望获 embedded OS

望 获 操 作 系 统

—— 高可靠嵌入式实时操作系统 ——

使用说明手册

（v1.1）

北京国科环宇科技股份有限公司

2024 年 4 月 30 日

前言

概述

本文档详细介绍了望获操作系统的安装启动、激活，启动参数配置，界面、网络、镜像源、时钟、备份还原等功能的配置、实时性的测试、优化等，使得用户对望获操作系统有比较全面深入的了解。

读者对象

本文档主要适用于以下人员：

- 使用望获操作系统的用户
- 基于望获操作系统进行开发的开发者
- 其他需要了解望获操作系统的人员

文档历史记录

版本号	实施日期	编写人	修订摘要
1.0	2023.10.10	刘洪岩	初始版本
1.1	2024.02.21	蔡纪良	增加“前言”章节和页眉页脚，并勘误； 增加系统激活； 增加实时性调优； 增加调试串口章节； 增加系统概述； 增加前置条件； 增加系统烧录； 增加系统启动安装； 增加全平台的备份还原。

目　　录

1 系统概述

1.1 支持平台一览

- Phytium(2000，e2000，d2000，d2000v)
- X86
- Rockchip(3588)
- ZYNQ7000

1.2 平台 BIOS 一览

表 1- 1 平台 BIOS 一览表

平台 \ BIOS	grub	uboot
Phytium	Yes	Yes
Rockchip	No	Yes
X86	Yes	No
ZYNQ7000	No	Yes

1.3 启动说明

几乎所有平台都支持从启动盘启动，进行试用或安装（将系统安装到板卡的其他存储介质，如 NVME、EMMC 等）。

Rockchip3588 平台对于启动盘的支持，比较特殊。该平台若想从启动盘启动，必须提前将 Uboot 镜像烧录到板卡 EMMC 的第一分区，那么，在烧录 Uboot 镜像时，将整个系统烧录到 EMMC 是更加便捷的。故该平台在使用时，通常首次烧录为整个系统镜像的烧录（包括 Uboot 镜像），后续在保证 Uboot 镜像正常运行的情况下，可以通过启动盘启动升级系统。

但以下平台不支持从启动盘启动：

- ZYNQ7000 平台：该平台的 Uboot 镜像，只能放在 SD 卡（视为启动盘）中，故完成启动盘启动，并安装系统到 EMMC 后，拔出 SD 卡，板卡将无 BIOS 可用。

1.4 镜像格式

望获操作系统的常用镜像格式为 wic，其通常包含两个分区（个别平台，略有不同），第一个分区为启动分区，内含诸如 Linux 内核、设备树、initramfs 等文件；第二分区为根文件系统。

Rockchip3588 平台的系统镜像，有 update.img 和 wic 格式两种，不同点在于，只有 update.img 可以用于无系统板卡的首次烧录，wic 常用于制作启动盘进行升级系统。wic 制做的启动盘，要求板卡中的 BIOS（在 Rockchip3588 中为 Uboot）运行正常；而 update.img 则无此限制。

ZYNQ 平台没有统一的镜像文件，需要手动制作 SD 卡分区并填充文件。

2 前置条件

2.1 数据备份

安装望获操作系统之前，最好将硬盘上的重要数据备份到软盘、U 盘、磁带等存储介质上，以避免在安装过程中发生意外，带来不必要的损失。通常要做备份的内容包括系统分区表、系统中的重要文件和数据等。

2.2 调试串口配置

本章不是必选项。

本章描述串口工具的配置方法，在安装启动的过程中，是否需要连接、配置串口，以各平台系统的安装启动教程为准。

2.2.1 Windows 串口

2.2.1.1 SerialPort 查询

将板卡的串口和 PC 机连接，打开电脑设备管理器获取调试串口的 Serial Port，如果有多个调试串口，可以通过插拔对比的方式，确认使用的是哪一个。如图 2-1 所示（此例中为 com8）。

