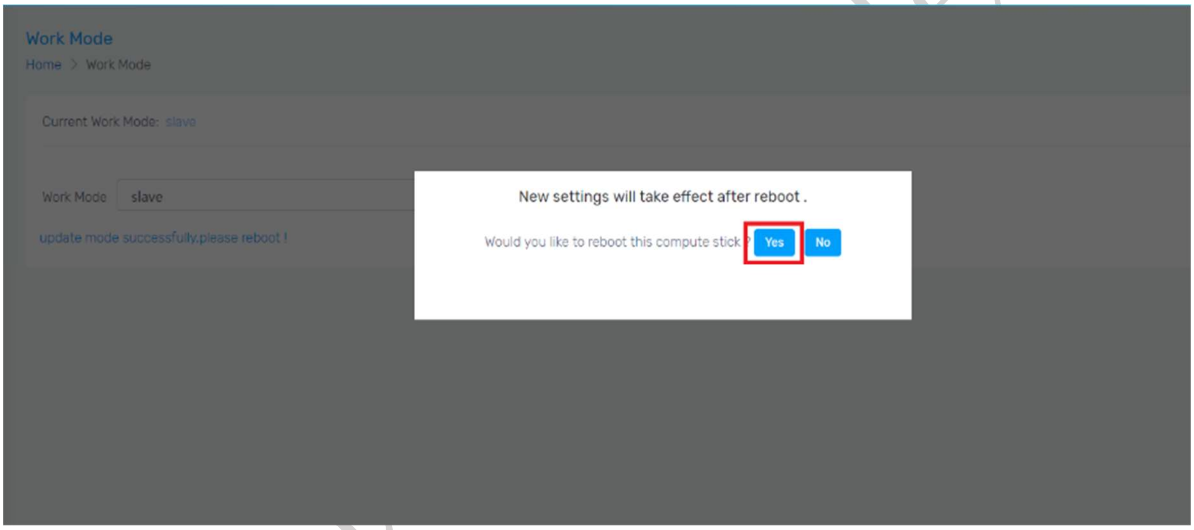
Function: rndis,ntb,mass

2. Work Mode: The working mode is master or slave.



Current Work Mode: slave

Work Mode: slave



Current Work Mode: slave

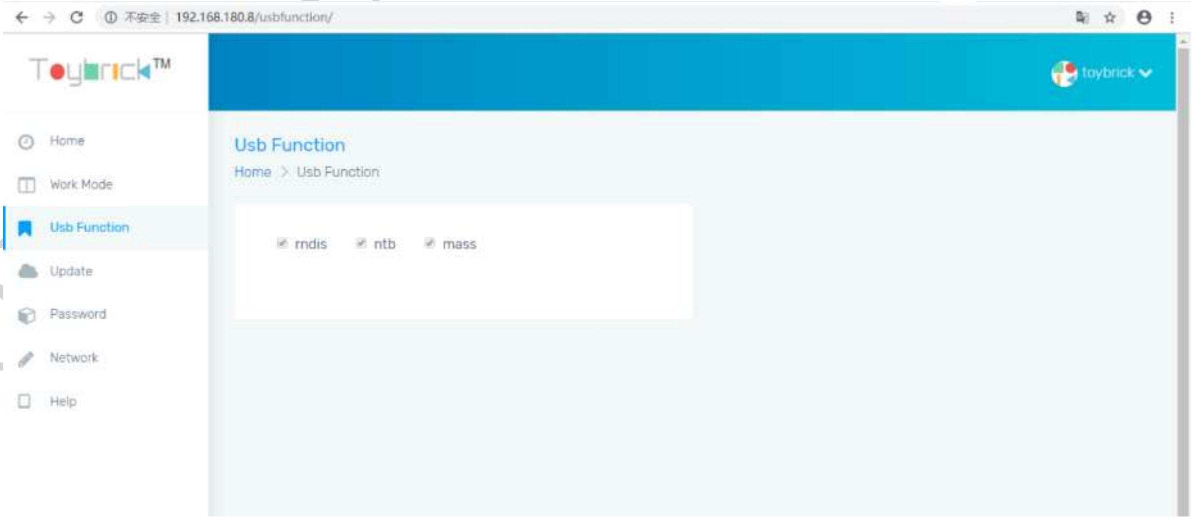
Work Mode: slave

update mode successfully,please reboot !

New settings will take effect after reboot .

