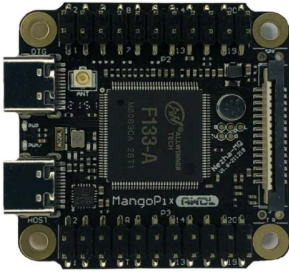


RT-Smart Setup

Setting up the RT-Smart micro-kernel on MangoPI RISC-V architecture (MANGOPI-NEZHA-MQ-01) using WSL 1 on Microsoft Windows 11.

HARDWARE

MANGOPI-NEZHA-MQ-01



Link:

https://www.mouser.com/ProductDetail/RT-Thread/MANGOPI-NEZHA-MQ-01?qs=Znm5pLBrCALHG%252BSjHvKpgq%3D%3D&mgh=1&gclid=Cj0KCQiA3eGfBhCeARIsACpJNU-lkpseEfvPrBpbanjVO611jCWP56vp-OipaSRnkwdCpEID-urCiRgaAmDPEALw_wcB

USB-TO-SERIAL



Link: <https://a.co/d/a2sa2Co>

USB C-TYPE



Link: <https://a.co/d/4ALR3CK>

SOFTWARE

Windows 11 (Home Edition)

Windows Subsystem for Linux Version 1

Ubuntu 18.0.4

RT-SMART INSTALLATION

Open the ubuntu command prompt and issue the following commands to git checkout

```
# git clone https://github.com/RT-Thread/userapps.git
```

```
# cd userapps
```

```
# git clone https://github.com/RT-Thread/rt-thread.git
```

To set the environment,

```
#source smart-env.bat riscv64
```

```
root@LAPTOP-J9VEUE58:/home/stanley/userapps# ./smart-env.sh riscv64
Arch      => riscv64
CC        => gcc
PREFIX    => riscv64-unknown-linux-musl-
EXEC_PATH => /home/stanley/userapps/tools/gnu_gcc/riscv64-linux-musleabi_for_x86_64-pc-linux-gnu/bin
root@LAPTOP-J9VEUE58:/home/stanley/userapps#
```

To download the respective tool chain change working directory to userapps/tools

```
#python3 get_toolchains.py riscv64
```

OR you can manually download it by following below steps

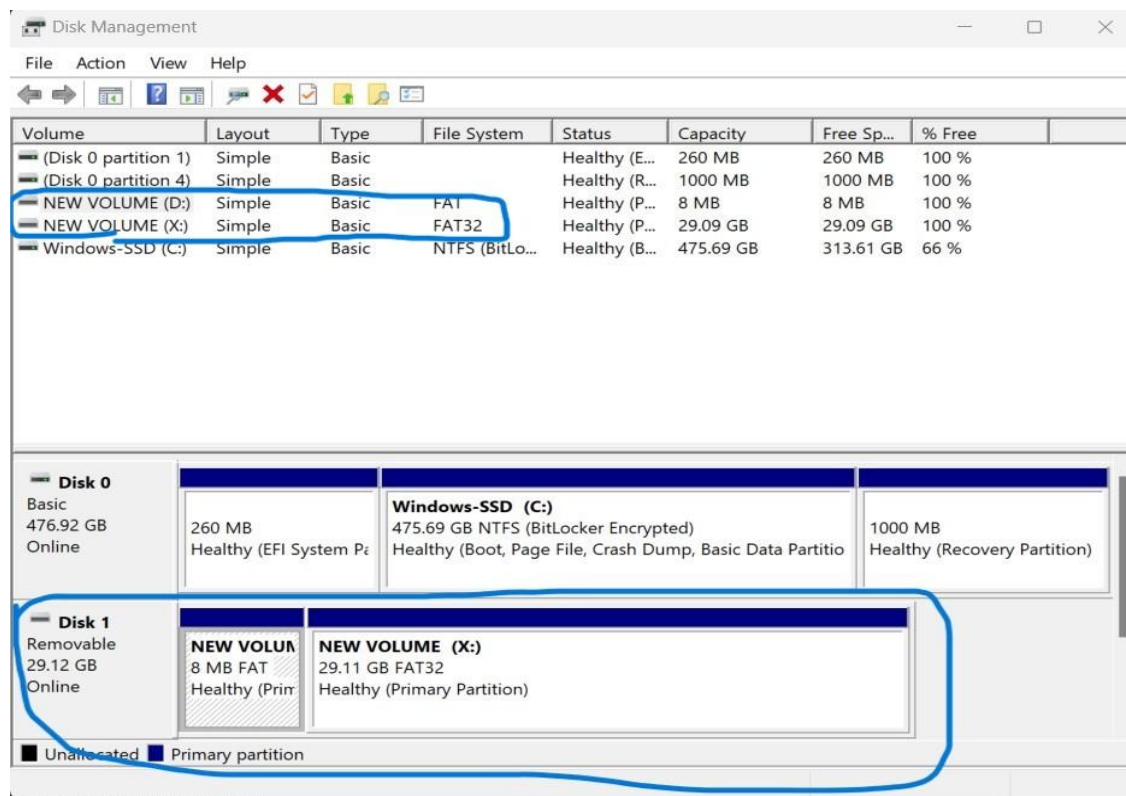
```
#wget
```

```
https://github.com/RT-Thread/toolchains-ci/releases/download/v1.7/riscv64-linux-musleabi\_for\_x86\_64-pc-linux-gnu\_latest.tar.bz2
```

```
#tar xjf riscv64-linux-musleabi_for_x86_64-pc-linux-gnu_latest.tar.bz2 -C /opt
```

