

PUSR



Linux Development Manual

Internet of Things (IoT) Controller

SH800

Content

1. SH800 Hardware Interfaces	3
2. General Interfaces	3
2.1. Cellular Network (USR-SH800-EW version don't support)	4
2.2. SIM card ICCID Query	5
2.3. WIFI	5
2.4. GPS	6
2.5. SSH Function	8
2.6. WukongEdge Application	9
2.7. WukongEdge Application Deactivation	10
3. Secondary Development	10
3.1 Read Data	11
3.2 Write Data	11
4. Ubuntu System	12
4.1 Common Command	12
4.2 Linux GCC Download the compiler	12
4.3 Ubuntu Version Upgrade	12
4.4 Query the version of Linux system	13
4.5 Linux Modify hostname	13
4.6 Switch the network card communication	13
4.7 Modify system language	13
4.8 Use the Systemd service startup file	14
4.9 Uninstallation and installation of xubuntu desktop	14
5. Application Installation	15
5.1 Docker Container	15
5.2 Install OpenPLC Runtime	16
5.3 Install Todesk	16

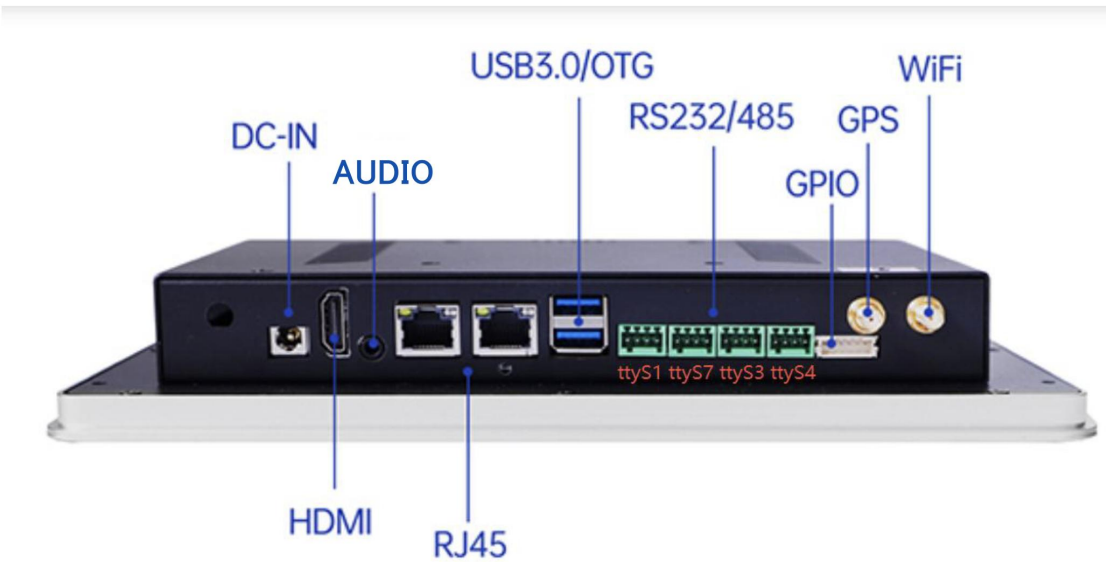
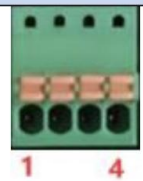
1. SH800 Hardware Interfaces

Description of serial port driver identification:

Interface	Driver Identifier
RS485-1	ttyS1
RS485-2	ttyS7
RS232-1	ttyS3
RS232-2	ttyS4

Details of the RS485/RS232 connector: the interface pin spacing is 2.0MM

NO.	Definition	Attribute	Description
1	3.3V	Output	3.3V voltage output
2	TX/A	Output	Transmit (TX/A)
3	RX/B	Input	Receive (RX/B)
4	GND	Ground	Ground



A demo for serial port transmission and reception is provided under the USR-EG828 directory: EG828_Uart_Rev.c. and EG828_Uart_Send.c.