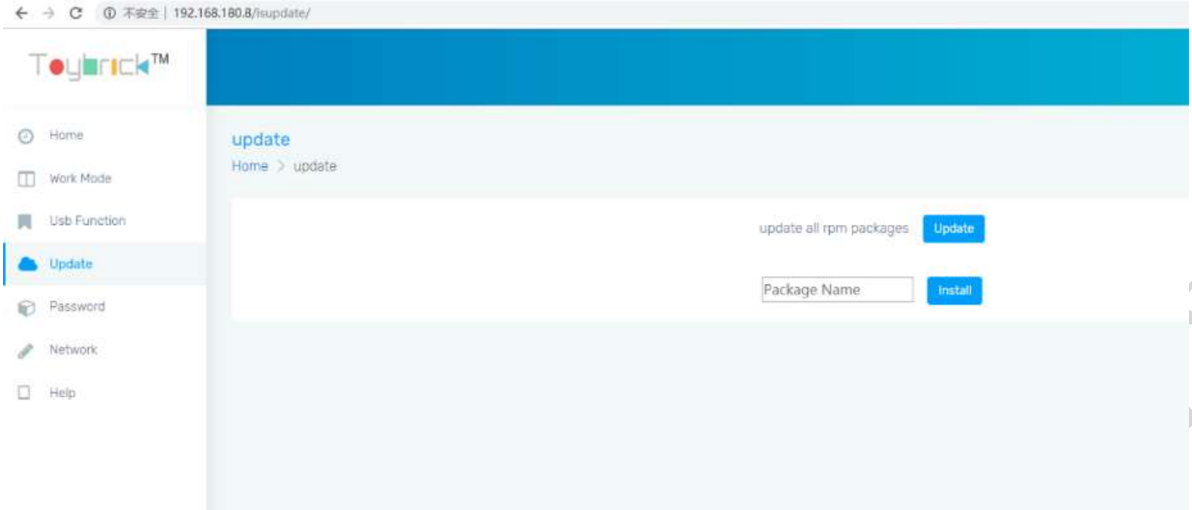
Would you like to reboot this compute stick?

