

RK1808 AI Compute Stick User manual

V1.2.0

Toybrick Confidential

Table of Contents

1	OVERVIEW.....	3
2	SPECIFICATION.....	4
2.1	MINI-SYSTEM REQUIREMENT	4
3	DOCUMENT DESCRIPTION.....	5
4	GETTING START	6
4.1	ENVIRONMENT PREPARATION.....	6
4.2	RKNN-TOOLKIT INSTALLATION	6
4.3	DEVICE ACCESS PERMISSION MODIFICATION	8
4.4	RUNNING DEMO ON RK1808 AI COMPUTE STICK.....	9
5	WEB CONFIGURATION	12
5.1	THE HOST NETWORK CONFIGURATION.....	12
5.2	WEB LOGIN	17
5.3	INTRODUCTION OF MAIN FUNCTIONS OF WEB PAGES	18
6	SLAVE MODE DEVELOPMENT.....	25
7	MASTER MODE DEVELOPMENT	27
7.1	AI DEVELOPMENT PROCESS INTRODUCTION UNDER HOST MODE.....	28
7.2	DEVELOPMENT TOOL TOYBRICK_DEPLOYC	28
7.2.1	<i>Development tool toybrick_deployc command list and command illustration</i>	<i>28</i>
7.2.2	<i>Development process based on Toybrick platform.....</i>	<i>31</i>
7.2.3	<i>Introduction of product deployment.....</i>	<i>32</i>

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°C~40°C
Size	82x31x13mm

2.1 Mini-system requirement

- Ubuntu 16.04 or 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	RK1808_RKNN_SDK_DEVELOPER_GUIDE_EN.pdf	The guide of RK1808 RKNN SDK programming for C API and python API.
	RK1808_RKNN_SDK_DEVELOPER_GUIDE_CN.pdf	
	RK3399Pro_Linux&Android_RKNN_API_EN.pdf	The guide of RKNN SDK programming for C API in the Host.
	RK3399Pro_Linux&Android_RKNN_API_CN.pdf	
	RKNN_Toolkit User_Guide_EN.pdf	RKNN-Toolkit and python API user guide.
	RKNN_Toolkit User_Guide_CN.pdf	
	RK1808_AI_Compute_Stick_User_manual_EN.pdf	RK1808 AI compute stick user manual.
	RK1808_AI_Compute_Stick_User_manual_CN.pdf	
	RK1808_AI_Compute_Stick_Easy_Start_Demo_Guide_EN.pdf	The guide of Easy start yovov3 demo.
	RK1808_AI_Compute_Stick_Easy_Start_Demo_Guide_CN.pdf	
driver	ntb	USB ntd windows driver.
example	mobilenet_v1	Mobilenet demo.
	yolov3	The guide of Easy start yovov3 demo.
rknn	Npu_transfer_proxy	RKNN-Toolkit the Host communication agenda.
	Rknn-api	The Host C library and head file for RKNN SDK programming.
	Rknn-toolkit	RKNN-Toolkit installation package.
tool		Script and tool.

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

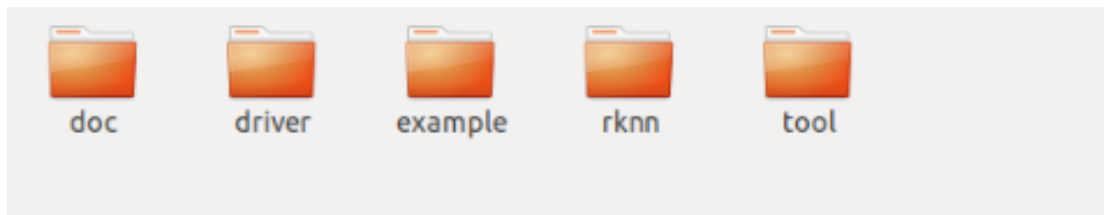
```
sudo apt-get install python3.6
```

- Install 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 directory structure of the usb device is as follows:



- Perform the following steps:

1. Enter root directory of U disk:
2. Enter rknn/rknn-toolkit/ directory.

```
cd rknn/rknn-toolkit/
```

3. Install Python requirement:

```
pip3 install --user -r requirements-ubuntu18.04-python3.6.txt
```

4. Install RKNN-Toolkit:

```
pip3 install --user rknn_toolkit-1.1.0b1-cp36-cp36m-linux_x86_64.whl
```

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

5. 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 “/tmp” directory. Modify update_rk1808_ai_cs_rule.sh with executable permission.
 - 2) update_rk1808_ai_cs_rule.sh must be executed with root authority.

```
sudo /tmp/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.

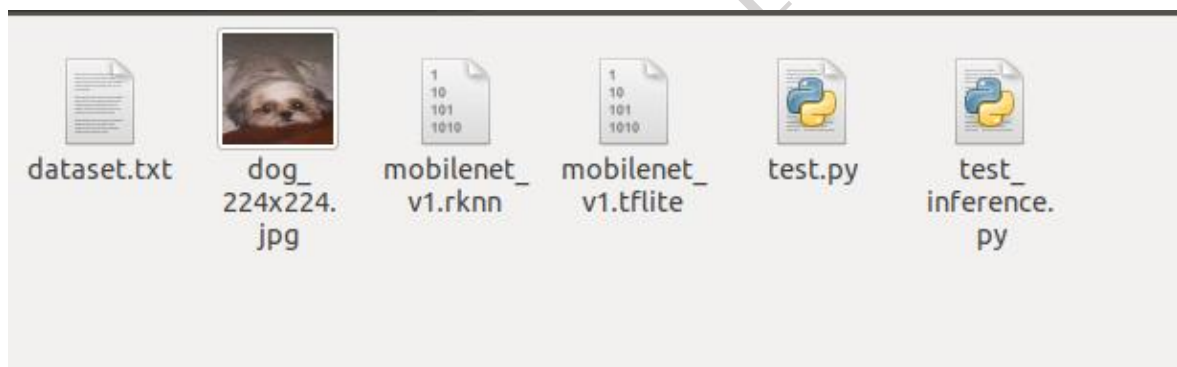
- 2) The execution result as shown below (The correct read and write permissions are shown in the red mark):

```
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. Back to U disk root directory, copy example/mobilenet_v1 directory to /tmp, then switch to this directory :

```
cp -rf example/mobilenet_v1/ /tmp/  
cd /tmp/mobilenet_v1/
```

2. Execute test.py and get the following results:

```
python3.6 test.py
```

```
--> 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): 7320  
FPS: 136.61  
=====
```

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.

Notice:

1.Yolov3 easy start demo refer to 《RK1808 AI_Compute_Stick_Easy Start_Demo_Guide_EN.pdf》.

2.For more detailed usage and interface instructions of the RKNN-Toolkit, please refer to 《RKNN-Toolkit user guide_v1.1.0》.

toybrick Confidential

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”.

