UBUNTU OS USER MANUAL

OS

1. OS Version

a) Name: Ubuntu 22.04
b) Kernel Version: linux 5.10

2. OS Login

a) Username: namtso
b) Password: namtso

3. Note

The system is equipped with power-up protection, which requires pressing the Power button to switch on the system.

Processor Set

The RK3588 integrates four high-performance Arm Cortex-A76 CPU cores and four low-power Cortex-A55 CPU cores, along with a built-in high-frequency Mali-G52 GPU and an NPU co-processor.

1. CPU Temperature

a) Chip centre temperature soc-thermal:

```
cat /sys/class/thermal/thermal_zone0/temp
```

b) CPU big core A76_0/1; CPU4 和 CPU5 temp:

```
cat /sys/class/thermal/thermal_zone1/temp
```

c) CPU big core A76_2/3; CPU6 和 CPU7 temp:

```
cat /sys/class/thermal/thermal_zone2/temp
```

d) CPU little core A55_0/1/2/3; CPU0、CPU1、CPU2、CPU3 temp:

```
cat /sys/class/thermal/thermal_zone3/temp
```

a) GPU temp:

```
cat /sys/class/thermal/thermal_zone5/temp
```

b) NPU temp:

```
cat /sys/class/thermal/thermal_zone6/temp
```

2. CPU Point Description

Point	Description
policy0	to set and read CPU little core 0~3
policy4	to set and read CPU big core 4~5
policy6	to set and read CPU big core 6~7

3. CPU Working Mode

a) CPU Mode Description.

Mode	Description
interactive	Runs at maximum frequency, gradually decreases depending on CPU compliance, disadvantage of high power consumption
conservative	Gradual and smooth CPU frequency adjustment, dynamic adjustment at upper and lower frequency limits
ondemand	The CPU switches to the highest frequency when it is performing calculations and drops to the lowest frequency at the end of the calculation.
userspace	Provide API for users to set CPU frequency independently
powersave	CPU fixed at lowest frequency
performance	Fixed operation at maximum frequency
schedutil	The system automatically adjusts the frequency according to the load

b) CPU operational mode reading.

```
cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_governors
cat /sys/devices/system/cpu/cpufreq/policy4/scaling_available_governors
cat /sys/devices/system/cpu/cpufreq/policy6/scaling_available_governors
```

c) CPU operation mode setting.

```
echo "mode" | sudo tee \
/sys/devices/system/cpu/cpufreq/policy0/scaling_governor
echo "mode" | sudo tee \
/sys/devices/system/cpu/cpufreq/policy4/scaling_governor
echo "mode" | sudo tee \
/sys/devices/system/cpu/cpufreq/policy6/scaling_governor
```

4. CPU Operating Frequency

The default CPU working mode is schedutil mode, which does not support frequency setting. To set the frequency, you need to set the CPU working mode to userspace mode first.

a) Get the current CPU supported frequency.

```
cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies
cat /sys/devices/system/cpu/cpufreq/policy4/scaling_available_frequencies
cat /sys/devices/system/cpu/cpufreq/policy6/scaling_available_frequencies
```

b) Set the CPU operating mode to usespace mode.

```
echo userspace | sudo tee \
/sys/devices/system/cpu/cpufreq/policy0/scaling_governor
echo userspace | sudo tee \
/sys/devices/system/cpu/cpufreq/policy4/scaling_governor
echo userspace | sudo tee \
/sys/devices/system/cpu/cpufreq/policy6/scaling_governor
```

c) Setting the CPU frequency.

```
echo xxx | sudo tee /sys/devices/system/cpu/cpufreq/policy0/scaling_setspeed
echo xxx | sudo tee /sys/devices/system/cpu/cpufreq/policy4/scaling_setspeed
echo xxx | sudo tee /sys/devices/system/cpu/cpufreq/policy6/scaling_setspeed
```

d) Check if the setup is successful.

```
cat /sys/devices/system/cpu/cpufreq/policy0/cpuinfo_cur_freq
cat /sys/devices/system/cpu/cpufreq/policy4/cpuinfo_cur_freq
cat /sys/devices/system/cpu/cpufreq/policy6/cpuinfo_cur_freq
```