3. USB Function: RK1808 AI compute stick Usbfunction(rndis/ntb/mass)。

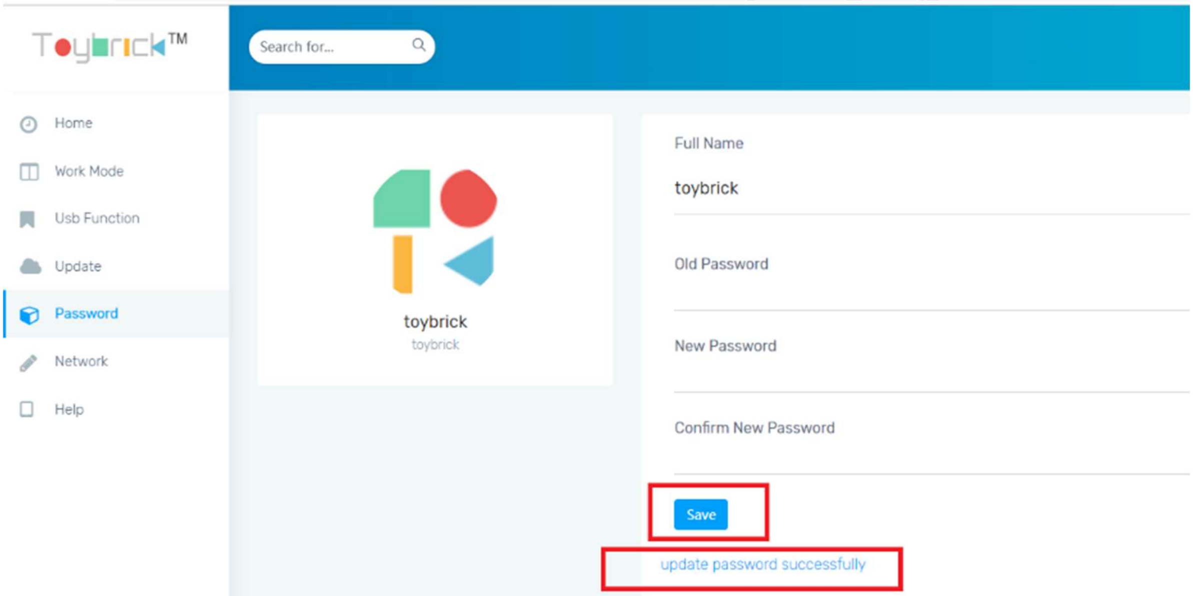


☑ rndis ☑ ntb ☑ mass

4. Update: Update,Install RK1808 AI compute stick RPM resources.

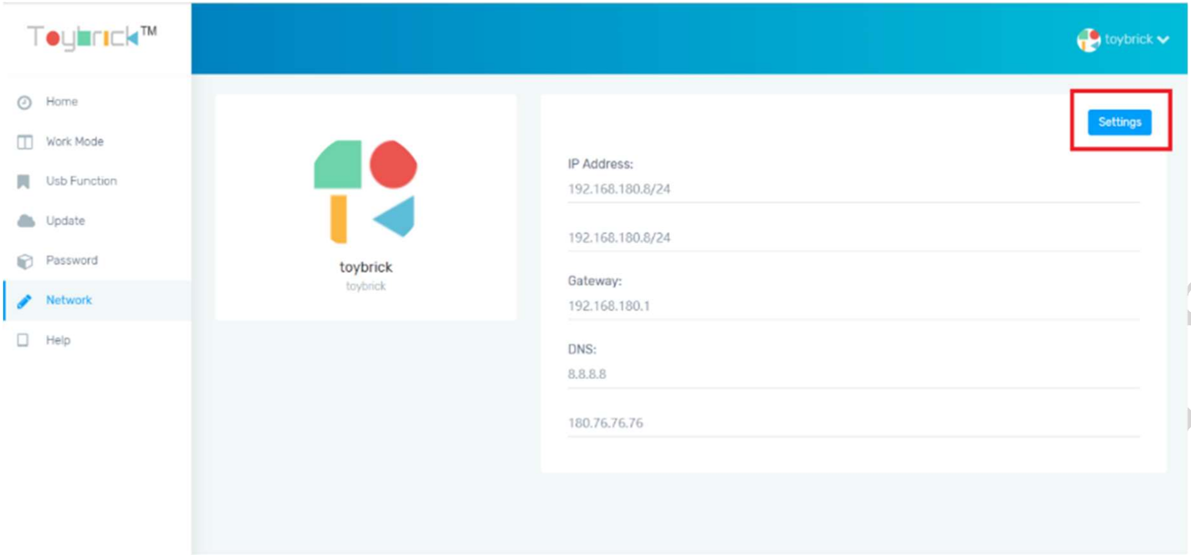


5. Password: You can modify the login password here. After the modification, you need to login again. The blue word in the figure indicates that the modification is successful.