SD Card Partition

Format the new SSD card as in below screenshot using Microsoft disk management tool or other 3rd party tool.



Compilation Guide

To compile the image, change the working directory to `/root/rt-thread/bsp/allwinner/d1s/`

`#scons`

Note: if there is any error, issue the following command to see if it does resolve the issue

`#scons --menuconfig`

Do not change anything but just exit and save it. Re-run the following command again.

`#scons`

During the compilation process, ignore the warning messages. At the end of the process there will be a `sd.bin` at the current working directory as below screenshot.

```
root@LAPTOP-J9VEUE58:~/rt-thread/bsp/allwinner/d1s# ls
Kconfig          SConstruct      board           link_stacksize.lds  ports           rtthread.bin    sd.bin
README-M7.md    SConstruct      build           mkimage             rtconfig.h      rtthread.elf    toc1.cfg
README-MQ.md    __pycache__     figures         mkcardcard.sh       rtconfig.py     rtthread.map    tools
README.md        applications    link.lds        mkcardcard.sh       rtconfig.py.bk  sbi.bin         u-boot.dtb
root@LAPTOP-J9VEUE58:~/rt-thread/bsp/allwinner/d1s#
```

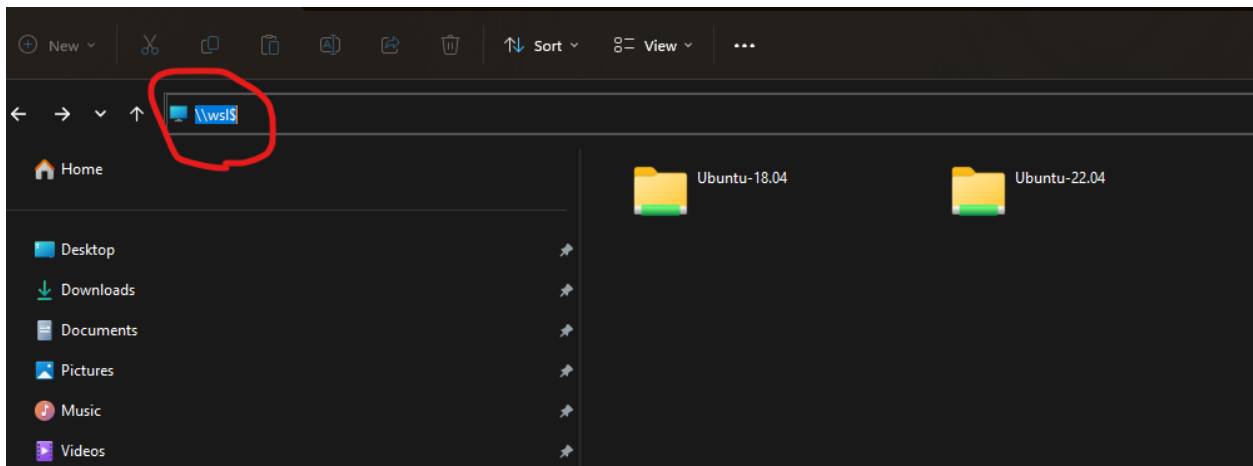
Transfer rt-smart images to SD card

Transfer the following files to your host windows machine before burning to SD card,