5. GPU Operating frequency

a) Get the frequency supported by the GPU.

```
cat /sys/class/devfreq/fb000000.gpu/available_frequencies
```

b) Set GPU working mode.

```
echo userspace | sudo tee /sys/class/devfreq/fb000000.gpu/governor
```

c) Setting GPU frequency.

```
echo xxx | sudo tee /sys/class/devfreq/fb000000.gpu/userspace/set_freq
```

d) Check if the setup is successful.

```
cat /sys/class/devfreq/fb000000.gpu/cur_freq
```

6. NPU Operating Frequency

a) Get the frequency supported by the NPU.

```
cat /sys/class/devfreq/fdab0000.npu/available_frequenciess
```

b) Setting the NPU working mode.

```
echo userspace | sudo tee /sys/class/devfreq/fdab0000.npu/governor
```

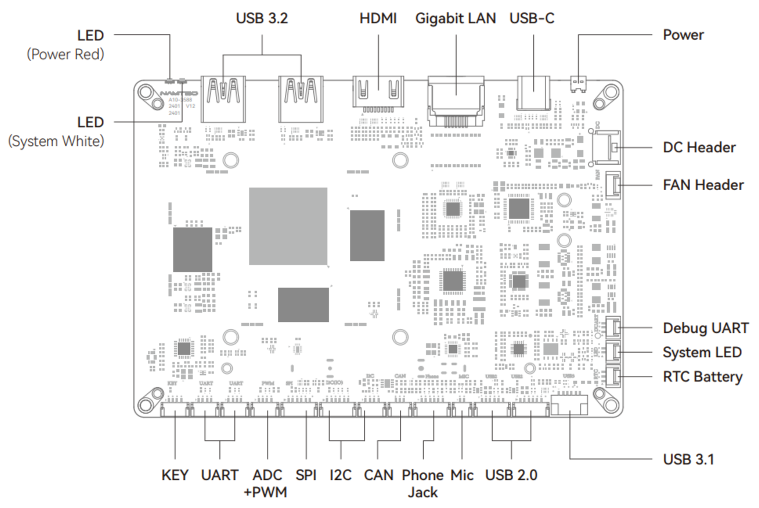
c) Setting NPU frequency.

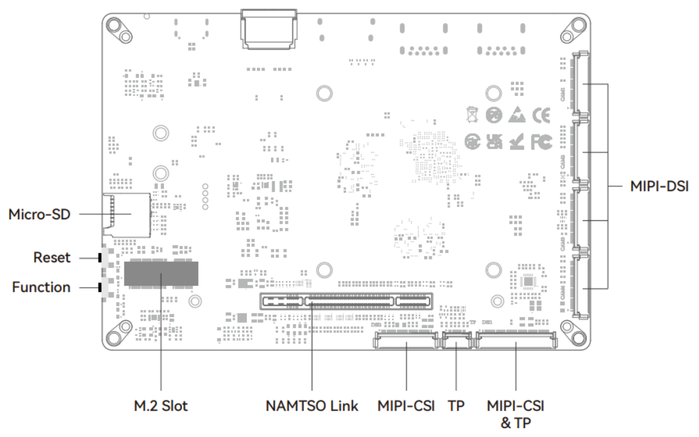
```
echo xxx | sudo tee /sys/class/devfreq/fdab0000.npu/userspace/set_freq
```

d) Check if the setup is successful.

```
cat /sys/class/devfreq/fdab0000.npu/cur_freq
```

Board Level Setup





1. LED Settings

a) Power(Red) LED does not support modification, only System(White) LED supports customization.

b) To query System LED settable status.

```
cat /sys/class/leds/white_led/trigger
```

c) Set to "default-on" as an example.

```
echo default-on | sudo tee /sys/class/leds/white_led/trigger
```

d) Additional notes on the "timer" model.