update password successfully

6. Network: RK1808 AI compute stick network configuration page will display the current IP, gateway, DNS information.



IP Address:
192.168.180.8/24

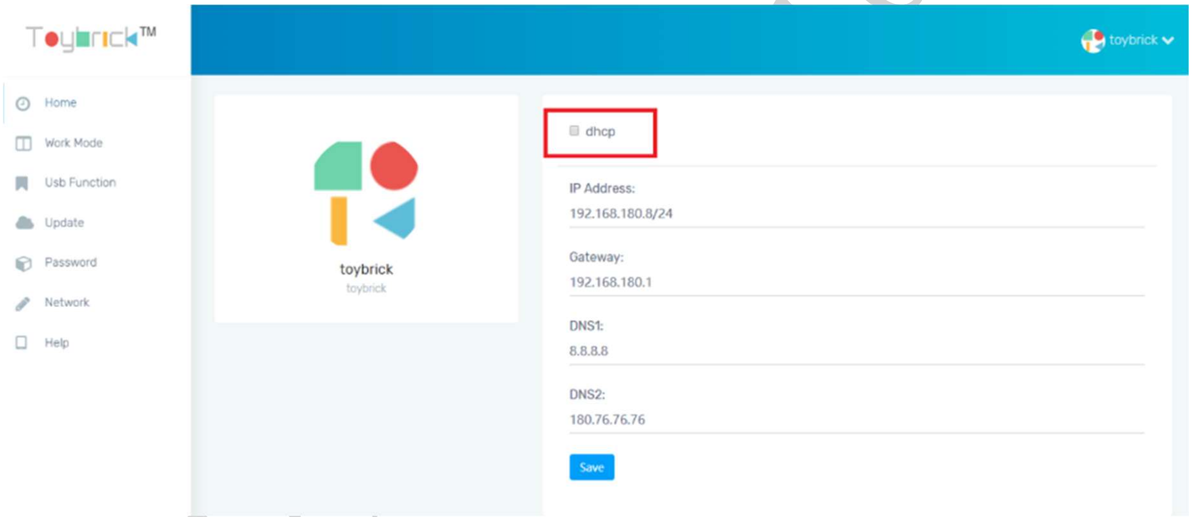
192.168.180.8/24

Gateway:
192.168.180.1

DNS:
8.8.8.8

180.76.76.76

7. Click the setting button to enter the configuration page and configure static IP, gateway and DNS information.



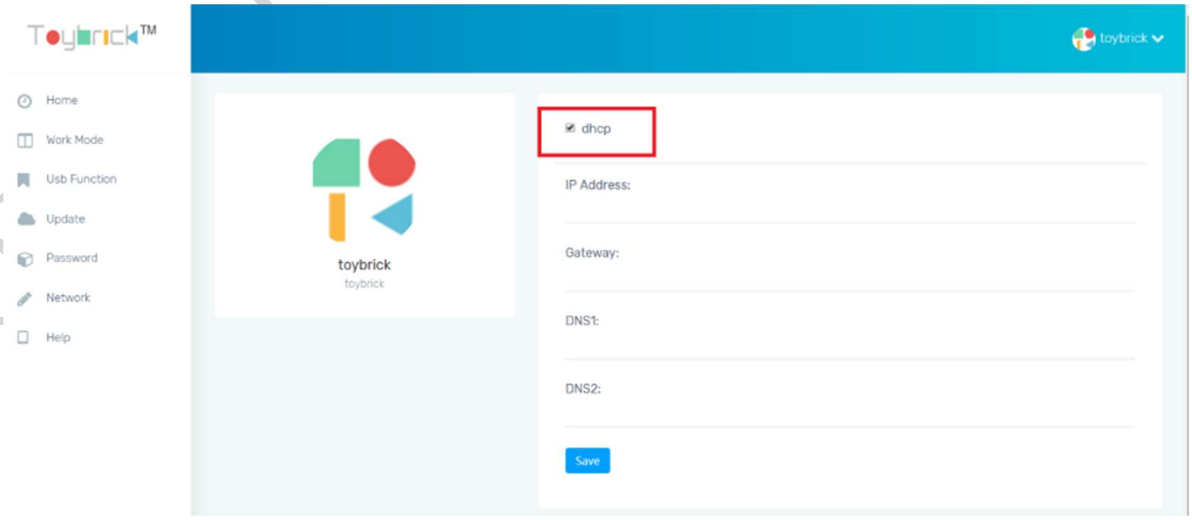
□ dhcp

IP Address:
192.168.180.8/24

Gateway:
192.168.180.1

DNS1:
8.8.8.8

DNS2:
180.76.76.76



☑ dhcp

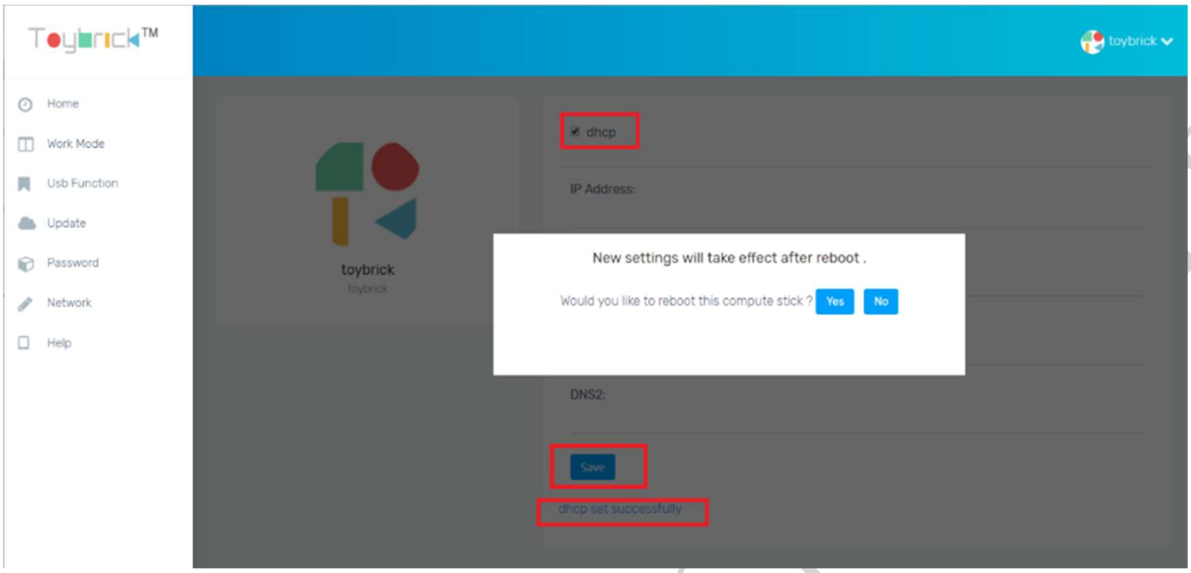
IP Address:

Gateway:

DNS1:

DNS2:

8. If dhcp is checked, the Host needs to be configured with bridging network. Refer to the wiki for details on how to configure the Host with bridging network.



☑ dhcp

IP Address:

New settings will take effect after reboot .