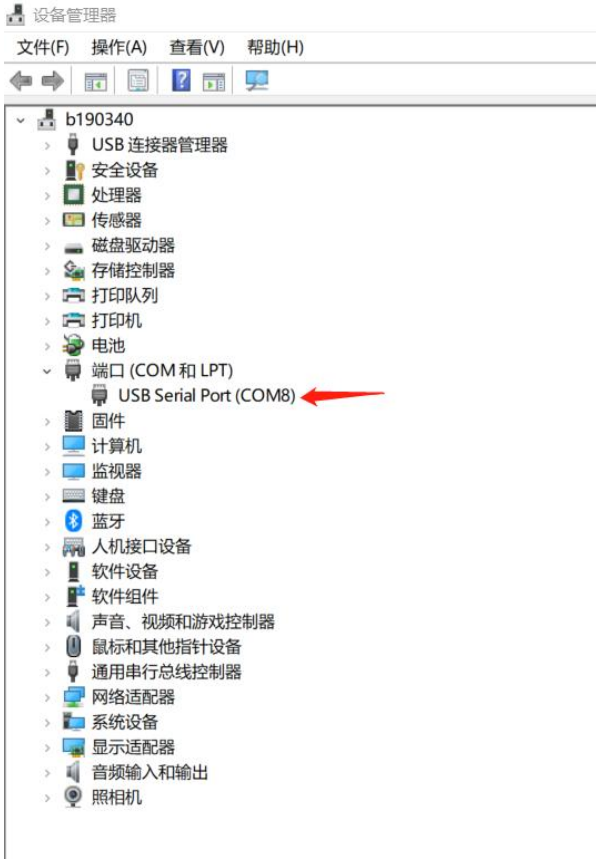

图 2- 1 SerialPort 查询

2.2.1.2 串口工具配置

打开串口调试工具（SecureCRT，MobaXterm 等均可），配置 Port、波特率 115200 并禁止流控，如图 2-2 所示。

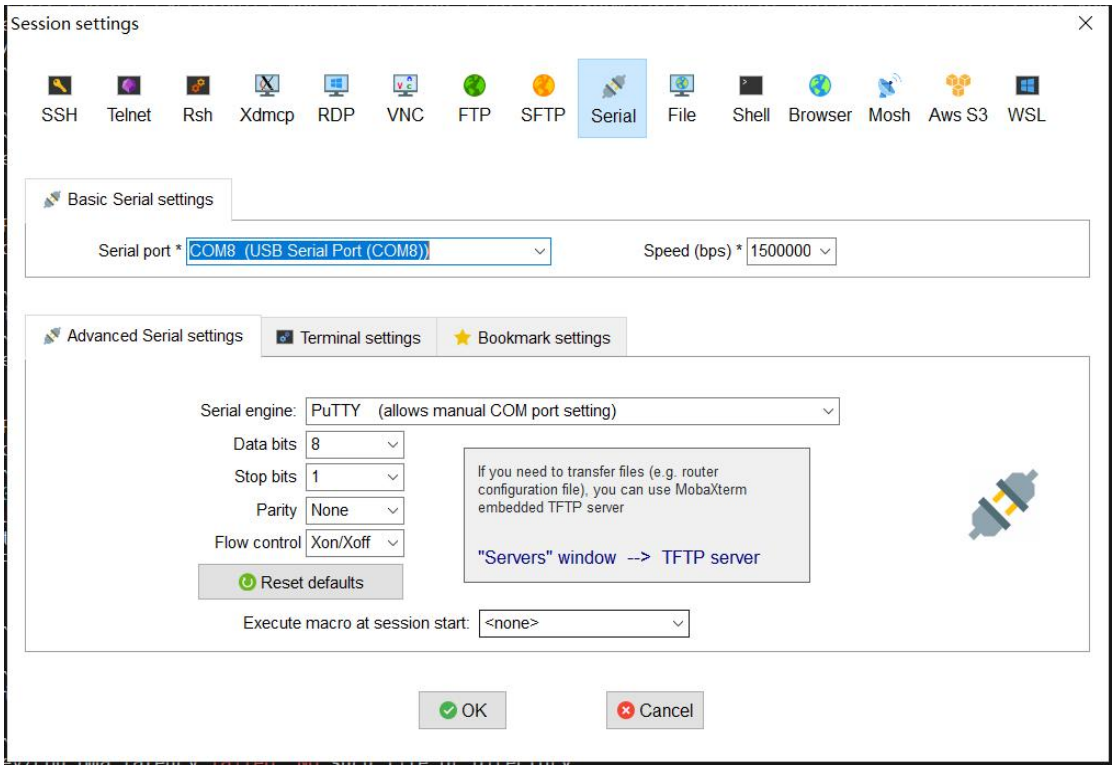

图 2- 2 MobaXterm 配置串口

2.2.2 Ubuntu 串口

2.2.2.1 串口设备查询

1. 将板卡的串口和 PC 机连接，执行命令 ls /dev
2. 将板卡的串口和 PC 机断开连接，执行命令 ls /dev
3. 对比上述两次命令的结果，会发现第一步比第二步多一个设备，则该设备为当前的串口设备，一般为/dev/ttyUSB0（本小节将以此为例，请根据实际情况修改）
4. 打开串口工具（工具不唯一，根据个人喜好即可，这里以 minicom 为例），配置为 8N1 115200，无流控，如图 2-3 所示。

图 2- 3 minicom 配置串口

2.3 BIOS 烧录

本章不是必选项。

对于板卡中存在可用的 BIOS 的情况，可以不必重新烧录 BIOS；本章并非描述了所有平台的 BIOS 烧录方式，若本章中不存在您所需要的 BIOS，建议您联系硬件厂商获取 BIOS 二进制及烧录方式。

2.3.1.1 Phytium 平台

Phytium 平台有两种 BIOS，一种是 Uboot，一种是 UEFI。在下文中提到的 fip-all.bin 文件，既有可能是 Uboot，也有可能是 UEFI，根据实际的业务需求选择即可。

2.3.1.1.1 准备工作

- 准备一台 Windows 主机
- 下载 Flypro，并安装。下载链接：https://www.sflytech.com/download/Software/FlyPRO_Setup.zip
- 获取 fip-all.bin 文件