Timer mode generates two profiles for setting the length of time the LEDs are on and off.

i. Take setting the LED to light up for 100ms as an example:

```
echo 100 | sudo tee /sys/class/leds/white_led/delay_on
```

ii. Take setting the LED off for 900ms as an example:

```
echo 900 | sudo tee /sys/class/leds/white_led/delay_off
```

2. LAN Setting

a) LAN ON

i. Connect the Internet and assign IP automatically.

b) WoL Function Setting

i. Enable Wake-on-LAN function.

```
echo 1 | sudo tee /sys/class/wol/eth0_enable
```

ii. Enable WoL wake-up function and reset system function.

```
echo 3 | sudo tee /sys/class/wol/eth0_enable
```

iii. Disable WoL function.

```
echo 0 | sudo tee /sys/class/wol/eth0_enable
```

3. Button Functional Description

a) Power on and off

i. After powering up the product, press the Power button briefly to switch on the device.

ii. Press and hold the Power button to turn off the power.

b) Device reset

Short press Reset button, system reset directly reboot.

c) Firmware burning mode

Quickly press the Function button 3 times in a row, the device enters the Maskrom burning mode.

4. Wi-Fi & BT Setting

a) Wi-Fi&BT and NAMTSO Link are multiplexed PCIe, default is off, enable method:

i. edit "/boot/uEnv.txt":

```
wifi=off → wifi=on
```

ii. Reboot device

b) Connect Wi-Fi via command:

```
nmcli dev wifi list
nmcli --ask dev wifi connect "SSID" password "Password"
```

c) Connect Bluetooth via command, turn on Bluetooth device:

```
rfkill unblock all
sudo hciconfig hci0 up
```

f) Use Bluetooth control function:

```
bluetoothctl
```

g) Initialize and scan Bluetooth:

```
[bluetooth]# agent on
[bluetooth]# default-agent
[bluetooth]# power on
[bluetooth]# discoverable on
[bluetooth]# pairable on
[bluetooth]# scan on
[device list]
```

h) Connect Bluetooth Device:

```
[bluetooth]# connect [device address]
```

5. FAN Setting

a) Get the current working mode of the fan:

```
fan.sh mode
```

b) Fan operating mode setting:

```
Disable → fan.sh off
Enable → fan.sh on
Auto Mode → fan.sh auto
Manual Mode → fan.sh manual
```

C) Fan speed setting(Only Manual Mode).

i. Maximum speed mode:

```
fan.sh highest
```

ii. High-speed mode:

```
fan.sh high
```

ii. Medium speed mode:

```
fan.sh mid
```

iii. Low speed mode:

```
fan.sh low
```

iii. Minimum speed mode:

```
fan.sh lowest
```

Overlay

1. Introduction of Overlay

Overlay function means to quickly overwrite the DTB and modify the system tree settings through dtbo file without recompiling the system source code. The Overlay function makes it easy to quickly test the reusable functions.

2. Overlay Function

Currently the A10-3588 has two default Overlay functions. PWM as well as SPI respectively, which can be changed to normal GPIOs after startup with Overlay named:

```
PWM: pwm-gpio-overlay.dtbo
SPI: spi-gpio-overlay.dtbo
```

3. Overlay Usage

a) Overlay files specific location:

```
/boot/dtb/rockchip/rk3588-namtso-a10-3588.dtb.overlays/
```

b) View current configurable overlay features:

```
ls /boot/dtb/rockchip/rk3588-namtso-a10-3588.dtb.overlays/
pwm-gpio-overlay.dtbo     spi-gpio-overlay.dtbo
```

c) Overlay function profiles:

```
/boot/dtb/rockchip/rk3588-namtso-a10-3588.dtb.overlay.env
```

e) Take "pwm-gpio-overlay.dtbo" for example, save the edits and reboot the system, reload the DTB to take effect, after enabling this pwm port as a normal gpio use.

```
edit → /boot/dtb/rockchip/rk3588-namtso-a10-3588.dtb.overlay.env
fdt_overlays=pwm-gpio-overlay
```

f) Use "pwm-gpio-overlay.dtbo", save the edits and reboot the system, reload the DTB to take effect, after disabling this port as PWM functionality.

```
edit → /boot/dtb/rockchip/rk3588-namtso-a10-3588.dtb.overlay.env
fdt_overlays
```

g) For custom Overlay function, please refer to the SDK development documentation.