2. General Interfaces

2.1. Cellular Network (USR-SH800-EW version don't support)

Cellular networks include 4G and 5G. After connecting to the carrier network, data is transmitted via the carrier's public or private network. Cellular networks connect automatically by default. Use the following commands for verification:

Note: If using a carrier-specific SIM card (private network), configure APN parameters manually.

Command	Interface
ifconfig	Wwan OR usb0
ip addr	Wwan OR usb0

```
root@linux:~/test# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
3: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether f2:31:cf:61:47:e6 brd ff:ff:ff:ff:ff:ff
4: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether ee:31:cf:61:47:e6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.35/24 brd 192.168.10.255 scope global dynamic noprefixroute eth1
        valid_lft 61585sec preferred_lft 61585sec
    inet6 fe80::8e74:3458:ee5f:93b1/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
5: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether e8:51:9e:cb:1f:0b brd ff:ff:ff:ff:ff:ff
6: usb0: <BROADCAST,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether ca:2e:1c:f1:21:83 brd ff:ff:ff:ff:ff:ff
    inet 10.27.135.218/30 brd 10.27.135.219 scope global dynamic noprefixroute usb0
        valid_lft 3992sec preferred_lft 3992sec
    inet6 fe80::bbbd:4ed7:2a2c:a55c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
root@linux:~/test#
```

```

eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether f2:31:cf:61:47:e6 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 34

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.35 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::8e74:3458:ee5f:93b1 prefixlen 64 scopeid 0x20<link>
    ether ee:31:cf:61:47:e6 txqueuelen 1000 (Ethernet)
    RX packets 131806 bytes 8723759 (8.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 101714 bytes 12370253 (12.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 46

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 674 bytes 66492 (66.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 674 bytes 66492 (66.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

usb0: flags=4291<UP,BROADCAST,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.27.135.218 netmask 255.255.255.252 broadcast 10.27.135.219
    inet6 fe80::bbbd:4ed7:2a2c:a55c prefixlen 64 scopeid 0x20<link>
    ether ca:2e:1c:f1:21:83 txqueuelen 1000 (Ethernet)
    RX packets 514 bytes 72630 (72.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1915 bytes 302665 (302.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether e8:51:9e:cb:1f:0b txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@linux:~/test#

```

2.2. SIM card ICCID Query

The ICCID is the SIM card number used for top-up. For the secondary development, you can query the ICCID using the following commands:

Command	Function
cat /dev/ttyUSB2 &	Enable echo
echo -e "at+qdsim=1\r\n" > /dev/ttyUSB2	Switch SIM card, 0 as card 1, 1 as card 2
echo -e "at+qccid\r\n" > /dev/ttyUSB2	Query ICCID of SIM card

2.3. WIFI

SH800 supports STA mode. Connect to an AP using the following commands:

Command	Function
nmcli dev wifi list	Scan WiFi AP
nmcli --ask dev wifi connect <SSID> password <password>	Connect to a WiFi AP
ifconfig	Query network status (WLAN0)
nmcli device disconnect wlan0	Disconnect WiFi
nmcli connection delete id <SSID>	Clear WiFi configuration

```

28:9C:6E:9C:2B:8A  HN_EP05160386093PBKME-00831  Infra 1    65 Mbit/s
35  ** WPA2
48:98:CA:63:78:E9  EZVIZ_C44642345  Infra 3    65 Mbit/s
32  ** WPA2
root@linux:~# nmcli --ask dev wifi connect USR-AP password www.usr.cn
Device 'wlan0' successfully activated with '7f72c85e-b4f3-43b6-8893-e84cddb3d09'.
root@linux:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether f2:31:cf:61:47:e6 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 34

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.35 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::f861:85ca:2448:2b70 prefixlen 64 scopeid 0x20<link>
    ether ee:31:cf:61:47:e6 txqueuelen 1000 (Ethernet)
    RX packets 765 bytes 68622 (68.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 871 bytes 124170 (124.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 46

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 947 bytes 84427 (84.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 947 bytes 84427 (84.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.11.47 netmask 255.255.254.0 broadcast 192.168.11.255
    inet6 fe80::f07f:9100:54fc:c118 prefixlen 64 scopeid 0x20<link>
    ether e8:51:9e:cb:1f:0b txqueuelen 1000 (Ethernet)
    RX packets 108 bytes 24189 (24.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 38 bytes 9423 (9.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@linux:~#