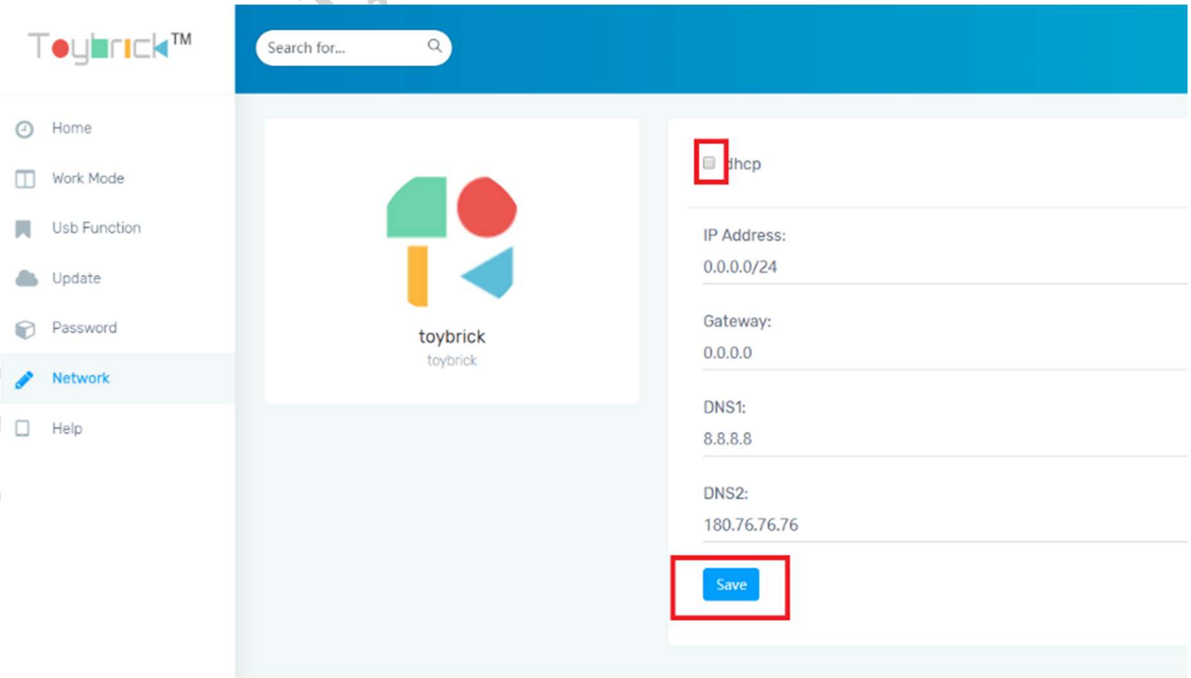
Would you like to reboot this compute stick ?

DNS2:

dhcp set successfully

9. When the configuration is complete, press the save button and click restart to take effect as prompted.

10. Network:RK1808 AI compute stick network configration， Modify static ip, gateway, DNS information and provide dhcp switch (use with master mode). After the modification, must reboot it.