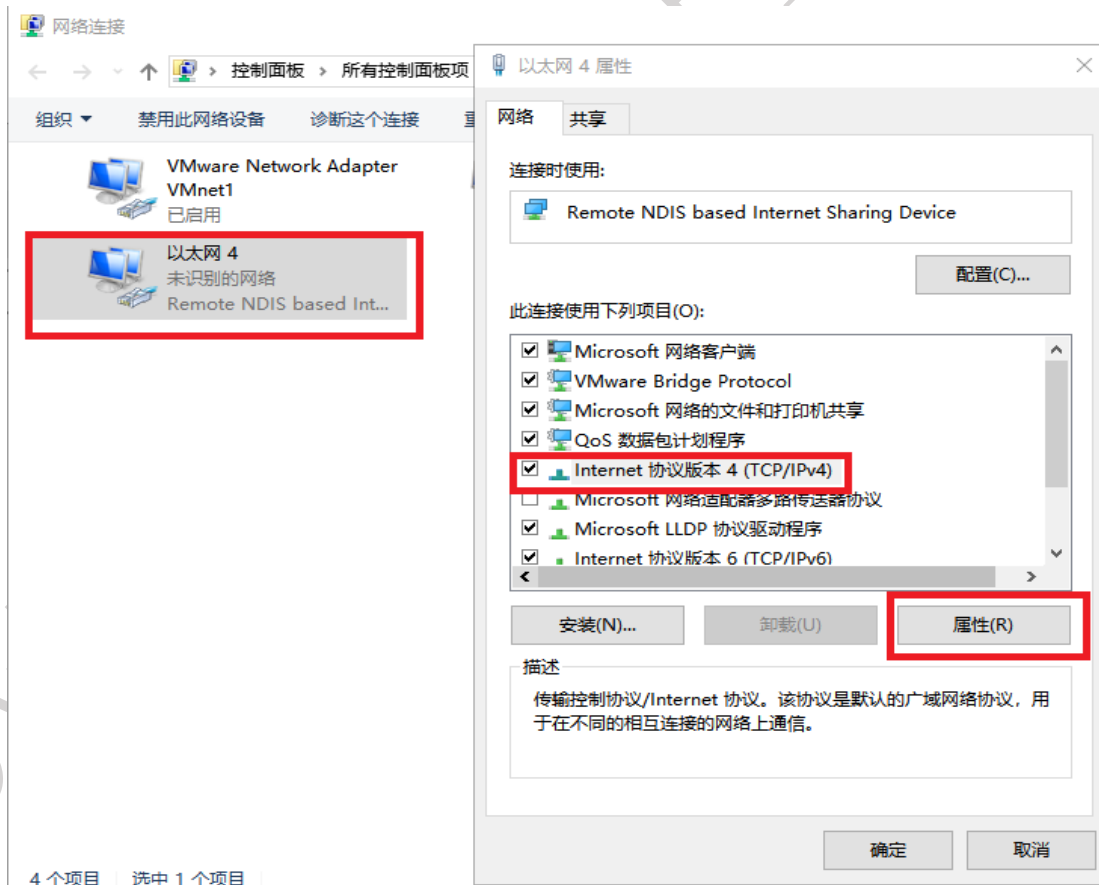


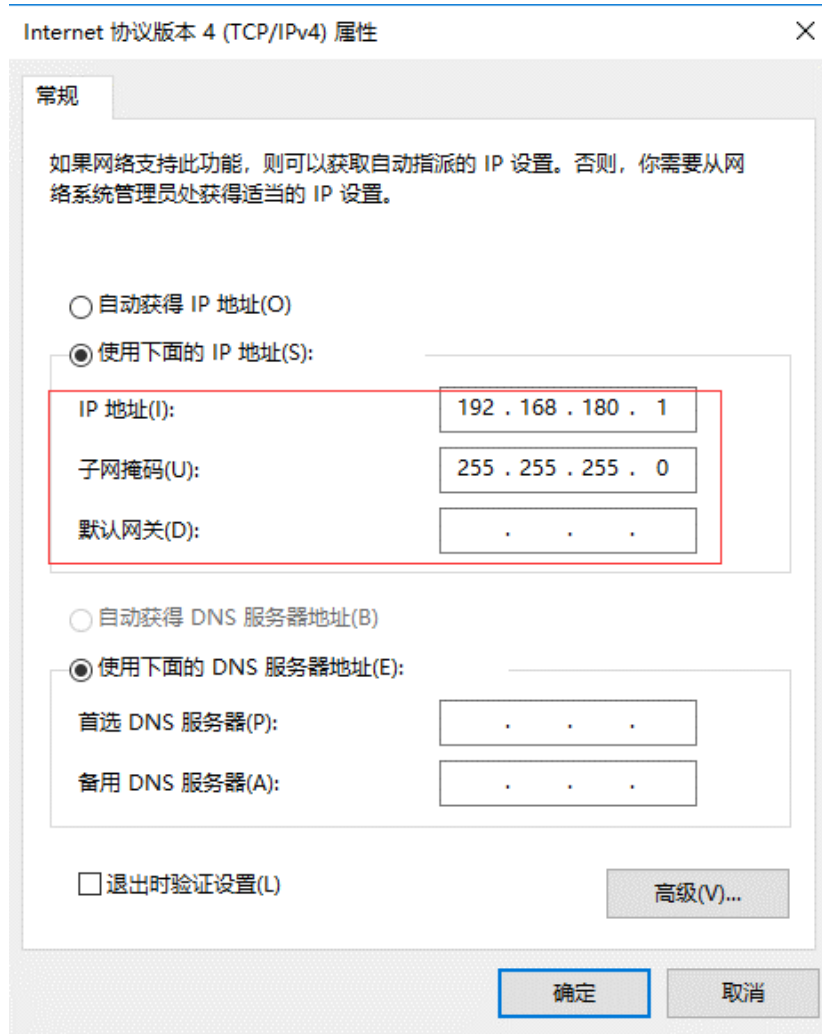
Notice: Some computer may have unrecognized devices, uninstall unrecognized devices, and wait for the RNDIS driver to reinstall.

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



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.





5. Check the IP address.

```
D:\软件备份\adb>ipconfig

Windows IP 配置

以太网适配器 本地连接:

    连接特定的 DNS 后缀 . . . . . :
    IPv4 地址 . . . . . : 172.16.9.43
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . : 172.16.9.1

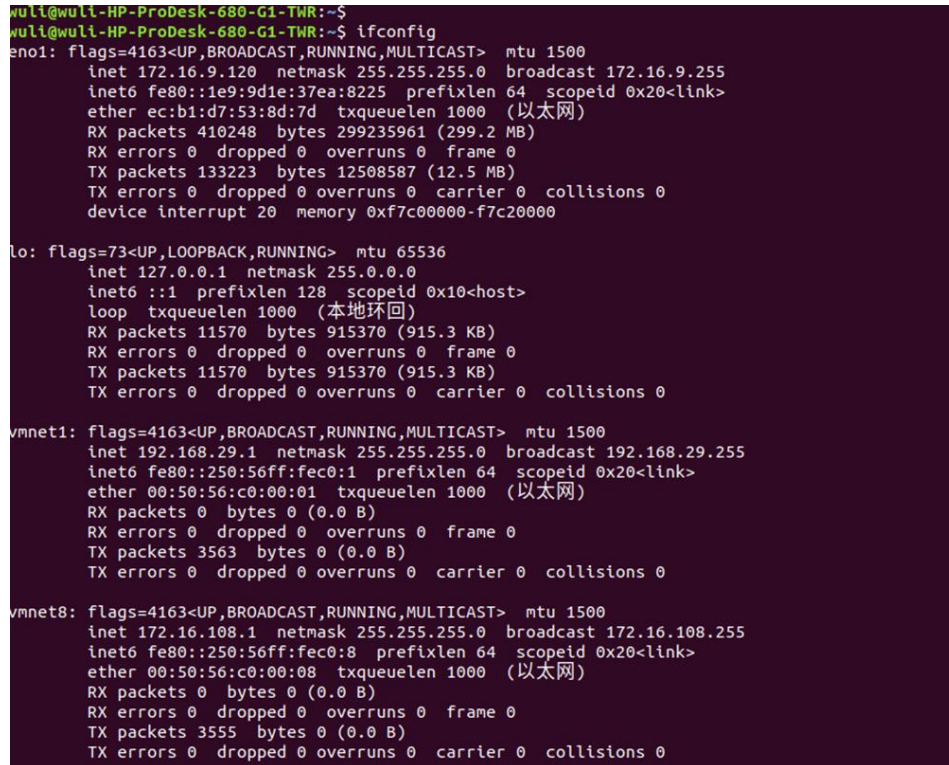
以太网适配器 以太网 2:

    连接特定的 DNS 后缀 . . . . . :
    本地连接 IPv6 地址 . . . . . : fe80::a406:b196:6ec8:7f06%5
    IPv4 地址 . . . . . : 192.168.180.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . :
```

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:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.9.120 netmask 255.255.255.0 broadcast 172.16.9.255
    inet6 fe80::1e9:9d1e:37ea:8225 prefixlen 64 scopeid 0x20<link>
    ether ec:b1:d7:53:8d:7d txqueuelen 1000 (以太网)
    RX packets 410248 bytes 299235961 (299.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 133223 bytes 12508587 (12.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xf7c00000-f7c20000

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 (本地环回)
    RX packets 11570 bytes 915370 (915.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11570 bytes 915370 (915.3 KB)
    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 (以太网)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3563 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vmnet8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.108.1 netmask 255.255.255.0 broadcast 172.16.108.255
    inet6 fe80::250:56ff:fec0:8 prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:c0:00:08 txqueuelen 1000 (以太网)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3555 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:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.9.120 netmask 255.255.255.0 broadcast 172.16.9.255
    inet6 fe80::1e9:9d1e:37ea:8225 prefixlen 64 scopeid 0x20<link>
    ether ec:b1:d7:53:8d:7d txqueuelen 1000 (以太网)
    RX packets 413671 bytes 299653869 (299.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 133854 bytes 12591302 (12.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xf7c00000-f7c20000