```

2.4. GPS

USR-SH800 supports GPS feature. Obtain GPS data as follows:

Command	Function
apt-get install gpsd gpsd-clients	Install GPSD Client
vim /etc/default/gpsd	Modify GPS Interface
echo -ne "at+qgps=1\r\n" > /dev/ttyUSB2	Enable GPS function
cgps -s	View positioning data

Steps:

1. Install GPSD Client: `apt-get install gpsd gpsd-clients`
2. Modify GPS interface:

Open the **gpsd** file by running **vim /etc/default/gpsd**. Then, press **i** to enter the insert mode. After that, change the information interface to USB1. The specific details are shown in the figure below.

```
# Devices gpsd should collect to at boot time.
# They need to be read/writeable, either by user gpsd or the group dialout.
DEVICES="/dev/ttyUSB0"

# Other options you want to pass to gpsd
GPSD_OPTIONS=""

~
~
~
~
~
```

3. After the modification is completed, hold down **Ctrl + X** to save (It's best to press it twice) Then, press the **esc** key to exit the input mode. Enter **:wq** to save the file and return to the command interface.

```
# Devices gpsd should collect to at boot time.
# They need to be read/writable, either by user gpsd or the group dialout.
DEVICES="/dev/ttyUSB1"

# Other options you want to pass to gpsd
GPSD_OPTIONS=""

:~$
```

- Before turning on the GPS, first ensure that the command channel is clear. Execute the command **cat /dev/ttyUSB2** to check if the channel is working properly. If you receive a command echo, it means the channel is normal, as shown in the figure below. Then, exit to the command mode.
- Execute the command **echo -ne "at+qgps=1\r\n" > /dev/ttyUSB2** to turn on the GPS function
- Execute the command **cgps -s** to enter the GPS information display interface. After waiting for a while, the GPS positioning information will appear.

Time: 2024-06-21T00:42:11.000Z		Seen 16/Used 21				
Latitude:	36.66561207 N	PRN	Elev	Azim	SNR	Use
Longitude:	117.09933975 E	GP 2	34.0	45.0	42.0	Y
Alt (HAE, MSL):	350.722, 363.845 ft	GP 21	17.0	49.0	28.0	Y
Speed:	0.00 mph	GP 3	43.0	108.0	18.0	N
Track (true, var):	0.0, -5.4 deg	GP 6	25.0	230.0	23.0	N
Climb:	1003.94 ft/min	GP 7	1.0	184.0	31.0	N
Status:	3D FIX (29 secs)	GP 8	0.0	0.0	0.0	N
Long Err (XDOP, EPX):	n/a, n/a	GP 14	82.0	229.0	17.0	N
Lat Err (YDOP, EPY):	n/a, n/a	GP 17	55.0	317.0	30.0	N
Alt Err (VDOP, EPV):	0.90, +/- 67.9 ft	GP 19	32.0	291.0	24.0	N
2D Err (HDOP, CEP):	1.10, +/- 68.6 ft	GP 22	63.0	309.0	22.0	N
3D Err (PDOP, SEP):	1.50, +/- 93.5 ft	GP 24	1.0	322.0	0.0	N
Time Err (TDOP):	n/a	GP 30	18.0	210.0	0.0	N
Geo Err (GDOP):	n/a	SB 39	0.0	0.0	34.0	N
ECEF X, VX:	n/a, n/a	SB 41	0.0	0.0	34.0	N
ECEF Y, VY:	n/a, n/a	SB 46	0.0	0.0	34.0	N
ECEF Z, VZ:	n/a, n/a	SB 50	0.0	0.0	34.0	N
Speed Err (EPS):	n/a					
Track Err (EPD):	n/a					
Time offset:	0.014 sec					
Grid Square:	OM86np					