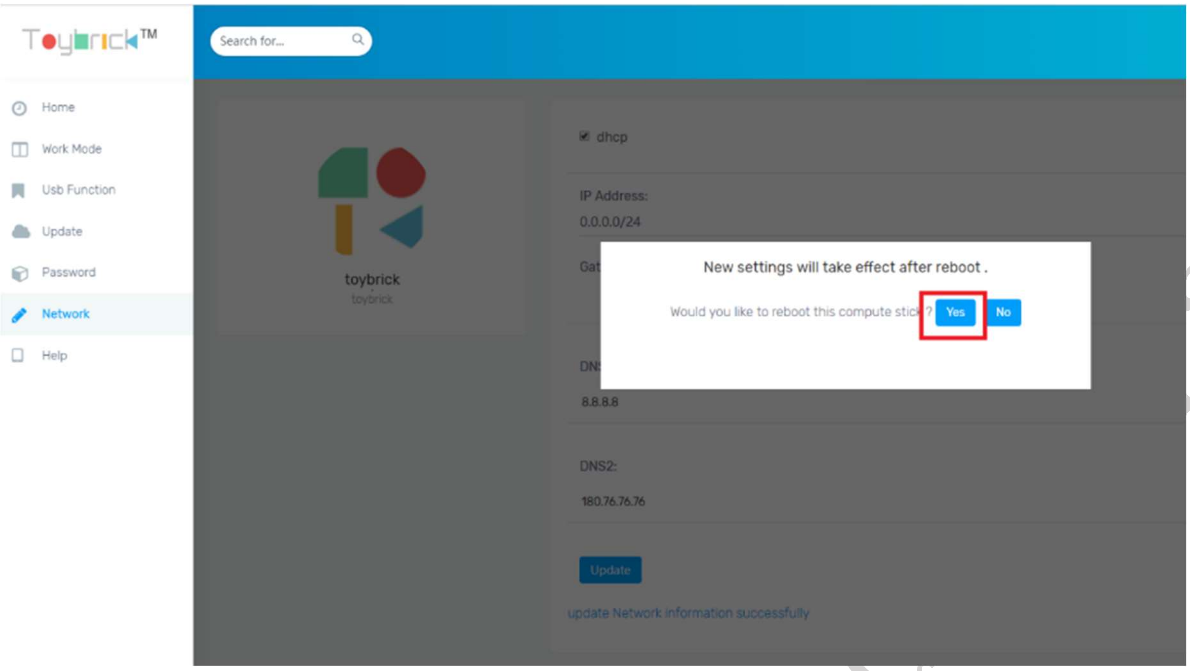
□ dhcp

IP Address:
0.0.0.0/24

Gateway:
0.0.0.0

DNS1:
8.8.8.8

DNS2:
180.76.76.76



☑ dhcp

IP Address:
0.0.0.0/24

DNS2:
180.76.76.76

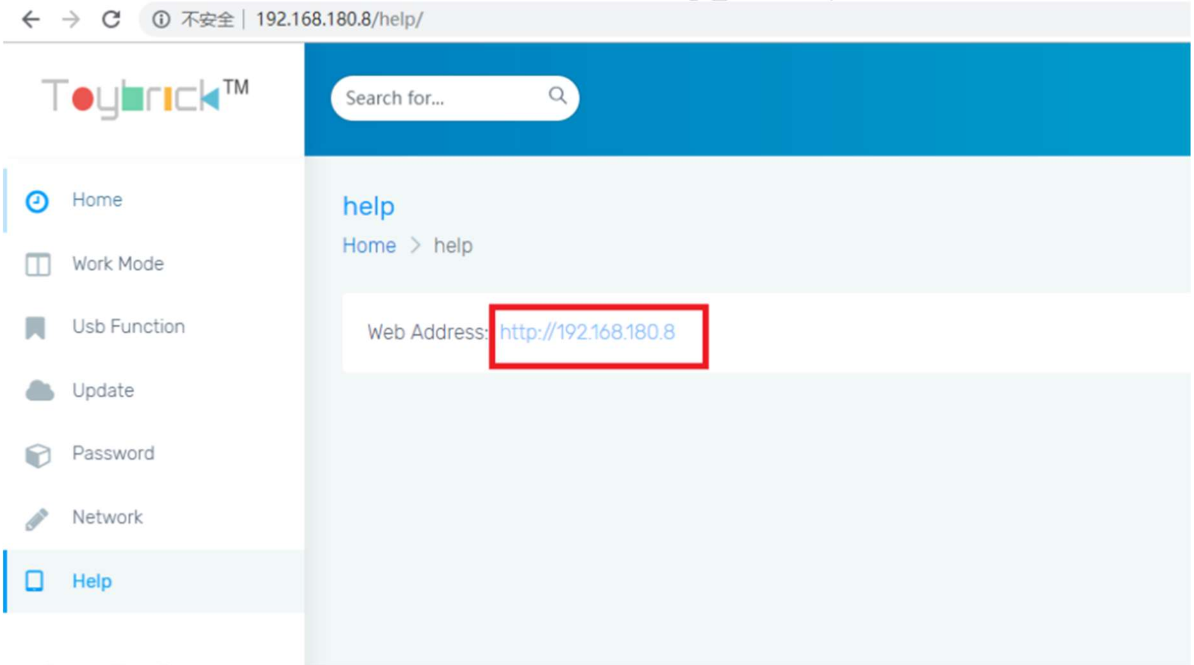
8.8.8.8

New settings will take effect after reboot .

Would you like to reboot this compute stick?