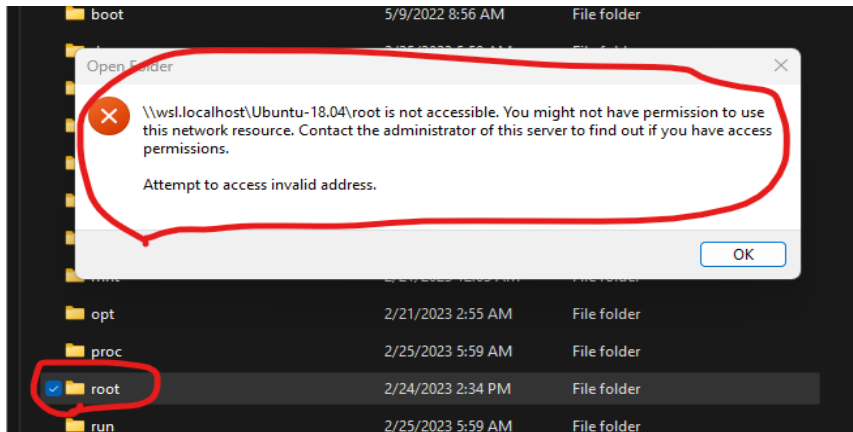
`/root/rt-thread/bsp/allwinner/d1s/sd.bin`

`/root/rt-thread/bsp/allwinner/d1s/tools/boot0_sdcard_sun20iw1p1_d1s.bin`

In order to transfer the above 2 files, open a folder and type `\\wsl$` in the folder address bar. Select Ubuntu-18.04 if you have multiple versions running on your machine. Follow the path and copy and paste the files to your local machine desktop.



Note: if there is permission issue opening root folder as below screenshot,



Follow the below steps in Ubuntu's command prompt by changing the current working directory to `/root/rt-thread/bsp/allwinner/d1s/`.

```
#pwd (double check the current working directory)
```

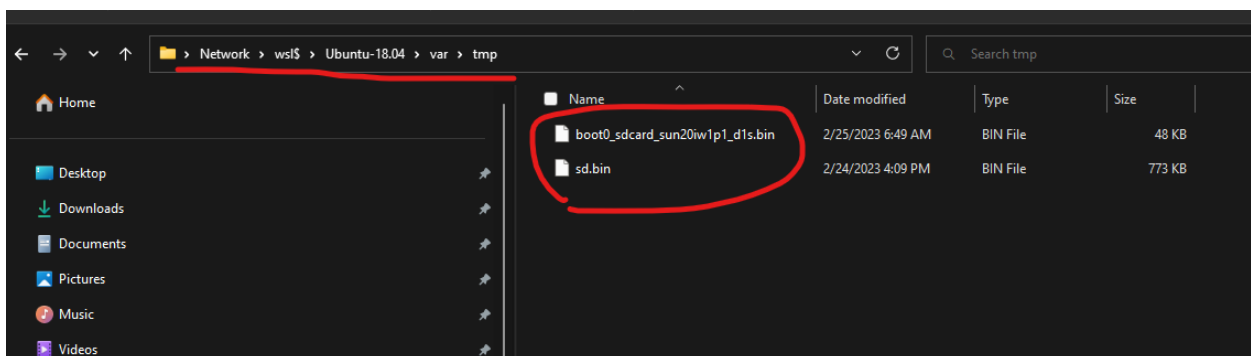
```
#ls -ald /var/tmp (double check the /var/tmp folder permission, it should be drwxrwxrwt)
```

```
#cp sd.bin /var/tmp/ ( copy the sd.bin to /var/tmp folder)
```

```
#cd tools
```

```
#cp boot0_sdcard_sun20iw1p1_d1s.bin /var/tmp (copy the boot file to /var/tmp folder)
```

Now open the folder `/var/tmp` on your local machine to copy (not cut) the 2 files.