7. You can also view the raw GPS data by executing the command **cat /dev/ttyUSB1**. Additionally, you can obtain and process the data through the USB1 interface driver.

```
$GPVTG,0.0,T,5.4,M,0.0,N,0.0,K,A*22
$GPRMC,004110.00,A,3639.954385,N,11705.959783,E,0.0,0.0,210624,5.4,W,A,V*59
$GPGSA,A,3,02,21,,,,,,,,,9.7,9.6,0.9,1*26
$GPGSV,5,1,20,02,34,045,41,03,43,108,19,06,25,230,23,07,01,184,28,1*66
$GPGSV,5,2,20,14,82,229,15,17,55,317,27,21,17,049,29,22,63,309,27,1*6C
$GPGSV,5,3,20,30,18,210,24,08,00,092,,19,32,291,,24,01,322,,1*68
$GPGSV,5,4,20,33,,34,38,,34,40,,34,42,,34,1*6E
$GPGSV,5,5,20,46,,34,48,,34,49,,34,50,,34,1*60
$GPGGA,004111.00,3639.953607,N,11705.959742,E,1,02,9.6,105.9,M,-4.0,M,,*77
$GPVTG,0.0,T,5.4,M,0.0,N,0.0,K,A*22
$GPRMC,004111.00,A,3639.953607,N,11705.959742,E,0.0,0.0,210624,5.4,W,A,V*5D
$GPGSA,A,3,02,21,,,,,,,,,9.7,9.6,0.9,1*26
$GPGSV,5,1,20,02,34,045,41,03,43,108,19,06,25,230,23,07,01,184,27,1*69
$GPGSV,5,2,20,14,82,229,15,17,55,317,28,21,17,049,29,22,63,309,28,1*6C
$GPGSV,5,3,20,30,18,210,23,08,00,092,,19,32,291,,24,01,322,,1*6C
$GPGSV,5,4,20,33,,34,38,,34,40,,34,42,,34,1*6E
$GPGSV,5,5,20,46,,34,48,,34,49,,34,50,,34,1*60
$GPGGA,004112.00,3639.952966,N,11705.959730,E,1,02,9.6,105.9,M,-4.0,M,,*78
$GPVTG,0.0,T,5.4,M,0.0,N,0.0,K,A*22
$GPRMC,004112.00,A,3639.952966,N,11705.959730,E,0.0,0.0,210624,5.4,W,A,V*52
$GPGSA,A,3,02,21,,,,,,,,,9.7,9.6,0.9,1*26
```

2.5. SSH Function

The USR-SH800 has the SSH function enabled and root privileges opened. The username is root, and the default password is 123456. After logging in with the root account, it is recommended to change the root user password promptly to ensure system security. The instructions for changing the password are as follows:

Command	Interface
passwd	After execution, you will be prompted to enter a new password. Once the new password is entered consistently twice, the new password will be successfully modified.
reboot	Restart the device

```

root@EG628:~/test# passwd
New password:
Retype new password:
passwd: password updated successfully
root@EG628:~/test# passwd
New password:
Retype new password:
passwd: password updated successfully
root@EG628:~/test# reboot

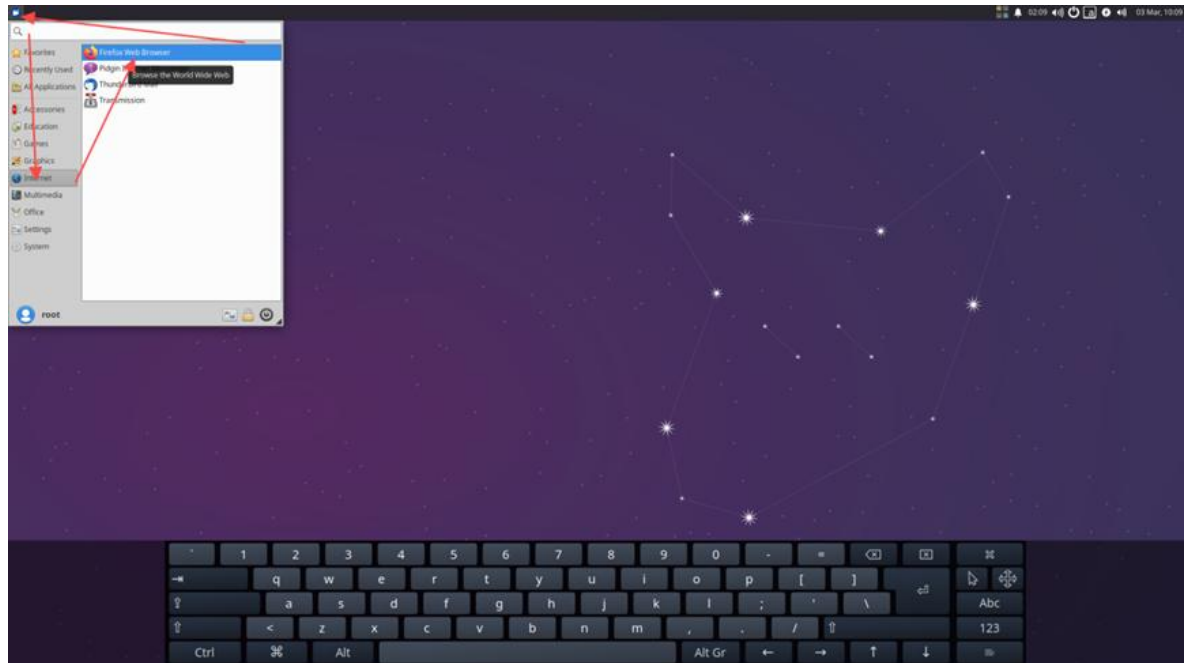
```

2.6. WukongEdge Application

WukongEdge is a 4-i -1 intelligent data-processing architecture integrating edge computing, network management, local configuration, and PLC programming. It's easy to use, with a parameter-config webpage. Users log in to the WEB interface to use it.

There are two login methods:

1. **Desktop login:** For EG628 (with a desktop system), connect a mouse, keyboard, and HDMI screen. Open a browser on the desktop, enter the default LAN IP (192.168.1.1) to log in. If no browser, open it as follows.



2. **External PC login:** Connect PC's network port to the device's LAN port. Set PC to auto-obtain IP. Open a browser on PC, enter the device's LAN IP to log in to the WEB for parameter configuration.

For function details, check the WukongEdge manual:

<https://www.pusr.com/support/download/User-Manual-WukongEdge-V1.html>

2.7. WukongEdge Application Deactivation

When doing secondary development, if you don't need the WukongEdge function, you can turn it off via commands. You can execute commands through SSH or on the desktop. Just follow the commands below.

Command	Interface
gateway_tool switch 0	Disable WukongEdge
gateway_tool switch 1	Enable WukongEdge
reboot	Reboot Device

3. Secondary Development

EG devices support Linux-based secondary development and have the WukongEdge edge-computing function built-in. Scenarios where the two are used in combination are common.

Therefore, WukongEdge provides API interfaces for secondary development to call edge data, making it convenient for customers to integrate WukongEdge with Linux application development.

3.1 Read Data

Read-only Interface Function for Edge Computing Function:

Read the data collected in the edge computing data point table.

```
Function declaration: BOOL edge_read(edge_rw_msg_t *edge_operate_msg,  
edge_access_multi_node_t *edge_response_msg);
```

Parameters:

edge_operate_msg: Query data information

edge_response_msg: Returned result

Information related to the position point:

edge_rw_msg_t structure parameters

client_name: Task name, can be customized but not repeatable

server_name: read the channel name, the default reading is edge_read, can't be modified

3.2 Write Data

Write Data Read&write Interface Function for Edge Computing Function:

Write data into the edge computing data point table to achieve data- sending control.

```
BOOL edge_read_write(edge_rw_msg_t *edge_operate_msg, edge_access_multi_node_t  
*edge_response_msg);
```

Parameters:

edge_operate_msg: Query data information

edge_response_msg: Returned result

Information related to the position point:

edge_rw_msg_t structure parameters

client_name: Task name, can be customized but not repeatable

server_name: read the channel name, the default reading is edge_read, can't be modified

You can refer to the "Data Reading and Writing" demo for the specific parameters when using and testing.