update Network information successfully

11. Help: RK1808 AI compute stick help information for web configuration.

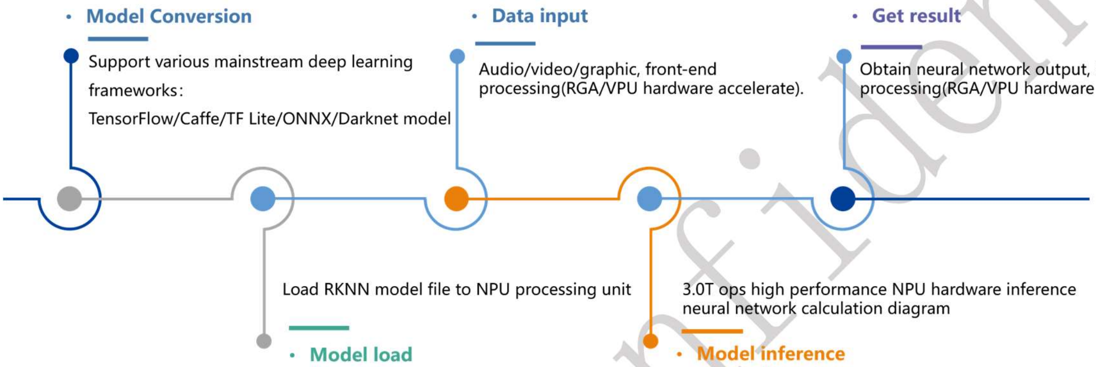


Web Address: http://192.168.180.8

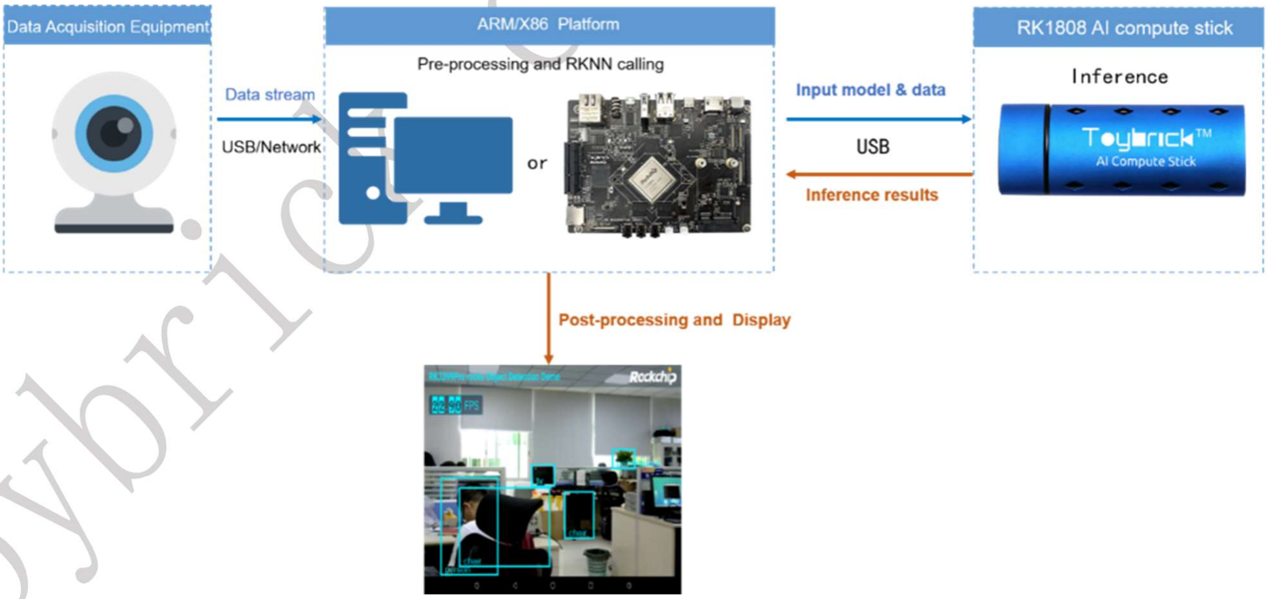
6 Slave mode development

In slave mode, RK1808 AI compute stick is a general AI accelerator. Model and data after pre-processing are inputted into the Host side(model and data are transmitted through rknn toolkit), RK1808 AI compute stick completes inference, and sends result back to the Host side, the Host side does post-processing, display and other manipulations.

Slave mode development overall flowchart:


• Model Conversion
Support various mainstream deep learning frameworks:
TensorFlow/Caffe/TF Lite/ONNX/Darknet model
• Data input
Audio/video/graphic, front-end processing(RGA/VPU hardware accelerate).
• Get result
Obtain neural network output, back-end processing(RGA/VPU hardware accelerate)
Load RKNN model file to NPU processing unit
• Model load
3.0T ops high performance NPU hardware inference neural network calculation diagram
• Model inference

Slave mode data stream flowchart:



In slave mode, we will provide python and C API for host programming.

- When use Python programming, you need install RKNN-Toolkit in the Host. The link which is http://repo.rock-chips.com/python/ have provided the RKNN-Toolkit installation package. The detail instruction about RKNN-Toolkit please refer to "Rockchip_User_Guide_RKNN_Toolkit". For more RKNN-Toolkit document please refer to http://repo.rock-chips.com/rk1808/rknn-toolkit_doc/.

- There are library files and header files for the Host C programming under http://repo.rock-chips.com/rk1808/rknn-api/.The detail instruction about C programming please refer to "Rockchip_User_Guide_RKNN_API". Before executing the compiled executable program of the C language, the host computer needs to run npu_transfer_proxy to communicate with the compute stick. The download link of npu_transfer_proxy is .

RKNN-Toolkit under Windows requires pre-installation of ntb driver, ntb driver please download from http://repo.rock-chips.com/rk1808/driver/windows/ntb/,ntb driver installation guide please refer to wiki.