2.3.1.1.2 建立 PC 与烧录器的连接

Flash 芯片安置到烧录器，并通过 USB 将烧录器和 PC 进行连接。

2.3.1.1.3 检测芯片型号

运行 Flypro.exe，点击芯片按钮，选择芯片的相应类别，开始检测，双击芯片型号。具体方法如图 2-4 所示。注：具体使用的芯片可能不同，根据实际情况选择。

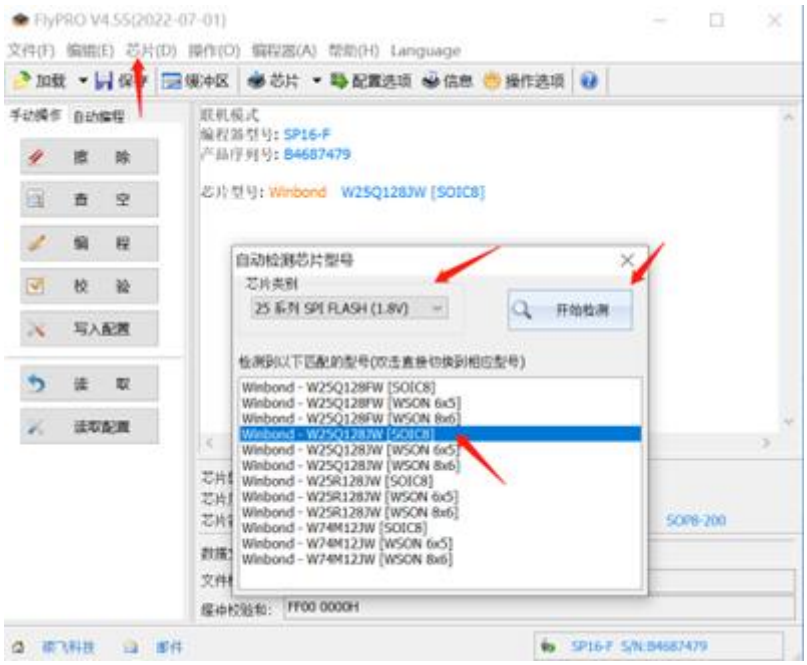

图 2- 4 检测芯片型号

2.3.1.1.4 加载 fip-all.bin 文件

点击加载按钮，在弹出的窗口中选择 fip-all.bin 文件，并打开。方法如图 2-5 所示。

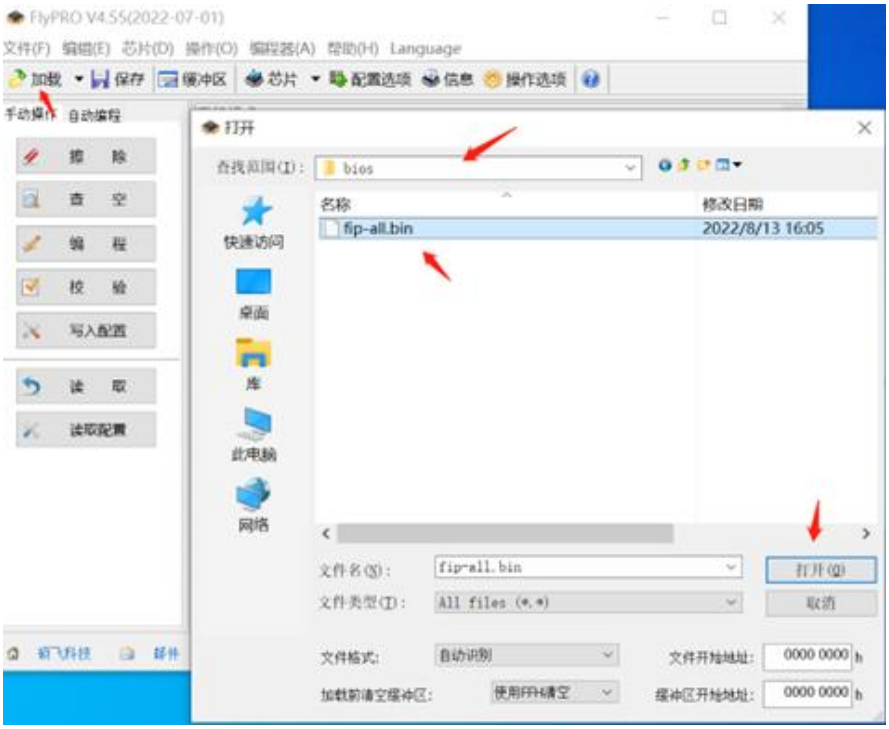

图 2- 5 加载 fip-all.bin

2.3.1.1.5 烧录芯片

点击自动编程按钮，点击单次烧录按钮，进行烧录。方法如图 2-6 所示。烧录成功将会出现如图 2-7 所示的相应内容。

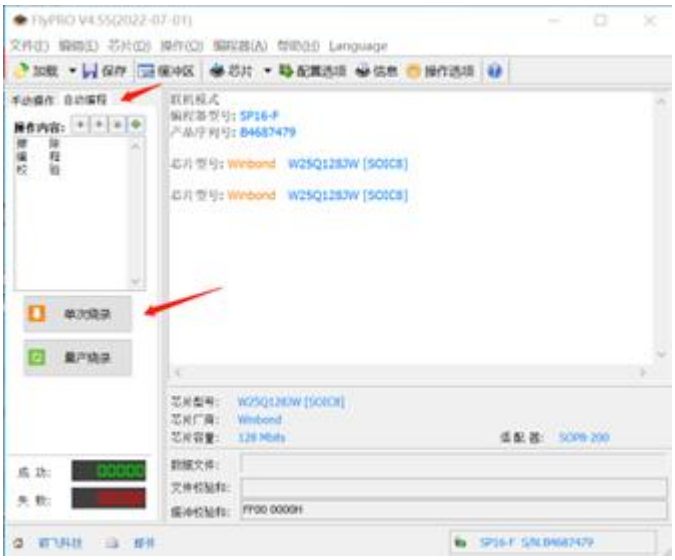

图 2- 6 烧录

图 2- 7 烧录成功

2.3.1.2 Rockchip 平台

Rockchip 平台只支持 Uboot 作为 BIOS，且其 Uboot 镜像会被一同打包到系统镜像中，在系统烧录时，统一烧录到 EMMC，故此章节不需描述单独烧写 Uboot 的方法。