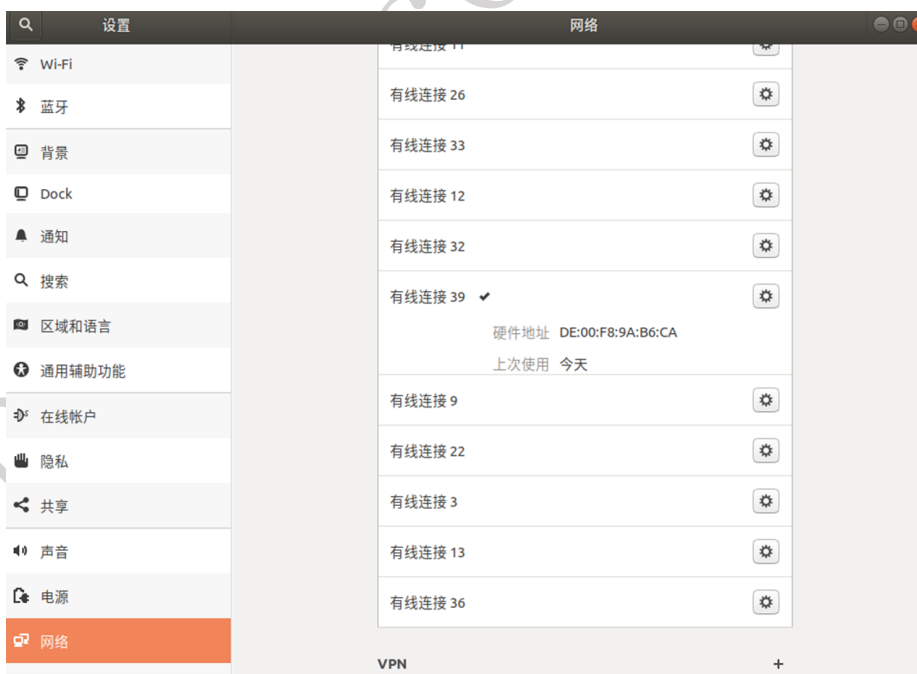
enp0s20u11: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::c803:409f:4aac:24eb prefixlen 64 scopeid 0x20<link>
    ether de:00:f8:9a:b6:ca txqueuelen 1000 (以太网)
    RX packets 46 bytes 1576 (1.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 396 bytes 99557 (99.5 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 (本地环回)
    RX packets 12126 bytes 965472 (965.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12126 bytes 965472 (965.4 KB)
    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 (以太网)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3625 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vmnet8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.108.1 netmask 255.255.255.0 broadcast 172.16.108.255
    inet6 fe80::250:56ff:fec0:8 prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:c0:00:08 txqueuelen 1000 (以太网)
    RX packets 0 bytes 0 (0.0 B)
```

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.