For more information on slave mode development, please visit the official forum: http://t.rock-chips.com/.

7 Master mode development

The AI program developed in master mode is divided into two parts: the RK1808 AI compute stick AI program and the Host AI application.

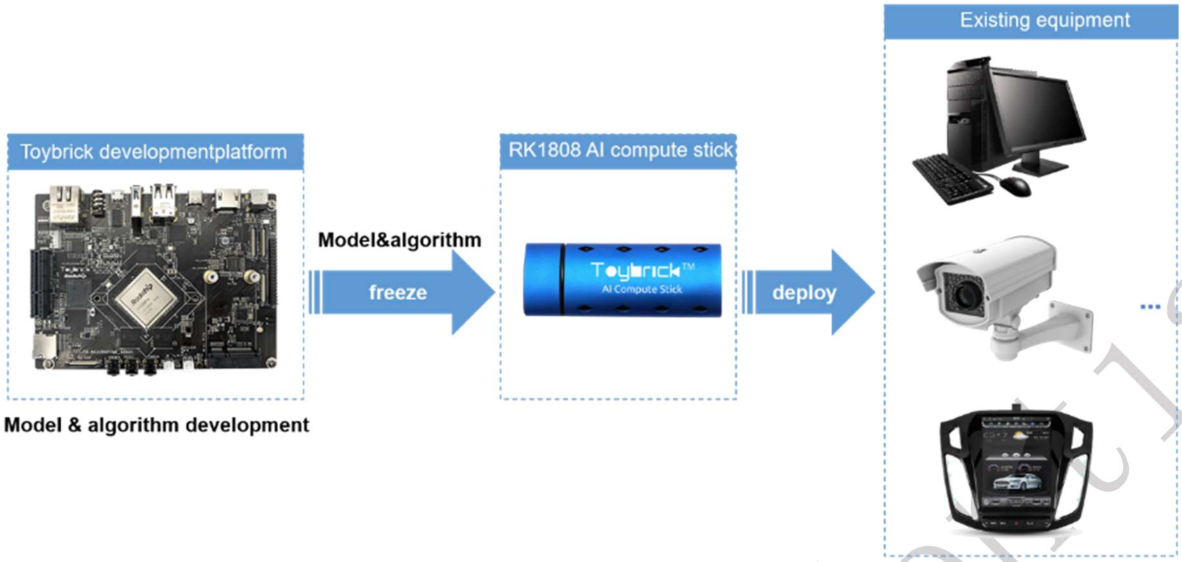
- The model is pre-deployed on the storage of the RK1808 AI compute stick, initialization environment of the stick AI program and loads the model, starts the socket server, receives the data pushed by the Host, performs the inference, and returns the result of the inference to the Host.
- The Host AI application collects data (such as camera data), pushes the data to the RK1808 AI compute stick through the socket client, and receives the processing result returned by the computing stick through the socket client, and performs further processing (such as display).

Under master mode, RK1808 AI computing stick is a professional AI application module. As a master device, RK1808 AI computing stick solidifies the model and algorithm. Host only needs to input data to computing stick through USB (for example image and video flow), and RK1808 AI compute stick automatically finish data's pre-processing, inference, post-processing, then output processing result through USB to Host.

In order to make it convenient for users to transmit data through USB, RK1808 AI computing stick will visualize USB port as standard devices such as network interface card. Users only need to manipulate standard device (for example network socket programming) to accomplish input and output of data in computing stick.

RK1808 AI compute stick with fedora operation system, Users can log in the fedora system for development and debugging via ssh. Root user's and normal user's password both are "toybrick".

Master mode development overall flowchart:



7.1 AI development process introduction under host mode

RK1808 AI compute stick RKNN API calling under master mode refer to "Rockchip_RK1808_Developer_Guide_Linux_RKNN".

RK1808 AI computing stick has already offered RKNN API C/C++ needed library and header files. At the same time it has also installed RKNN API's python 3.6 lib. User can develop and deploy C/C++ or python active AI program in RK1808 AI compute stick.

7.2 Introduction of product deployment

1. The model and algorithm are freeze in the RK1808 AI compute stick, and the program set in the RK1808 AI compute stick is set to start up.
2. RK1808 AI compute stick is inserted into target equipment, such as web camera equipment, PC, drone, smart car, etc.
3. Running the Host service program in target device, and display processing result.
4. You find the Yolov3 master mode demo in wiki.

7.3 Development tool toybrick_deployc

In order to facilitate customer development and deployment, a set of master mode development tool toybrick_deployc is provided. This tool can only be run on the toybrick development platform.

For the specific usage of the toybrick_deployc tool, please refer to the wiki.