Download the images to SD card

Downloading the below 2 files required a 3rd party SD card writer software.

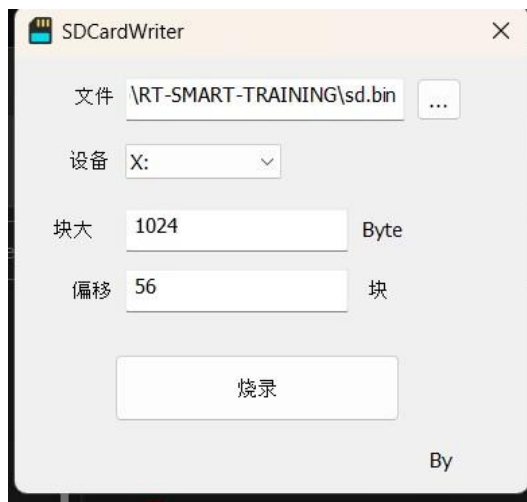
Boot0_sdcard_sun20iw1p1_d1s.bin
sd.bin

I used the following open SD card writer - <https://github.com/malasy/SDCardWriter/releases>.
And download the files to the SD card according to the below screenshots.

Boot0_sdcard_sun20iw1p1_d1s.bin

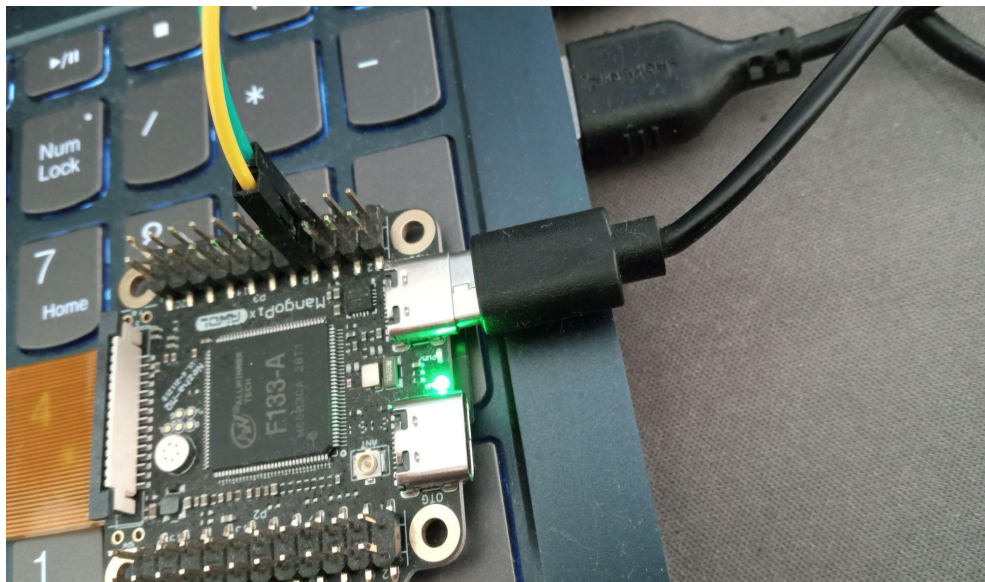
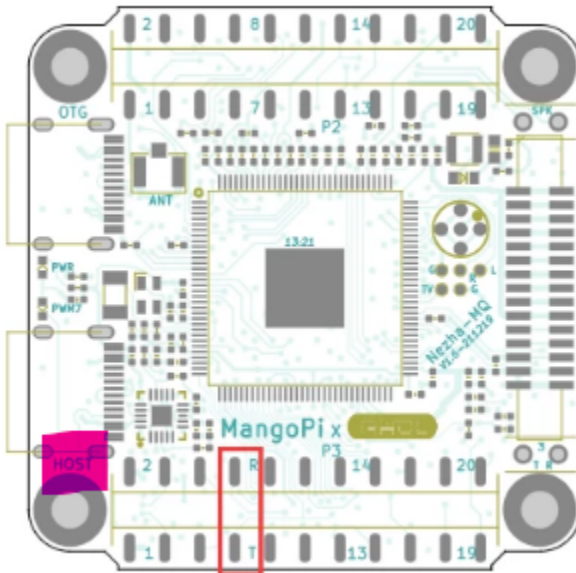