4. Ubuntu System

4.1 Common Command

Command	Interface
cd /Path	Switch directory
rm File name	Delete a file
Mkdir File name	Create a new folder
Touch File name	Create a new file
df -h	Check flash space

4.2 Linux GCC Download the compiler

Command	Interface
apt-get update	Update the Linux software
apt-get install gcc	Download and install the GCC compiler
gcc hello.c -o hello	Compile the hello.c file
gcc --version	Check the GCC version

* If you use a cross - compiler, you need to download it from a specific website.

4.3 Ubuntu Version Upgrade

The commonly used commands are as follows: (If you don't have root privileges, you need to add the `sudo` command.)

Command	Function
apt-get update	Update the software sources
apt-get upgrade	Update the installed software packages
apt-get dist-upgrade	Handle the dependencies
apt-get install update-manager-core	Install the upgrade tool
do-release-upgrade	Perform the upgrade
lsb_release -a	Check the version

4.4 Query the version of Linux system

Command	Feature
lsb_release -a	Display information such as the Ubuntu distribution ID, description, version number, and code.
uname -a	Show all system information, including the kernel version and system architecture.
hostnamectl	Display the static, dynamic, and transient hostname settings of the system

4.5 Linux Modify hostname

Command	Function
hostnamectl set-hostname <hostname>	<hostname>Delete the current routing interface for the new desired hostname.

4.6 Switch the network card communication

Command	Function
ip route del default	Delete the current routing interface.
ip route add default via <eth1_gateway_ip> dev eth1	Add eth1 as the default routing interface. The eth1_gateway_ip is the gateway address of the eth1 network card, which can be found via `ip route show`
ping ip_addr	Check the network by pinging an address
apt-get update	Update the package sources
apt install ifmetric	Install ifmetric
ifmetric eth1 200	

4.7 Modify system language

Change the system language as English

Command	Function
apt update	Update the package sources
apt install language-pack-en	Install the English language pack.
update-locale LANG=en_US.UTF-8	Update the system language environment to English
Reboot	Restart the system for the changes to take effect.

4.8 Use the Systemd service startup file

Command	Function
<code>gcc test.c -o /usr/local/bin/test</code>	Complete the writing of test.c and finish the compilation, then store it in a specific path
<code>vim /etc/systemd/system/test.service</code>	Create a new test.service service and open the file
<code>[Unit]</code> <code>Description=My custom test application</code> <code>[Service]</code> <code>ExecStart=/usr/local/bin/test</code> <code>[Install]</code> <code>WantedBy=multi-user.target</code>	In the test.service service, add content and then save the file.
<code>sudo systemctl enable test.service</code> <code>sudo systemctl start test.service</code>	Enable and start the service
<code>systemctl list-units --type=service</code>	Check the status of all services
<code>systemctl list-units --type=service --state=running</code>	Check all the running services.

4.9 Uninstallation and installation of xubuntu desktop

Uninstall command	Function
<code>apt-get remove xubuntu-desktop</code>	Uninstall the current desktop environment
<code>apt-get remove xubuntu*</code>	Delete files related to the current desktop
<code>apt-get autoremove</code>	Delete related dependencies
<code>reboot</code>	Restart the device

Install Command	Function
<code>apt-get update</code>	Update sources
<code>apt-get install Xubuntu-desktop</code>	Install the desktop
<code>reboot</code>	Restart the device