Expansion Header

1. Expansion LED

a) Querying the Expansion LED setting status:

```
cat /sys/class/leds/ext_led/trigger
```

b) Set to "default-on", for example:

```
echo default-on | sudo tee /sys/class/leds/ext_led/trigger
```

c) Supplementary note on the "timer" model.

Timer mode generates two profiles for setting the length of time the LEDs are on and off.

i. Take setting the LED to light up for 100ms as an example:

```
echo 100 | sudo tee /sys/class/leds/ext_led/delay_on
```

ii. Take setting the LED off for 900ms as an example:

```
echo 900 | sudo tee /sys/class/leds/ext_led /delay_off
```

2. CAN

a) Open CAN:

```
sudo ip link set can0 up
```

b) Close CAN:

```
sudo ip link set can0 down
```

c) View CAN configuration information:

```
sudo ifconfig -a | grep can
```

d) Set the baud rate:

```
sudo ip link set can0 type can bitrate 250000
```

e) Receive CAN messages:

```
candump can0
```

f) Send messages:

```
cansend can0 123#1122334455667788
```

3. I2C

If "/dev/i2c-2" and "/dev/i2c-4" exist, you can use i2c-tools to manipulate this I2C interface.

1) List all available I2C BUS:

```
i2cdetect -l
```

2) Retrieve devices on I2C4:

```
i2cdetect -y -r 4
```

3) Read the device connected to the I2C4:

```
i2cget -f -y 4 0x1d 0x0d
```

Its device address is "0x1d" and its register address is "0x0d".

4) Set the device connected to the I2C4:

```
i2cset -f -y 4 0x1d 0x0d 0x02
```

Its device address is "0x1d" and register address is "0x0d", modify it to "0x02".

4. SPI

Just use "ls /dev/spidev3.0" to confirm that the SPI BUS is turned on.

5. UART

1. Device Point

The left serial port is UART0, the right serial port is UART1.

```
/dev/ttyWCH0
/dev/ttyWCH1
```

2. Baud rate setting

Take UART0 as an example:

```
stty -F /dev/ttyWCH0 ispeed 115200 ospeed 115200 cs8
stty -F /dev/ttyWCH0 speed 115200 cs8 -parenb -cstopb -echo
```

3. Send Data

```
echo “aaa” | sudo tee /dev/ttyWCH0
```

4. Read Data

```
cat /dev/ttyWCH0
```

6. Power KEY

Same function as Power Button for external expansion of power buttons.

Encoding and Decoding

1. Encoding

a) Check the supported encoding formats

```
gst-inspect-1.0 | grep mpp
rockchipmpp:   mpph264enc: Rockchip Mpp H264 Encoder
rockchipmpp:   mpph265enc: Rockchip Mpp H265 Encoder
rockchipmpp:   mppjpegdec: Rockchip's MPP JPEG image decoder
rockchipmpp:   mppjpegenc: Rockchip Mpp JPEG Encoder
rockchipmpp:   mppvideodec: Rockchip's MPP video decoder
rockchipmpp:   mppvp8enc: Rockchip Mpp VP8 Encoder
typefindfunctions: audio/x-musepack: mpc, mpp, mp+
```

b) Encoding Mode NV12 to H264

```
gst-launch-1.0 -v filesrc location=./test.yuv ! videoparse width=1920 height=1080
format=nv12 ! mpph264enc ! h264parse ! queue ! filesink location=./test.h264
```

c) Encoding USB camera

```
gst-launch-1.0 v4l2src device=/dev/video80 io-mode=mmap num-buffers=300 !
image/jpeg, width=1920, height=1080, framerate=30/1 ! mppjpegdec !
mpph264enc ! filesink location=./test.h264
```