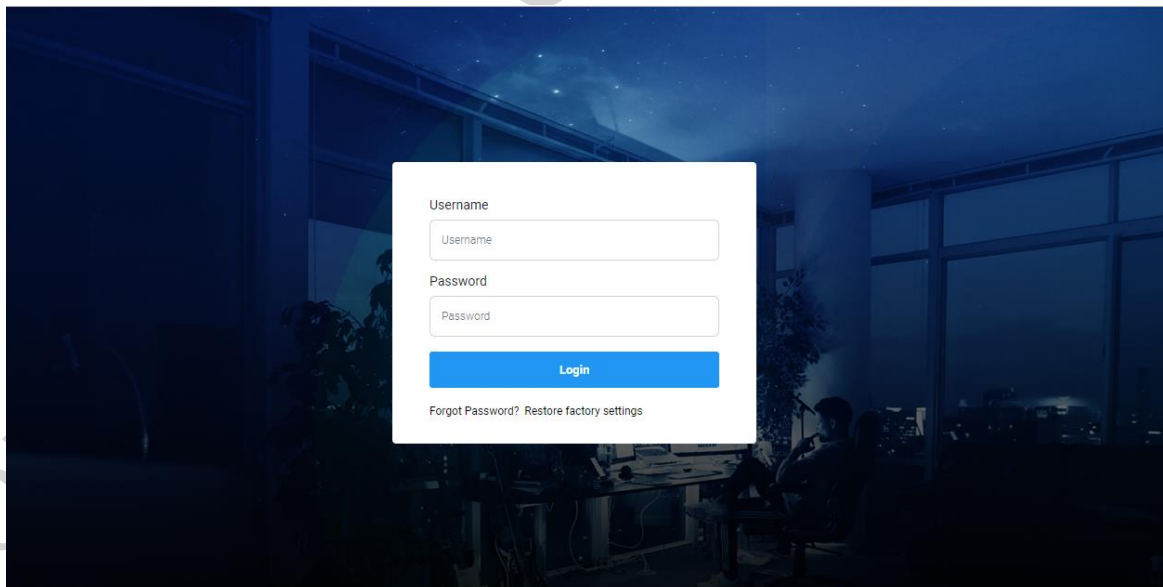


4. According to the figure below, select ipv4 ,select manually ,add a new IP -- 192.168.180.1 255.255.255.0.Finally, Click apply to complete the network .



5.2 WEB Login

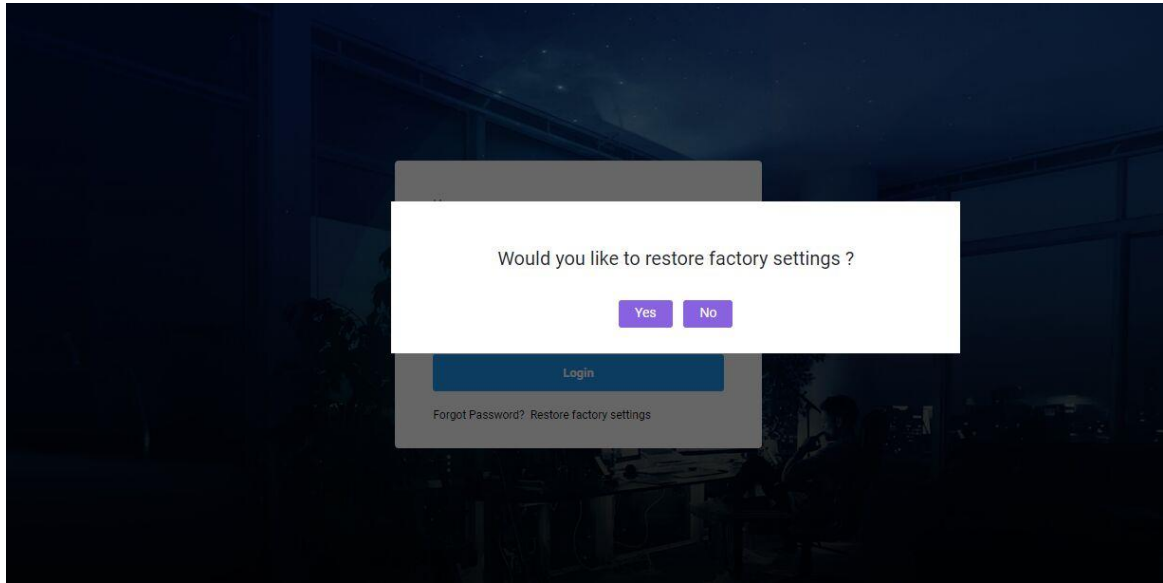
1. After Network configuration is completed, you can input `http://192.168.180.8` in the browser. And then into RK1808 AI compute stick Login page.



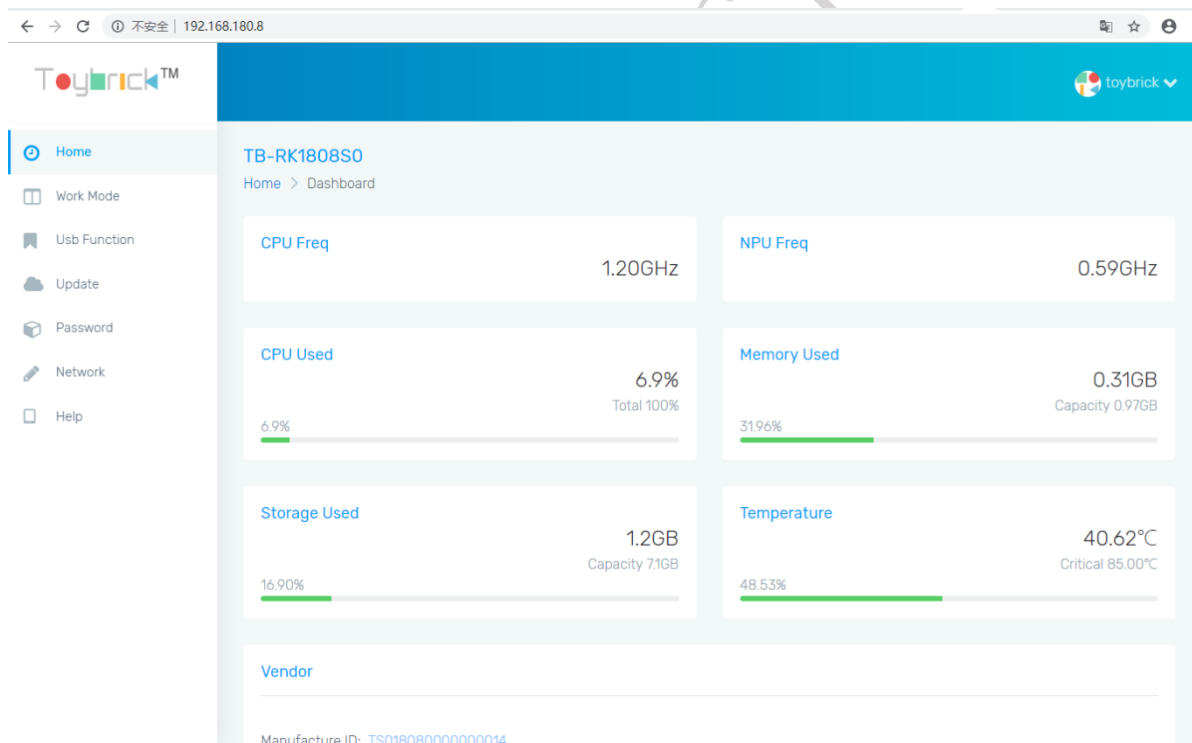
If you forget the password, please click “Forget Password”.

Notice: It will not only reset the password, but also restore the RK1808 AI compute stick to factory

Settings. Please be careful

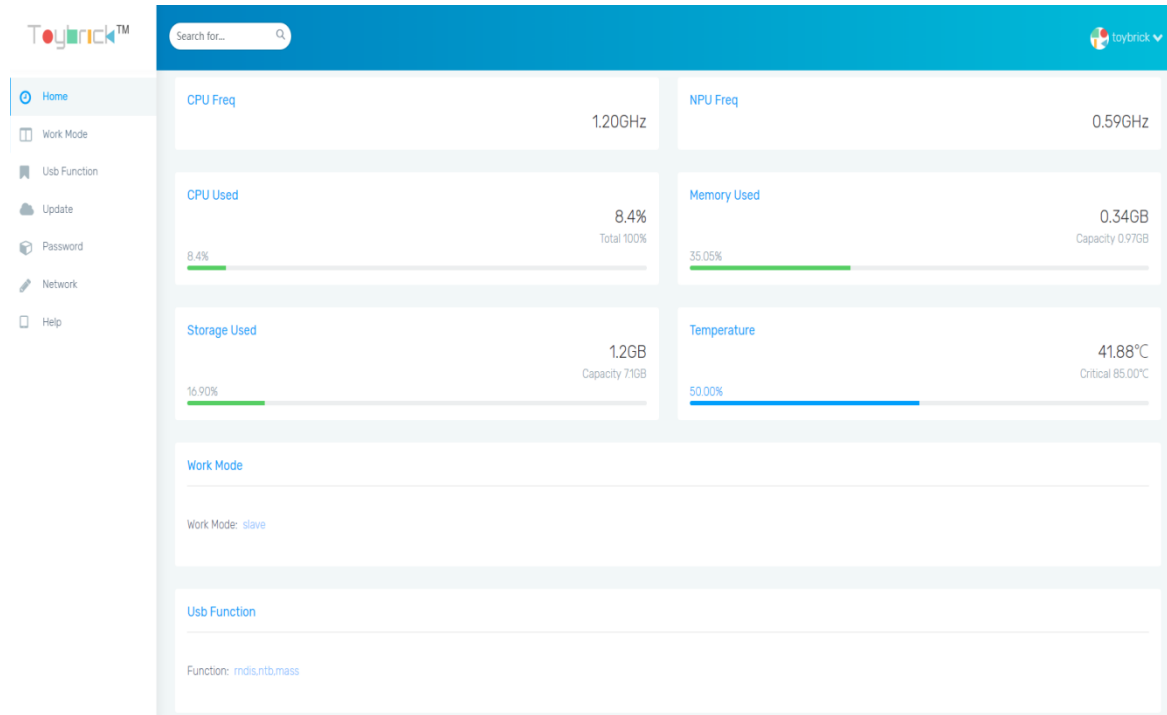


2. Enter username and password to login (default username and password are both toybrick), and then enter the Home page.

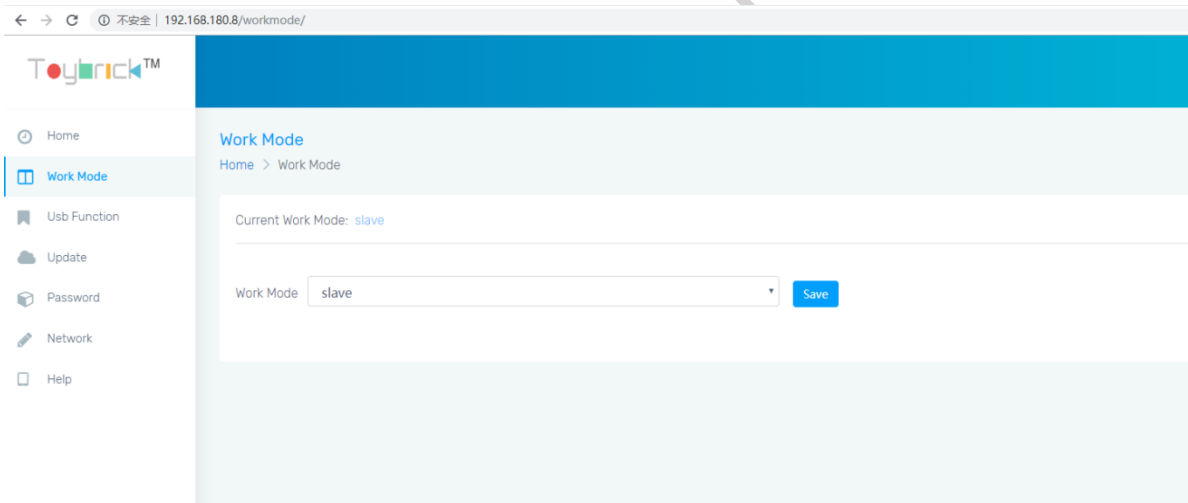


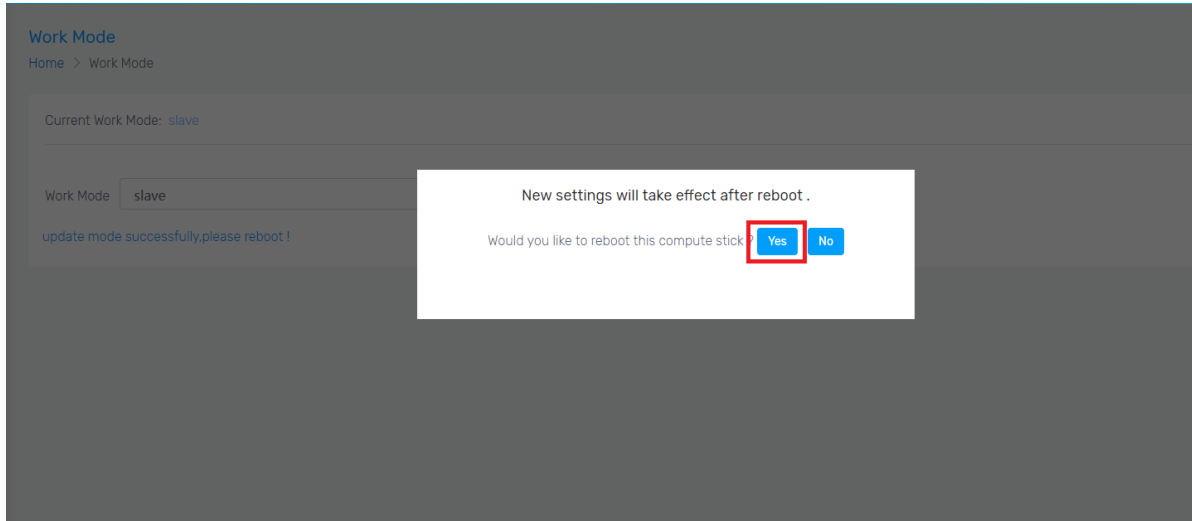
5.3 Introduction of Main Functions of WEB Pages

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