2.3.1.3 ZYNQ7000 平台

ZYNQ7000 平台只支持 Uboot 作为 BIOS，且其 Uboot 镜像是以文件的形式直接放置在启动盘的第一分区中的。详情参见其安装启动教程。

2.4 wic 镜像启动盘制作

2.4.1 Ubuntu 启动盘制作

以下步骤中，有些命令需要用到 root 权限。

1. 将 U 盘插入 Ubuntu（注意，该盘会被清空，如有重要资料请备份）
2. 使用 lsblk 或 df -h 命令，查看 U 盘是否被挂载，如果被挂载了，先卸载 U 盘。
3. 制作启动盘：dd if= wanghuo-image-standard-XXXX.wic of=/dev/sdb bs=1M oflag=direct status=progress
   需要注意的是，本命令中的参数，需要根据实际情况修改。其中 wanghuo-image-standard-XXXX.wic 为具体的系统镜像；/dev/sdb 应替换为实际的 U 盘设备。
4. 执行命令：sync

5. 确保 U 盘没有被自动挂载，如果挂载上了的话，先卸载。
6. 拔出 U 盘

2.4.2 Windows 制作启动盘

1. 进入 https://rufus.ie/zh/，下载烧写工具 Rufus 并安装。
2. 运行 Rufus，选择插入的 U 盘。

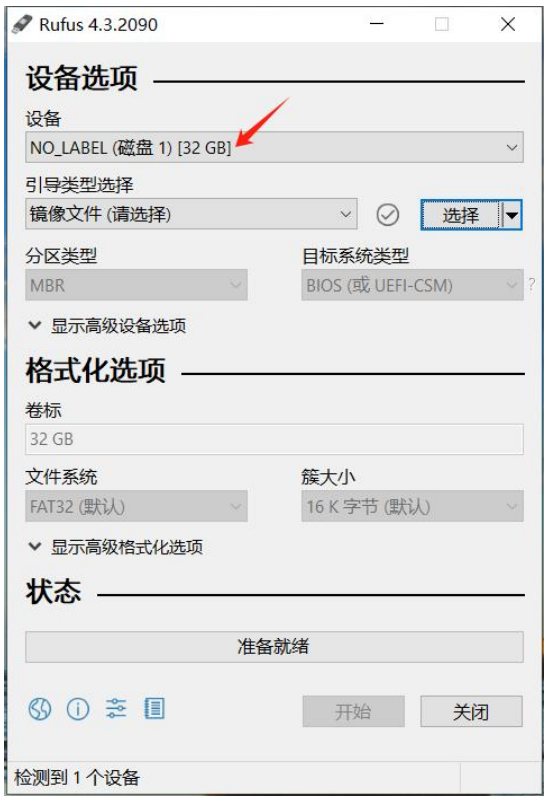

图 2- 8 选择 U 盘

3. 点击选择按钮，选择镜像文件，然后点击开始按钮。

图 2- 9 开始烧录

4. 烧录过程中，会有进度条显示。当“状态”区域显示绿色背景“准备就绪”时，表示已经烧写完成。注：烧写过程中可能会弹窗需要点击“确定”；确保烧写过程中没有报错，否则需要重新烧录。

图 2- 10 烧录成功

5. 在 Windows 上移除设备，拔出 U 盘。

3 系统烧录

对于完全支持启动盘启动的系统（详情见“1.3 启动说明”），本章描述的是制作启动盘的流程；对于不完全支持启动盘启动的系统，本章描述的是把系统烧录到启动介质的流程。

特别的，对于 Rockchip3588 平台，本章既描述其首次烧录完整系统镜像（内含 Uboot 镜像）的方式，又描述了其 Uboot 运行正常的情况下，制作启动盘的流程。

3.1 Phytium

3.1.1 硬件平台

目前支持 phytium2004，e2000，d2000，d2000v。

3.1.2 启动盘制作

启动盘的制作，需要使用.wic 格式的镜像，确保该镜像的文件名和所使用的硬件平台是对应的。具体制盘方法参考“2.4 wic 镜像启动盘制作”。

3.2 Rockchip

3.2.1 硬件平台

芯片支持列表：RK3588
主板支持列表：TB-RK3588X 等

3.2.2 系统镜像烧录

本章描述了烧写完整的系统镜像（Uboot 镜像+启动分区+根文件系统）到 EMMC 的方法。该方法可以作为板卡首次烧录的方法，也可以用作后续系统升级的方法。

3.2.2.1 前置条件

准备一台 Windows 主机。

进入下载链接，下载烧写工具 Flashtool 压缩包。

3.2.2.2 文件说明

update.img，完整的系统镜像，内含 Uboot 镜像、启动分区、根文件系统。

3.2.2.3 系统镜像烧录

3.2.2.3.1 安装驱动

FlashTool 有中文版和英文版，以中文版 FlashTool_CN 为例，双击 FlashTool_CN\Windows\DriverAssitant\DriverInstall.exe 打开安装程序，点击图 3-1 所示的“驱动安装”按钮，按提示安装 USB 驱动。如果已经安装旧版本的烧写工具，先点“驱动卸载”按钮卸载驱动，然后点击“驱动安装”按钮安装驱动。