2. Decoding

a) Check supported decoding formats

```
gst-inspect-1.0 | grep mpp
rockchipmpp:   mpph264enc: Rockchip Mpp H264 Encoder
rockchipmpp:   mpph265enc: Rockchip Mpp H265 Encoder
rockchipmpp:   mppjpegdec: Rockchip's MPP JPEG image decoder
rockchipmpp:   mppjpegenc: Rockchip Mpp JPEG Encoder
rockchipmpp:   mppvideodec: Rockchip's MPP video decoder
rockchipmpp:   mppvp8enc: Rockchip Mpp VP8 Encoder
rockchipmpp:   mppvpxalphadecodebin: VP8/VP9 Alpha Decoder
typefindfunctions: audio/x-musepack: mpc, mpp, mp+
```

b) Decode MP4 files as an example

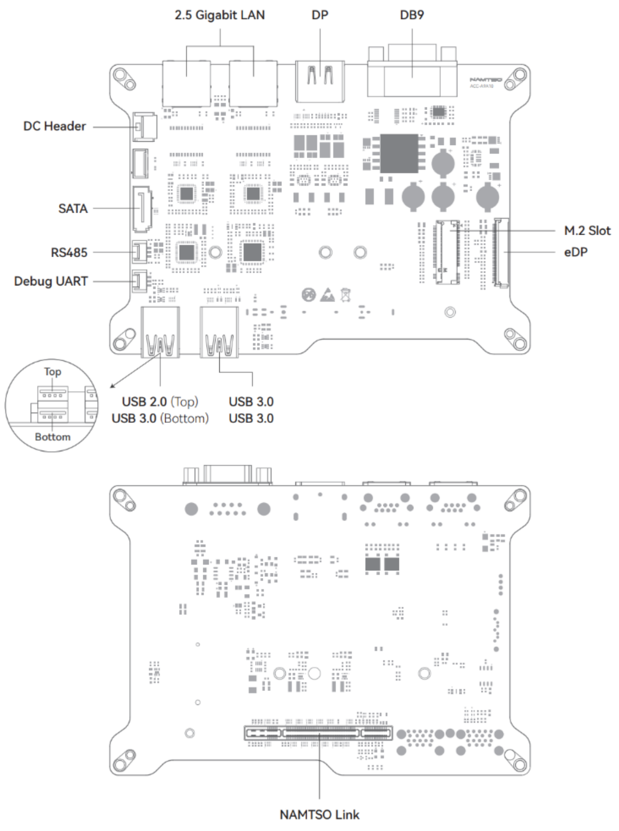
```
gst-launch-1.0 filesrc location=./test.mp4 ! qtdemux name=d d.video_0 ! h264parse !
mppvideodec ! video/x-raw,format=NV12 ! filesink location=test.yuv
```

c) Decode H.264 files

```
gst-launch-1.0 filesrc location=./test.h264 ! h264parse ! mppvideodec ! video/x-
raw,format=NV12 ! filesink location=test.yuv
```

Accessories

1. Expansion Board A9A10



a) RS485

Refer to the UART section of the motherboard for usage. The device node is:

```
/dev/ttyWCH2
```

b) SATA

i. Formatting

```
sudo mkfs.ext4 /dev/sdxv
```

ii. Take the example of mounting to the mnt directory

```
sudo mount /dev/sdxv /mnt
```

c) 2.5 Gigabits LAN

i. LAN1 Point is eth1, LAN2 Point is eth2

ii. Refer to the motherboard LAN section for usage

d) RS232

Refer to the UART section of the motherboard for usage. The device node is:

```
/dev/ttyWCH3
```

e) M.2 Slot

i. Formatting

```
sudo mkfs.ext4 /dev/nvmexny
```

ii. Take the example of mounting to the mnt directory

```
sudo mount /dev/nvmexny /mnt
```