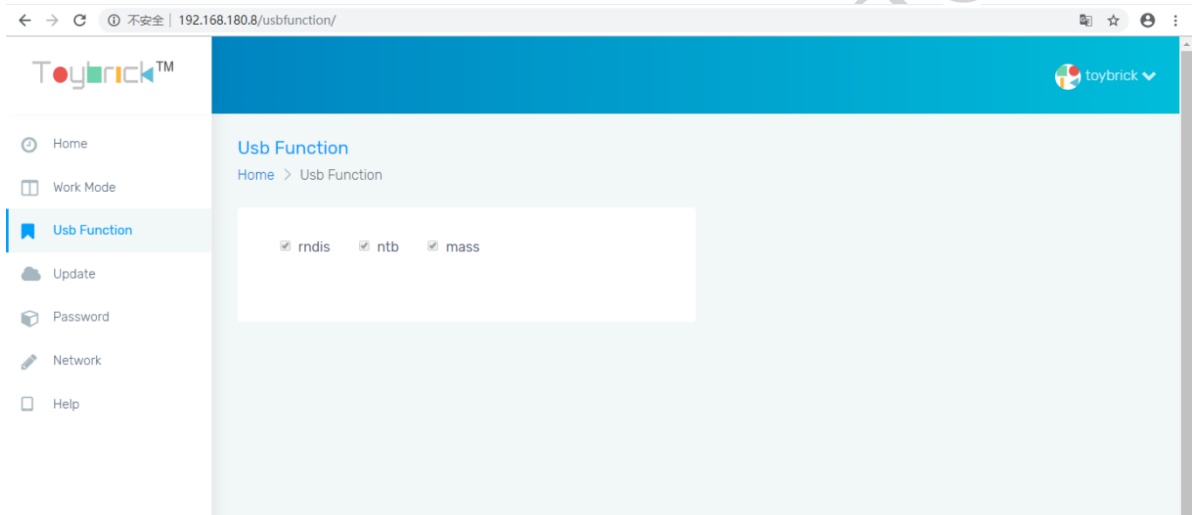


2. Work Mode: The working mode is master or slave.

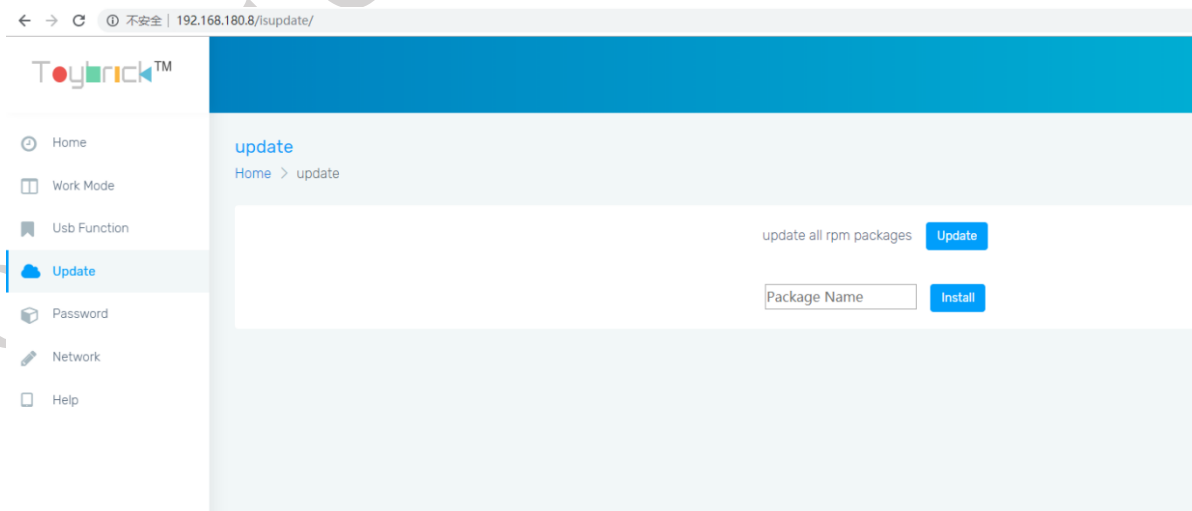




3. USB Function: RK1808 AI compute stick Usbfunction(rndis/ntb/mass)。

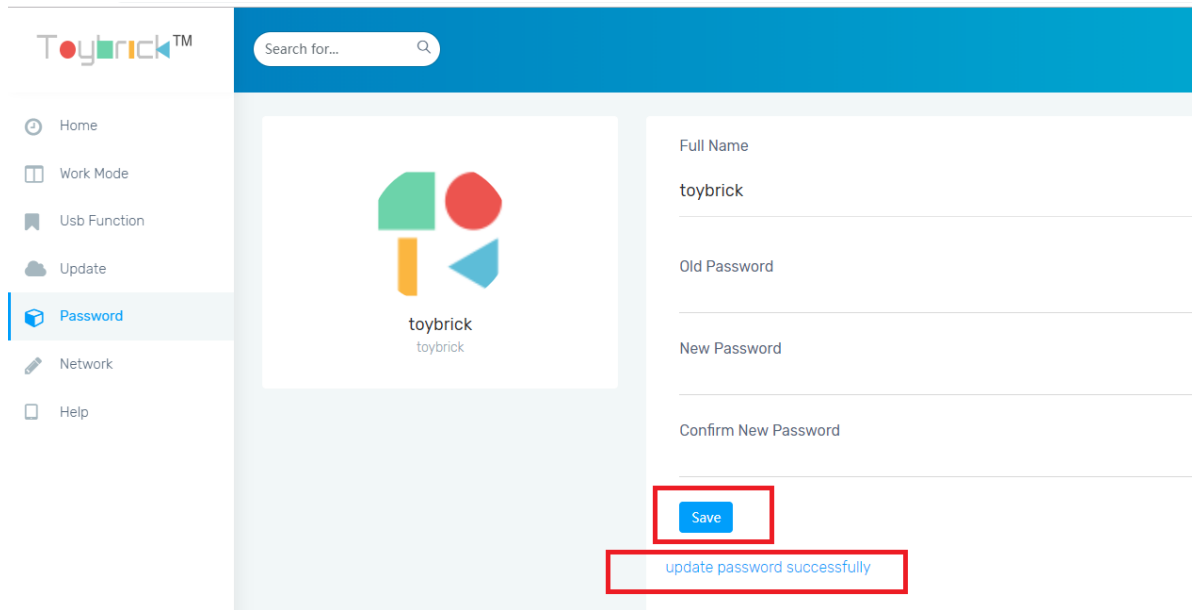


4. Update: Update, Install RK1808 AI compute stick RPM resources.

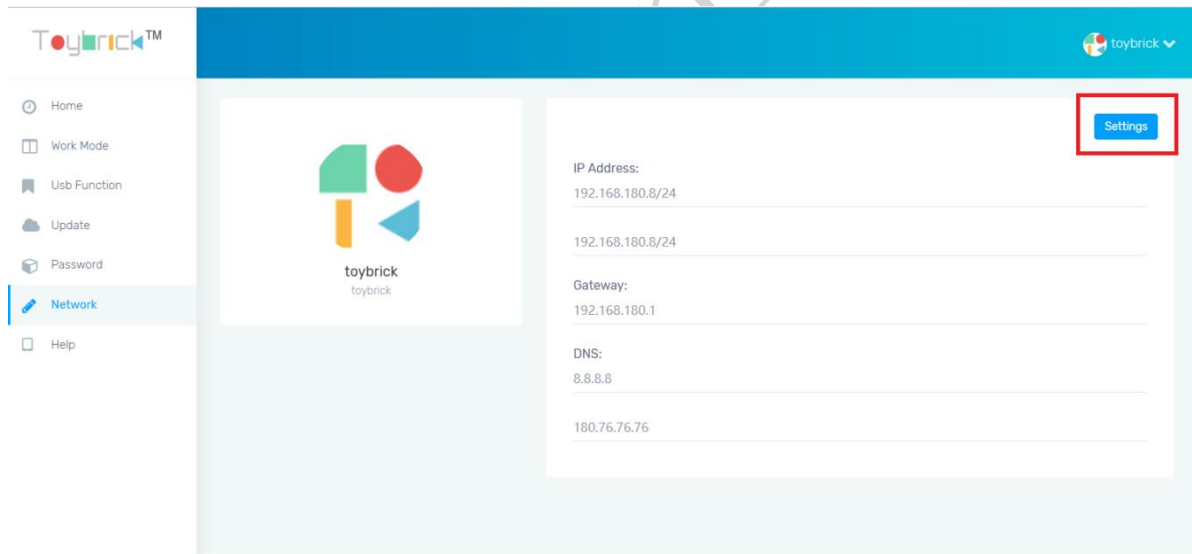


5. Password: You can modify the login password here. After the modification, you need to

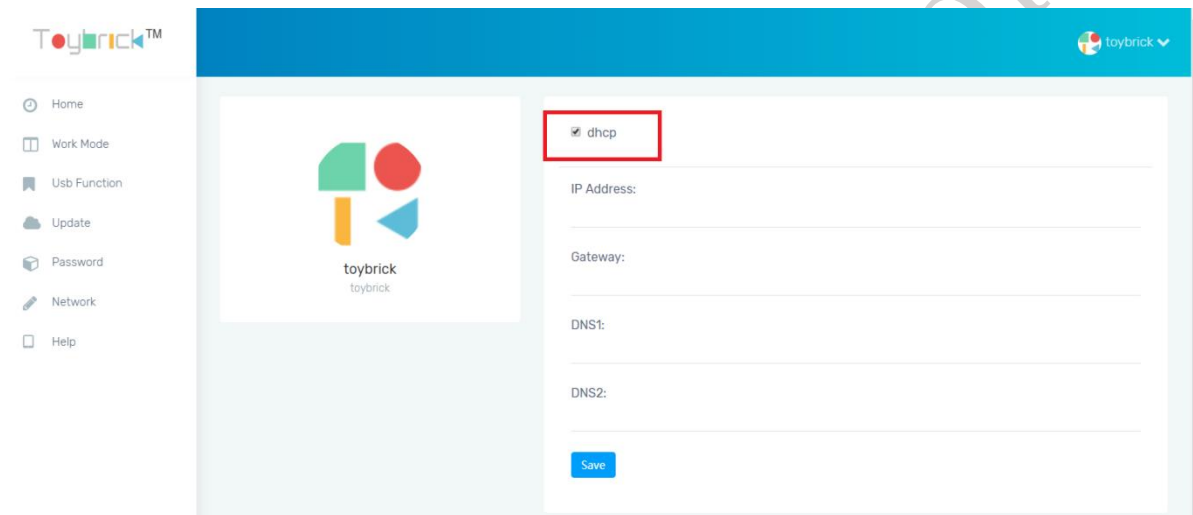
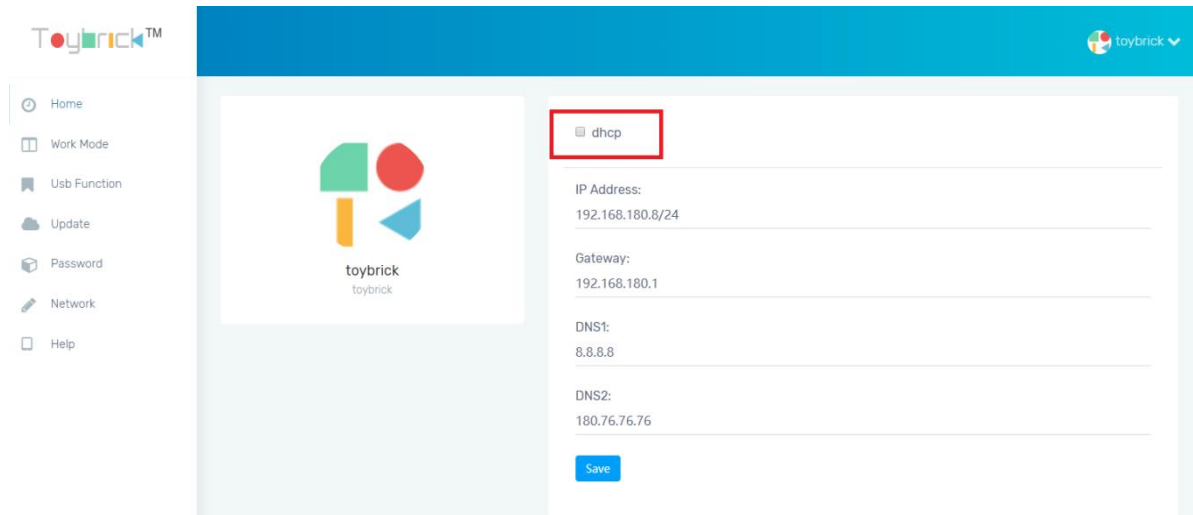
login again. The blue word in the figure indicates that the modification is successful.



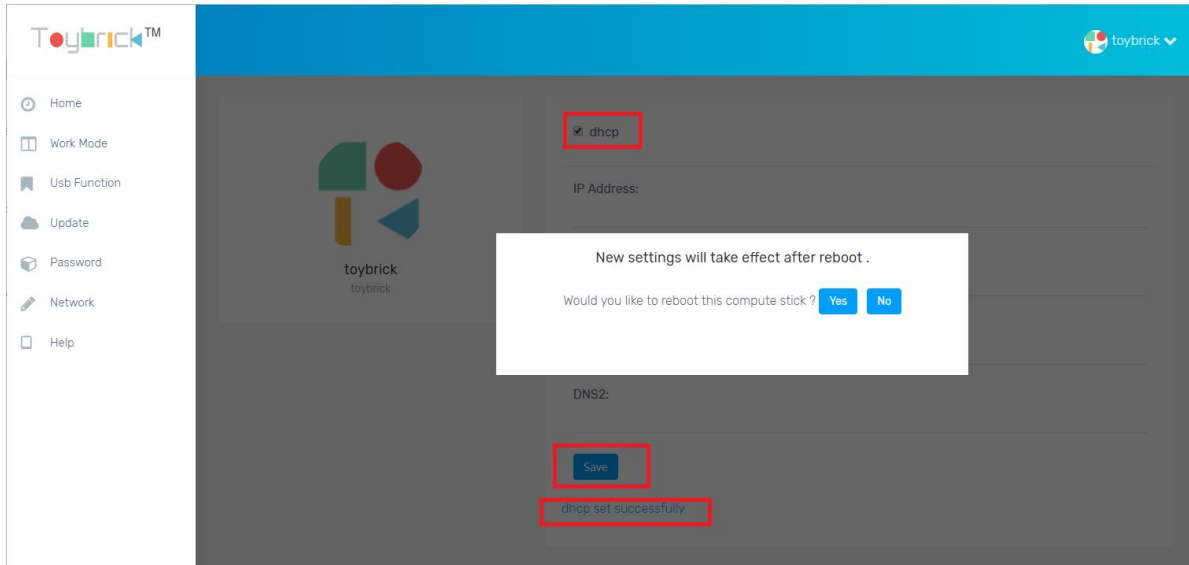
6. Network: RK1808 AI compute stick network configuration page will display the current IP, gateway, DNS information.



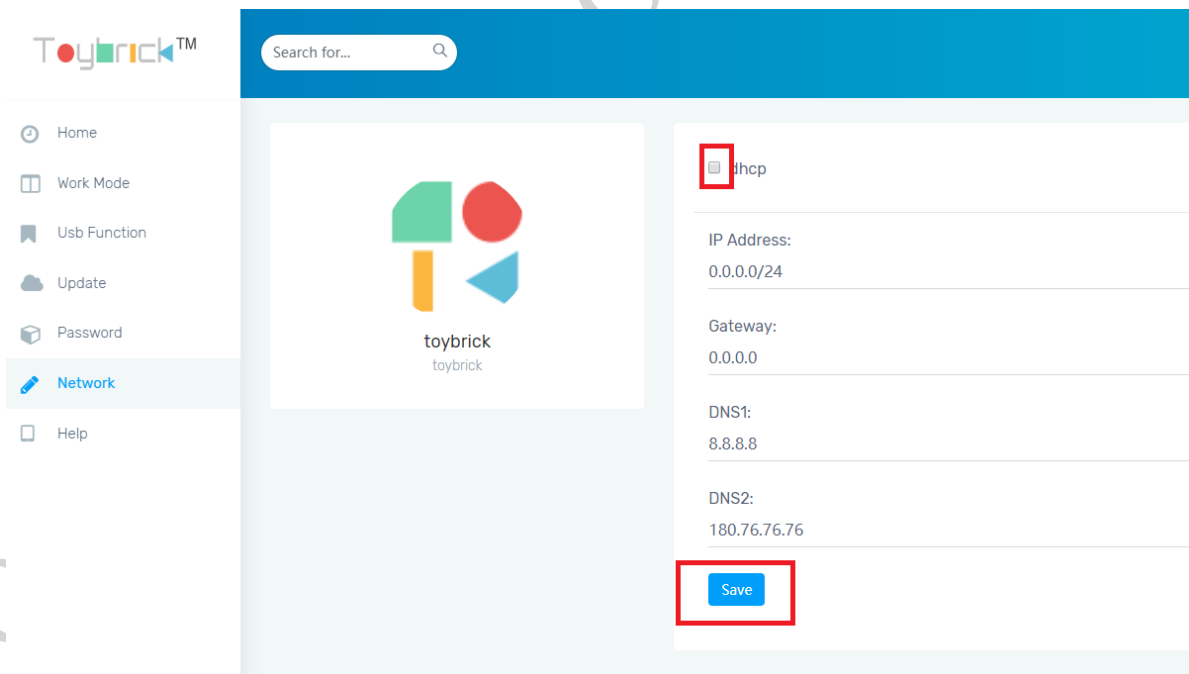
7. Click the setting button to enter the configuration page and configure static IP, gateway and DNS information.