图 3-1 安装驱动

3.2.2.3.2 打开烧写工具

以中文版 RKDevTool_Release_v2.84-CN 为例，双击 RKDevTool_Release_v2.84-CN\RKDevTool_Release_v2.84-CN\RKDevTool.exe，进入“升级固件”页面，如下图所示：

图 3- 2 升级固件页面

3.2.2.3.3 进入烧写模式

将板卡的烧写线（USB 接口）接入 windows 主机，然后通过以下方式进入烧写模式（键位说明如图 3-3 所示）：

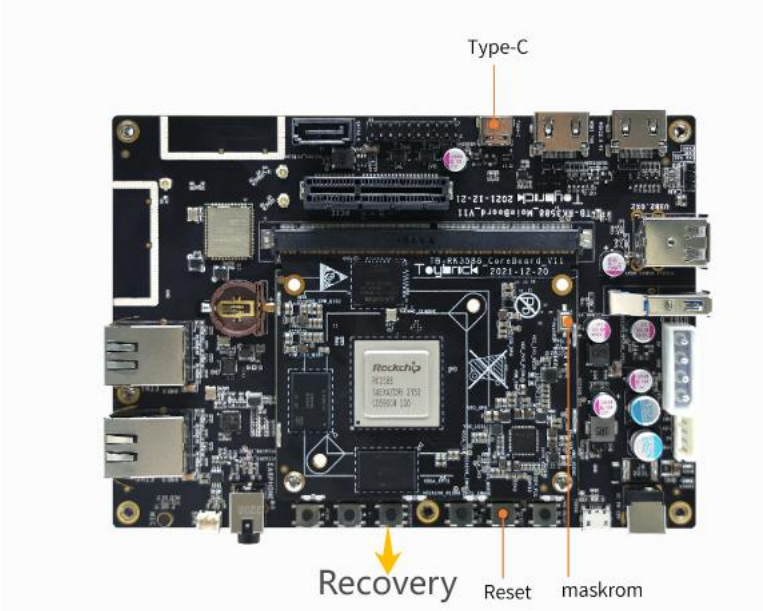

图 3- 3 按键说明

- loader 烧写模式：

  开发板上电=>按住主板的 Recovery 键不放=>点击一下 Reset 键=>当开发板进入 loader 模式后松开 Recovery 键。

图 3- 4 loader 模式

- maskrom 烧写模式：

  开发板上电-按住主板的 Maskrom 键不放-点击一下 Reset 键-当开发板进入 maskrom 模式后松开 Maskrom 键。

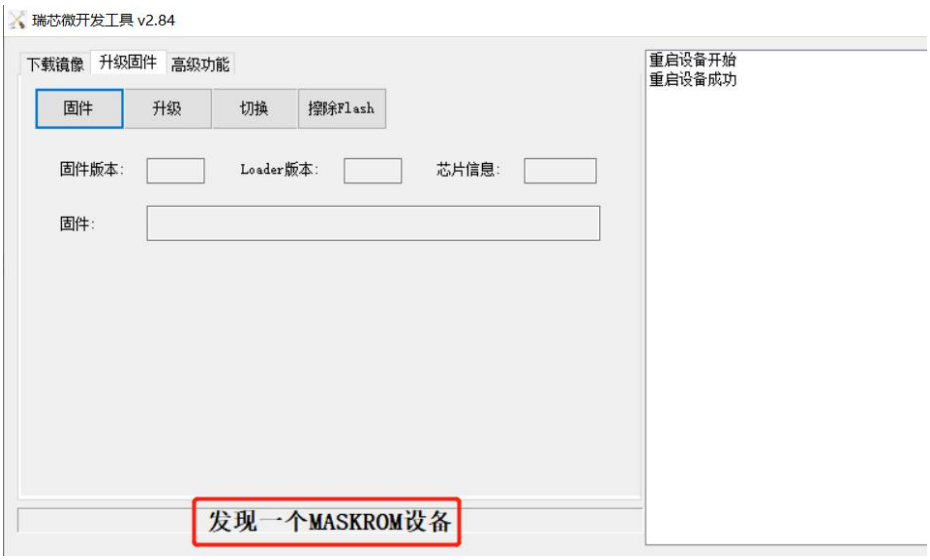

图 3- 5 maskrom 模式

3.2.2.3.4 烧写固件

1. 连接并配置好调试串口（也可以接键盘显示器）
2. 点击烧写工具中的“固件”按钮，选择 update.img，确保此时板卡处于烧写模式（loader 模式或者 maskrom 模式）。