sd.bin

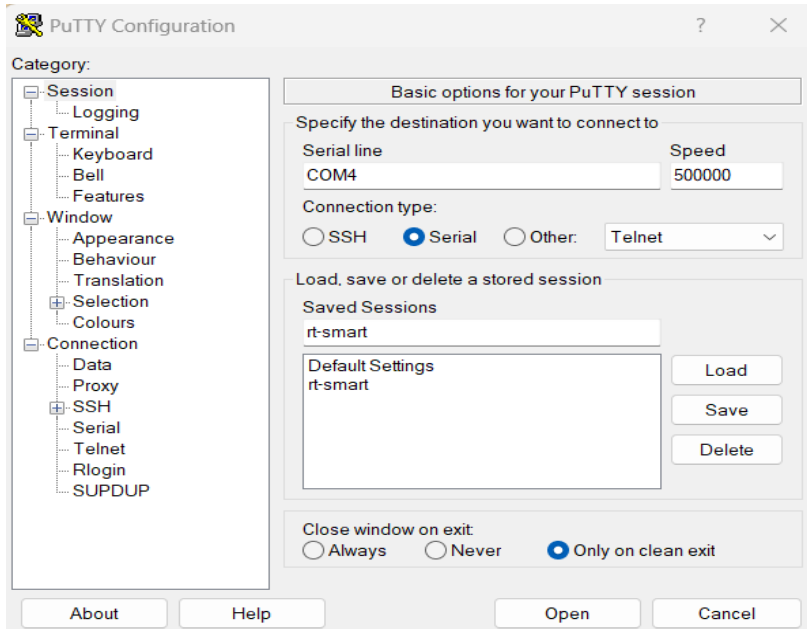


Connect to rt-smart terminal

Connect Rx and Tx cable to the dev board and use the usb c type cable to power up the device.



Finally, use the Putty.exe to connect to rt-smart's terminal using 500000 baud rate. Serial COM port numbers will vary depending on different machines. Device Manager to find out the exact COM port number respective to your machine.



```

\ | /
- RT -      Thread Smart Operating System
/ | \      5.0.0 build Feb 24 2023 16:06:52
2006 - 2022 Copyright by RT-Thread team
hal_sdc_create 0
card_detect insert
Initial card success. capacity :29818MB
sdmmc bytes_per_secotr:200, sector count:3a3d000
found part[0], begin: 1048576, size: 8.0MB
found partition:sd0 of mbr at offset 0x0000000000000800, size:0x0000000000004000
found part[1], begin: 9437184, size: 29.112GB
found partition:sd1 of mbr at offset 0x0000000000004800, size:0x0000000003a38000
hal_sdc_create 1
card_detect insert
Initial card failed!!
[E/drv-sdmmc] init sdmmc failed!
[E/drv-sdmmc] sdmmc init failed!
[D/FAL] (fal_flash_init:47) Flash device |          sdcard0 | addr: 0x00000000 | len: 0x47a00000 |
[I/FAL] ===== FAL partition table =====
[I/FAL] | name          | flash_dev | offset | length |
[I/FAL] |-----|-----|-----|-----|
[I/FAL] | download   | sdcard0  | 0x00800000 | 0x00800000 |
[I/FAL] | easyflash  | sdcard0  | 0x01000000 | 0x00100000 |
[I/FAL] | filesystem | sdcard0  | 0x01100000 | 0x00c00000 |
[I/FAL] |-----|-----|-----|-----|
[I/FAL] RT-Thread Flash Abstraction Layer initialize success

```

-Byte-me-stan
12/25/2023