5. Application Installation

5.1 Docker Container

Command	Function
apt-get update	Updated Source
apt-get upgrade	
apt install apt-transport-https ca-certificates curl software-properties-common	Install necessary software packages to allow apt to use repositories via HTTPS.
curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg apt-key add -	Add the domestic Docker repository to your system. Here, we take Alibaba Cloud as an example.
apt-key list	List all the GPG keys added to the system.
add-apt-repository "deb [arch=arm64] https://mirrors.aliyun.com/docker-ce/linux/ubuntu \$(lsb_release -cs) stable"	Add the Docker repository to the APT sources (using the Alibaba Cloud source as an example).
apt-get update	Update the apt package index again
apt-get install docker.io	Install the latest version of Docker CE (Community Edition).
<pre>mkdir -p /etc/docker tee /etc/docker/daemon.json <<-'EOF' { "registry-mirrors": ["https://dnbf7xuh.mirror.aliyuncs.com"], "iptables":false, "ip6tables": false, "ipv6":false } EOF systemctl daemon-reload sudo systemctl restart docker OR mkdir -p /etc/docker tee /etc/docker/daemon.json <<-'EOF' { "registry-mirrors": ["https://docker-0.unsee.tech"], "iptables":false, "ip6tables": false, "ipv6":false } EOF systemctl daemon-reload systemctl restart docker</pre>	Configure Docker to use the Alibaba Cloud image accelerator.
docker -- version	Verify that Docker is installed and running correctly.
docker run hello-world	Test if the container can be run

<pre> root@EG828:~# docker run hello-world Unable to find image 'hello-world:latest' locally latest: Pulling from library/hello-world c9c5fd25a1bd: Pull complete Digest: sha256:bfb0ccc14f13f9ed1ae86abc2b9f11181dc50d779807ed3a3c5e55a6936dbdd5 Status: Downloaded newer image for hello-world:latest Hello from Docker! This message shows that your installation appears to be working correctly. To generate this message, Docker took the following steps: 1. The Docker client contacted the Docker daemon. 2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (arm64v8) 3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading. 4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal. To try something more ambitious, you can run an Ubuntu container with: \$ docker run -it ubuntu bash Share images, automate workflows, and more with a free Docker ID: https://hub.docker.com/ For more examples and ideas, visit: https://docs.docker.com/get-started/ root@EG828:~# docker run nginx </pre>	
<pre> systemctl stop docker.socket systemctl disable docker.socket systemctl stop docker systemctl disable docker </pre>	docker Stop and disable Docker.
<pre> systemctl status docker </pre>	Check docker status
<pre> apt-get remove docker.io </pre>	Uninstall docker
<pre> rm -rf /var/lib/docker rm -rf /var/lib/containerd </pre>	Clean up the residual files.

5.2 Install OpenPLC Runtime

Official tutorial: [1.1 OpenPLC Overview – Autonomy](#)

Command	Function
apt-get update	Update Source
apt-get upgrade	
apt-get install git	Install git
git clone https://github.com/thiagoralves/OpenPLC_v3.git	Install OpenPLC Runtime
CD OpenPLC_v3	Switch the patch to OpenPLC_v3
./install.sh linux	Perform the installation on Linux.

After the installation is complete, log in via a browser at http://local_IP:8080. The default username and password are openplc (login) and openplc (password).

5.3 Install Todesk

Command	Function
apt-get update	Update source
apt-get upgrade -y	Update installed software packages
lscpu	Query the architecture (aarch64 is the arm64 architecture)
wget https://dl.todesk.com/linux/todesk_4.0.3_aarch64.deb	Download the installation package of Todesk with the arm64 architecture

<code>dpkg -i todesk_4.0.3_aarch64.deb</code>	Install Todesk
<code>apt-get install -f</code>	During the installation process, it is prompted that dependencies are missing. Execute this command
<code>todesk</code>	Enable ToDesk
After starting, you can see the running Todesk on the Ubuntu desktop.	
<code>systemctl stop todesk</code>	Stop ToDesk
<code>apt-get remove --purge todesk</code>	Uninstall ToDesk
<code>rm -rf /opt/todesk</code> <code>rm -rf ~/.local/share/todesk</code> <code>rm -rf ~/.config/todesk</code>	Clean up residual configurations
<code>apt-get autoremove</code> <code>apt-get autoclean</code>	Clean up dependencies