图 3- 6 选择固件

3. 点击“升级”按钮。

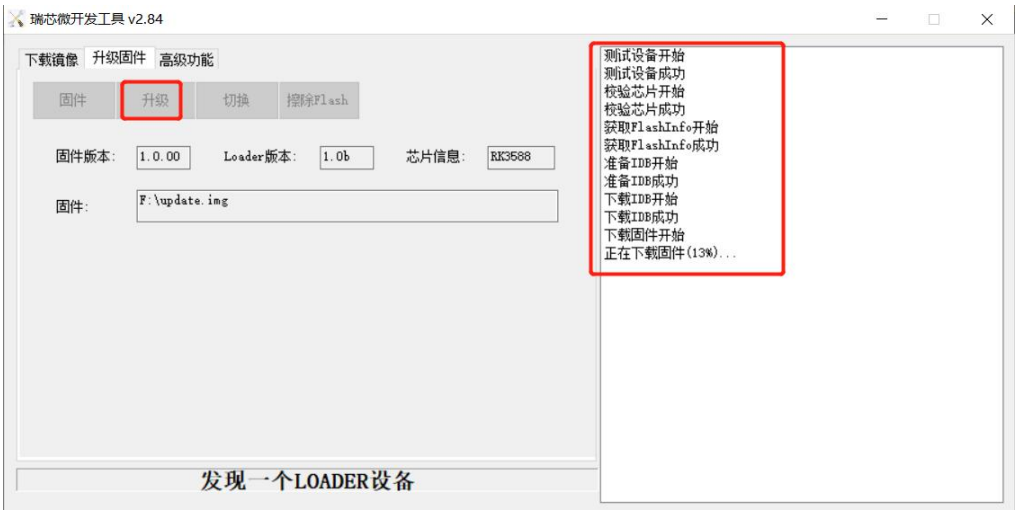

图 3- 7 点击升级按钮

4. 升级成功后会自动重启，参考“4.2 启动”所述启动登录即可。

图 3- 8 升级成功

3.2.3 启动盘制作

启动盘的制作，需要使用.wic 格式的镜像，确保该镜像的文件名和所使用的硬件平台是对应的。具体制盘方法参考“2.4 wic 镜像启动盘制作”。

3.3 x86 版本系统安装

3.3.1 硬件平台

目前支持酷睿系列、赛扬系列。

3.3.2 启动盘制作

启动盘的制作，需要使用.wic 格式的镜像，确保该镜像的文件名和所使用的硬件平台是对应的。具体制盘方法参考“2.4 wic 镜像启动盘制作”。

3.4 ZYNQ 版本系统安装

3.4.1 硬件平台

芯片支持列表：ZYNQ7000
主板支持列表：MZ7045FA

3.4.2 准备工作

- 32G 及以上容量 SD 卡一个，读卡器一个。
- 进入望获实时 Linux 系统(onewos.com)获取 ZYNQ7000 系统文件，分别是：
  - BOOT.BIN
  - uImage
  - milianke-system.dtb
  - uEnv.txt
  - wanghuo-image-standard-milianke.tar.gz

3.4.3 制作 SD 卡

3.4.3.1 SD 卡分区

1. 将 SD 卡插入 Ubuntu 系统，以/dev/sdc 设备为例。
2. sudo fdisk /dev/sdc，进入 fdisk 命令模式，一下为操作命令，#后为注释：
   a) p #打印当前 SD 卡分区
   b) d #删除分区，若有多个，多次执行 d 命令，依次删除即可
   c) o #新建一份空的 DOS 分区表，防止 SD 卡为 GPT 分区
   d) n #创建分区
   e) p #创建的分区为主分区
   f) 回车 #分区号，默认即可
   g) 回车 #第一扇区起始，默认即可
   h) +1G #设置分区大小

```
wzq@wzq-VirtualBox:~/work$ sudo fdisk /dev/sdc

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/sdc: 30 GiB, 32212254720 bytes, 62914560 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x97729e05

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-62914559, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-62914559, default 62914559): +1G

Created a new partition 1 of type 'Linux' and of size 1 GiB.
```

图 3- 9 创建分区 1

i) a #标记为可启动分区，后续可以通过 p 命令查看启动标记

```
Command (m for help): a
Selected partition 1
The bootable flag on partition 1 is enabled now.

Command (m for help): p
Disk /dev/sdc: 30 GiB, 32212254720 bytes, 62914560 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x97729e05

Device     Boot Start     End Sectors Size Id Type
/dev/sdc1  *     2048 2099199 2097152   1G 83 Linux

Filesystem/RAID signature on partition 1 will be wiped.

Command (m for help):
```

图 3- 10 可启动标记

j) n #创建分区

k) p #创建的分区为主分区

l) 回车 #分区号，默认即可

m) 回车 #第一扇区起始，默认即可

n) 回车 #设置分区大小，默认即可，剩余全部空间

```
Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (2099200-62914559, default 2099200):
Last sector, +sectors or +size{K,M,G,T,P} (2099200-62914559, default 62914559):

Created a new partition 2 of type 'Linux' and of size 29 GiB.
```

图 3- 11 创建第二分区

o) p #查看两个分区

p) w #保存分区

```
Command (m for help): p
Disk /dev/sdc: 30 GiB, 32212254720 bytes, 62914560 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x97729e05

Device     Boot   Start      End  Sectors Size Id Type
/dev/sdc1  *       2048  2099199  2097152   1G 83 Linux
/dev/sdc2       2099200 62914559 60815360  29G 83 Linux

Command (m for help): w
The partition table has been altered.
Syncing disks.

wzq@wzq-VirtualBox:~/work$
```

图 3- 12 分区完成

3.4.3.2 SD 卡内容填充

本章节继续以/dev/sdc 为例，实际情况不同，可能需要修改命令参数。

1. 格式化两个分区
   sudo mkfs.vfat -F 32 -n BOOT /dev/sdc1
   sudo mkfs.ext4 -L ROOT /dev/sdc2
2. 挂载/dev/sdc1 到系统/mnt 目录，将 wanghuo-image-standard-milianke.tar.gz 解压过去，卸载设备

```
wzq@wzq-VirtualBox:~/work$ sudo mount /dev/sdc1 /mnt/
wzq@wzq-VirtualBox:~/work$ sudo cp uImage milianke-system.dtb uEnv.txt BOOT.bin /mnt/
wzq@wzq-VirtualBox:~/work$ ls /mnt/
 BOOT.BIN   milianke-system.dtb  'System Volume Information'   uEnv.txt   uImage
wzq@wzq-VirtualBox:~/work$ sudo umount /mnt
wzq@wzq-VirtualBox:~/work$
```