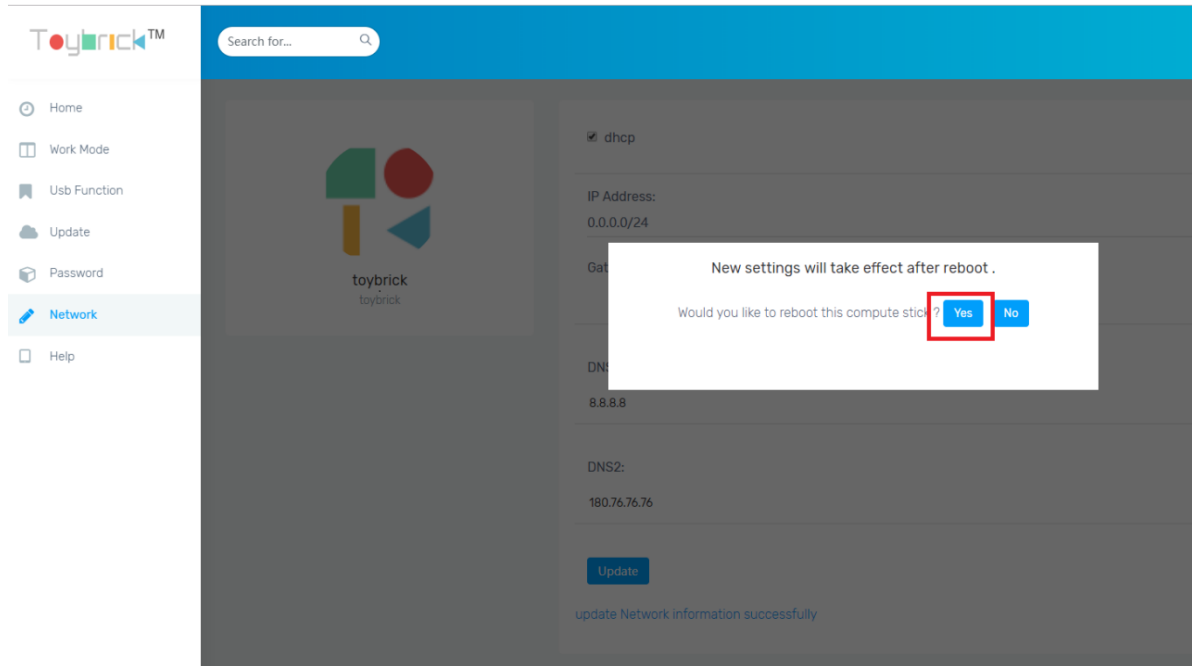


8. If dhcp is checked, the Host needs to be configured with bridging network. Refer to the wiki for details on how to configure the Host with bridging network.

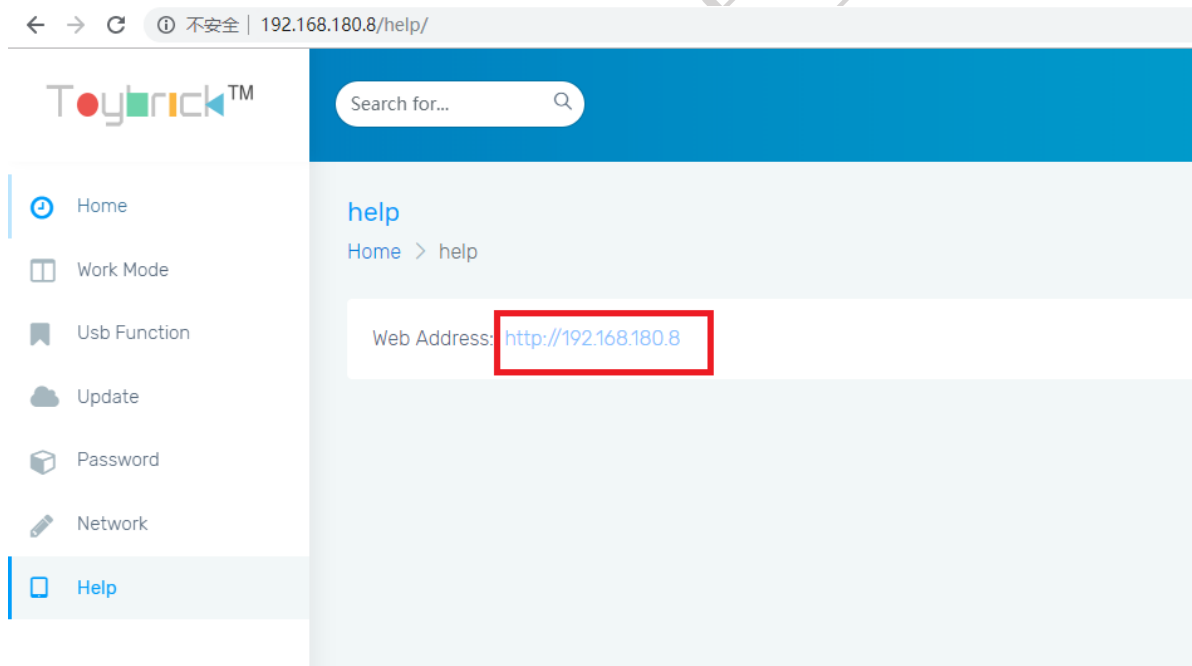


9. When the configuration is complete, press the save button and click restart to take effect as prompted.
10. Network:RK1808 AI compute stick network configuration, Modify static ip, gateway, DNS information and provide dhcp switch (use with master mode). After the modification, must reboot it.





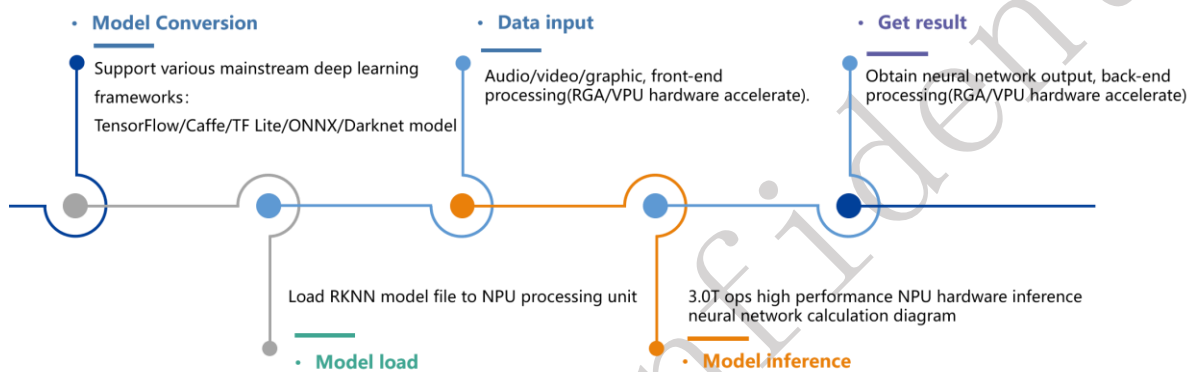
11. Help: RK1808 AI compute stick help information for web configuration.



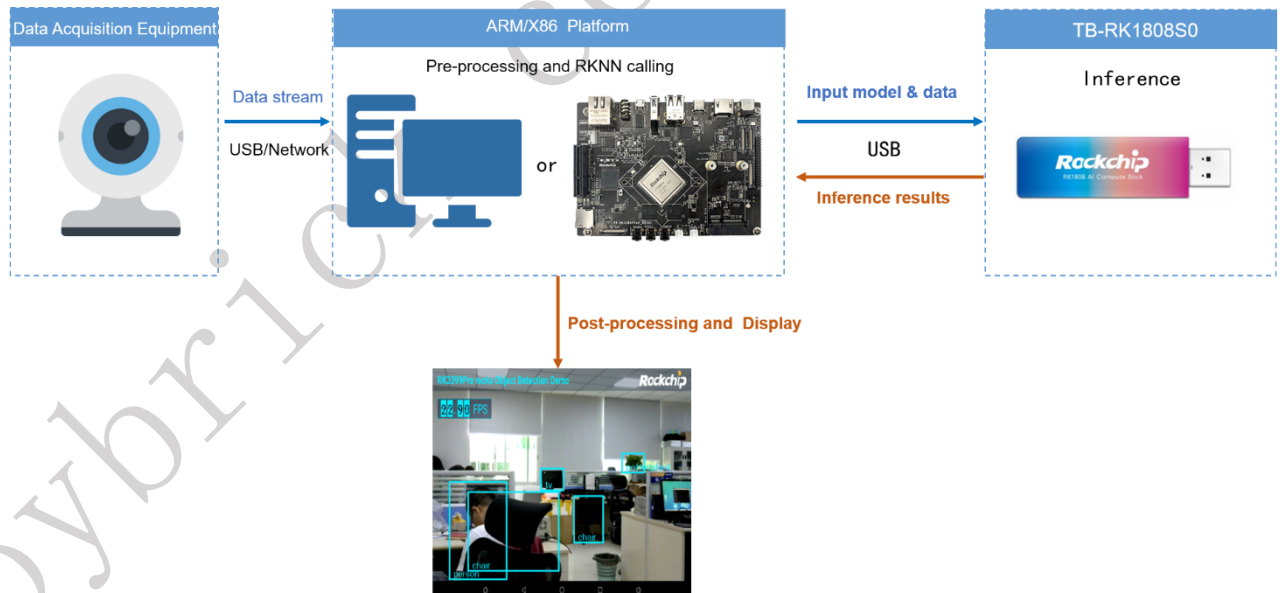
6 Slave mode development

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:



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. RK1808 AI compute stick U disk directory have provided the RKNN-Toolkit installation package.

The detail instruction about RKNN-Toolkit please refer to 《RKNN-