图 3- 13 填充第 1 分区

3. 挂载/dev/sdc2 到系统/mnt 目录，将 BOOT.BIN，uImage，uEnv.txt，milianke-system.dtb 四个文件拷贝过去，卸载设备

```
wzq@wzq-VirtualBox:~/work$ sudo mount /dev/sdc2 /mnt/
wzq@wzq-VirtualBox:~/work$ sudo tar -zxf wanghuo-image-standard-milianke.tar.gz -C /mnt/
wzq@wzq-VirtualBox:~/work$ sync
wzq@wzq-VirtualBox:~/work$ ls /mnt/
bin  boot  dev  etc  home  lib  log_lock.pid  media  mnt  proc  run  sbin  srv  sys  tmp  usr  var
wzq@wzq-VirtualBox:~/work$ sudo umount /mnt
wzq@wzq-VirtualBox:~/work$
```

图 3- 14 填充第 2 分区

3.4.4 启动系统

3.4.4.1 连接串口

1. 将制作好的 SD 卡启动盘插入到目标机器，上电启动,开头可以看到串口打印如下：

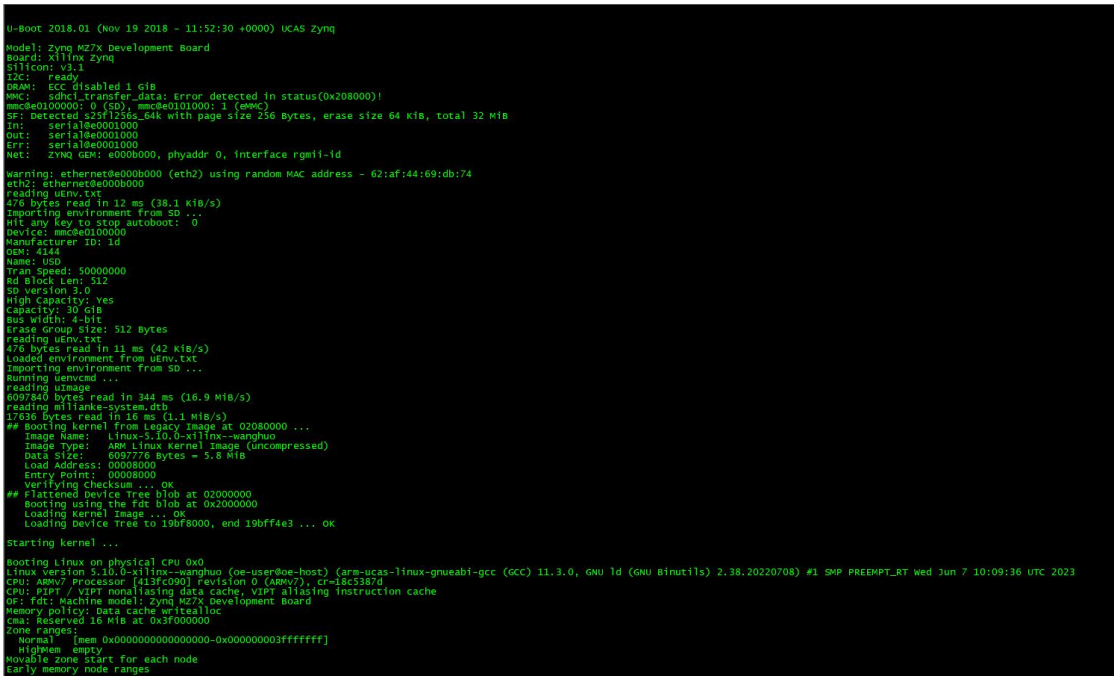

图 3- 15 串口打印

2. 进入到登录界面,输入用户名:root，密码:wanghuo，即可登录系统。

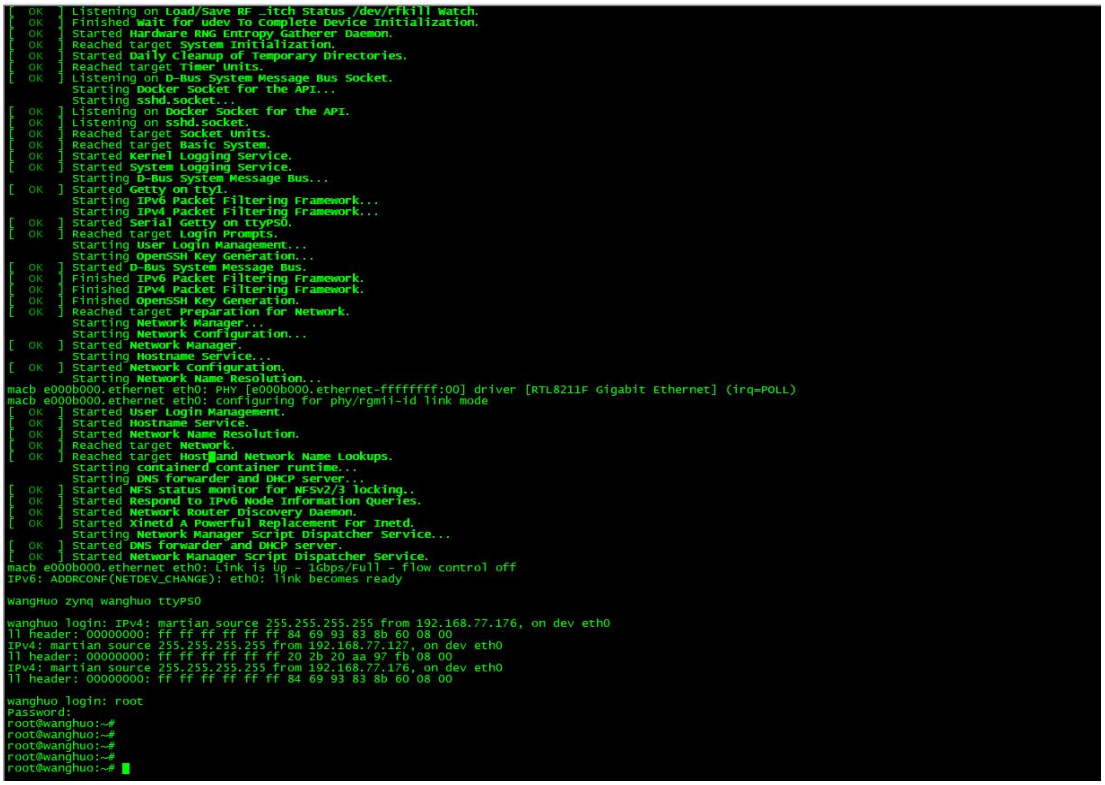

图 3- 16 登录系统

4 系统安装启动

对于支持启动盘启动的系统（详情见“1.3 启动说明”），本章描述的是从启动盘启动系统，并将系统安装到其他存储介质的流程；对于其他系统，本章描述的是从启动介质中启动系统的流程。

4.1 安装

本章描述的是，制作启动盘之后，用该启动盘启动并安装系统到其他存储介质的流程。

4.1.1 Uboot 进入安装模式

当基于 Uboot 启动时，请先按章节“2.2 调试串口配置”所述，连接并配置串口。