Toolkit_User_Guide_EN.pdf》 .

- There are library files and header files for the Host C programming under RK1808 AI compute stick U disk directory : rknn/rknn-api/. The detail instruction about C programming please refer to 《RK3399Pro_Linux&Android_RKNN_API_EN.pdf》 .

toybrick confidential

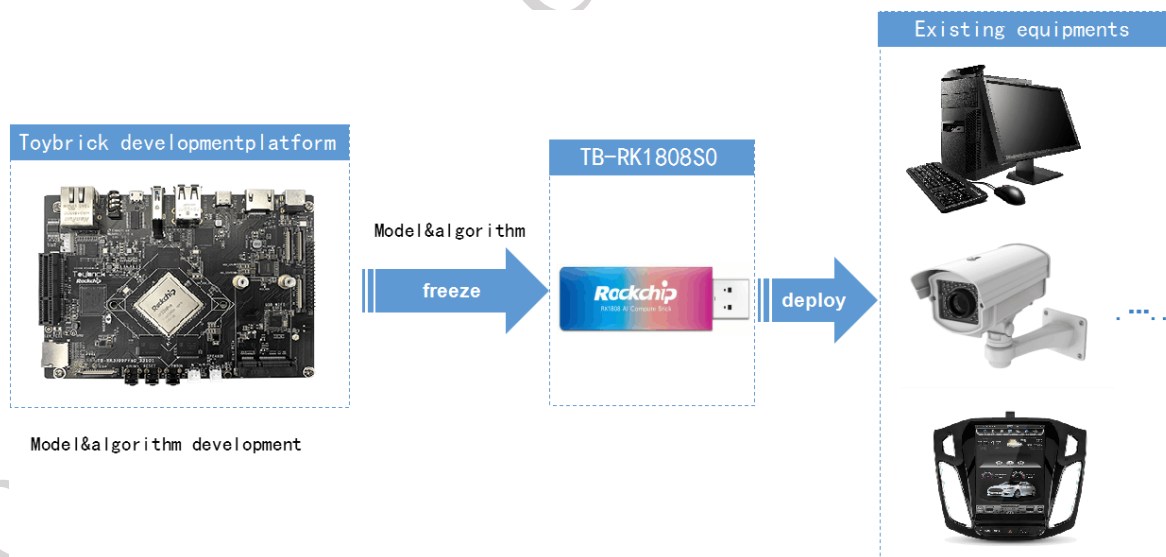
7 Master mode development

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: 《RK1808_RKNN_SDK_DEVELOPER_GUIDE_EN.pdf》 .

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 Development tool toybrick_deployc

In order to facilitate customer development, debugging and deployment, we also provide a set of master mode development tools. Master mode's development tool is toybrick_deployc, which can only run on toybrick development platform. The download address of toybrick development platform's firmware can be achieved by accessing wiki.

7.2.1 Development tool toybrick_deployc command list and command illustration

1、 Command list:

```

[toybrick@localhost 1808]$ toybrick_deployc -H
Usage: toybrick_deployc [OBJECT] [OPTION] <parameters>...
OBJECT = { set | install | deploy }
Options:
1) toybrick_deployc [OPTION] <parameters>
   -V,--version
   -H,--help
   -U,--update           Update system
   -P,--password password Change deploy password, default: toybrick

2) toybrick_deployc install [OPTION] <parameters>
   -r,--rpm packages    Install rpm packages
   -p,--python packages Install python packages

3) toybrick_deployc deploy [OPTION] <parameters>
   -s,--sync directory Sync directory to target compute stick
   -r,--run             Execute command: sudo -u <user> <directory>/boot.sh
   -b,--boot on|off    Enable or disable startup
   e.g.
       toybrick_deployc deploy -s dir -r -b on

4) toybrick_deployc set ip <IP> port <PORT> timeout <x secs>
   default:
       ip:          192.168.180.8
       port:        18080
       timeout:     60
   e.g.
       toybrick_deployc set ip 192.168.180.8 port 18080 timeout 60
   Note:
       These values will set in /home/toybrick/.deploy.conf
       Nerver timeout, if timeout is -1

```

2、Command illustration:

● Default object: null

- 1) `toybrick_deployc -V`: check `toybrick_deploy`'s version number;
- 2) `toybrick_deployc -H`: check command help information;
- 3) `toybrick_deployc -U`: Execute `dnf` update, upgrade computing stick's system software package ;
- 4) `toybrick_deployc -P`: revise `toybrick_deploy`'s login password, default one is "toybrick"

Notice: when executing `toybrick_deployc` command, it is required to input login password, forgetting password will cause that RK1808 AI compute stick cannot be used!

● Install software package

- 1) `toybrick_deployc install -r xxxx`: inform computing stick to install specific rpm

software package, computing stick root user execute command: `dnf install xxxx`;

- 2) `toybrick_deployc install -p xxxx`: inform computing stick to install specific python package, computing stick toybrick user execute command: `pip3 install --user xxxx`

- **deploy**

- 1) `toybrick_deployc deploy -s dir`: synchronize dir catalog to computing stick's application working catalog
- 2) `toybrick_deployc deploy -r`: after successful synchronization, execute `boot.sh` under catalog;
- 3) `toybrick_deployc deploy -b`: set `boot.sh` as startup;

- **set environment variable**

`toybrick_deployc set ip xxx.xxx.xxx.xxx port yyy timeout zzz`

- 1) `xxx.xxx.xxx.xxx`: set RK1808 AI compute stick's IP address, the default is 192.168.180.8;
- 2) `yyy`: set RK1808 AI compute stick's service port, the default is: 18080;
- 1) Set `toybrick_deployc`'s connection overtime: the default is 60s

Illustration: `-l` means waiting until CTRL+C force exit or receiving RK1808 AI compute stick's reply

- **Prompt illustration:**

- 1) Password for deploy: prompt users input `toybrick_deploy`'s login password
- 2) Local host's username: prompt users input the Host system's user name
- 3) Password for user[username]: prompt users input name according to system's user name
- 4) [INFO]: means command which return from service side executing detail

-
- 5) Command XXXXX done: means command returns successfully
 - 6) Command xxxxx error: means command executes wrong

7.2.2 Development process based on Toybrick platform

1. RK1808 AI compute stick is taken as network interface card device(rndis), configured as master mode through web configuration, for more information please see web configuration introduction(If it is needed to recover slave mode, please reset to slave mode).
2. Set toybrick_deploy login password.
3. Upgrade the system software packages in RK1808 AI computing stick through host mode development tool toybrick_deploy, to make sure the system software packages in RK1808 AI computing stick are in the latest version. Toybrick does maintenance for its own source server, and it would update system software package in a unscheduled way to increase system stability. For this step, user should make their own choice whether it is updated. When current system satisfies development requirement, it does not need to update.
4. RK1808 AI computing stick communicate with host machine through network socket, programming host machine service program and RK1808 AI computing stick lower machine program, through host mode development tool toybrick_deploy install rpm and python dependant package in RK1808 AI compute stick (if it is needed),solidify model and running program in RK1808 AI compute stick, and execute inference and related processing program in RK1808 AI compute stick, host machine receives the inference and related data processing result in RK1808 AI compute stick.
5. Set the program solidified in RK1808 AI compute stick starting after turning on computer through host mode development tool toybrick_deploy, and restart RK1808 AI compute stick, verify if it is configured successfully, do preparation for deployment.

7.2.3 Introduction of product deployment

1. Refer to the development process in Section 7.1.2 to solidify the model and algorithm into the RK1808 AI compute stick, and set the program start up.
2. Plug RK1808 AI compute stick into the target device, such as webcam equipment, PC, drone, smart car, etc.
3. Running the Host service program in target device, and display processing result.
4. You can find the Yolov3 master mode demo in wiki.

More information about master mode development please log in the official forum:

<http://t.rock-chips.com>