除了飞腾平台，其他平台仅需给 bootargs 环境变量，临时追加（不要保存）LABEL=install-efi 即可，然后执行 run bootcmd 即可将启动盘启动。

飞腾平台，则需要按照以下步骤进入安装模式。

4.1.1.1 查看启动文件

查看启动文件的主要目的，是确定各个文件名称（不同硬件平台所使用的启动文件的名称略有不同），以便 Uboot 去加载对应的文件。

将启动盘插在主机上（Ubuntu、Windows 均可），Ubuntu 一般会自动挂载，然后查看第一个分区的内容，可以看到如图 4-1 所示的内容（不同平台，文件名略有不同）。可以看到，内核有多种格式，一般我们使用其中的 uImage；initramfs 选取以.u-boot 结尾的文件；设备树选择以.dtb 结尾的文件。

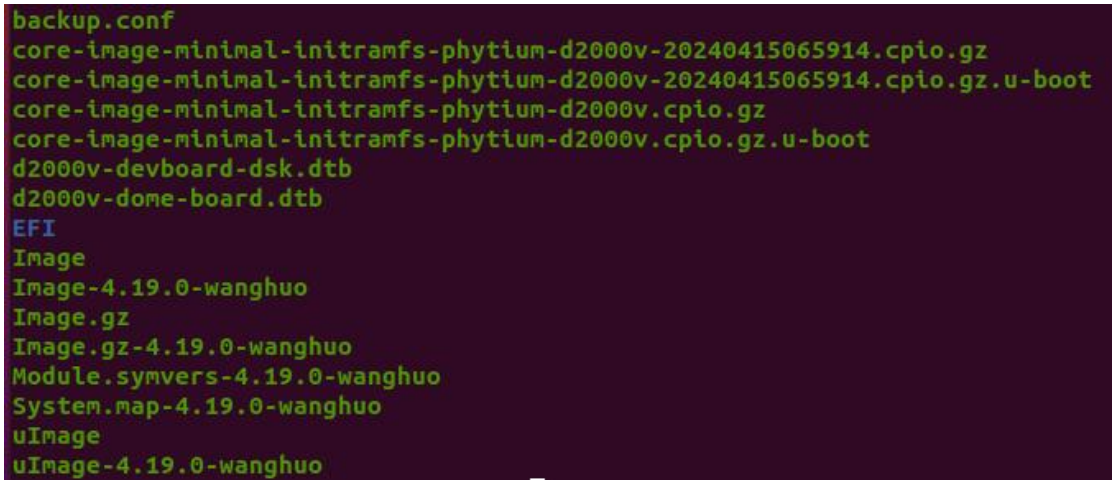

图 4- 1 分区 1 的内容示例

4.1.1.2 启动命令

对于飞腾平台，需要设置在 Uboot 中使用的启动命令（或环境变量），方法不唯一，仅以图 4-1 所示的飞腾平台为例。

加载各个文件到内存，文件名、内存地址，根据实际情况修改：

```
u-boot# usb start
u-boot# fatload usb 0:1 90100000 uImage
u-boot# fatload usb 0:1 95000000 d2000v-demo-board.dtb
u-boot# fatload usb 0:1 96000000 core-image-minimal-initramfs-phytium-d2000v.cpio.gz.u-boot
```

设置内核启动参数（根据实际硬件平台设置）：

```
u-boot# setenv bootargs ‘console=ttyAMA1,115200 earlycon=pl011,0x28001000 rdinit=/init root=/dev/sdb2 rw LABEL=install-efi’
```

启动：

```
u-boot# bootm 0x90100000 0x96000000 0x95000000
```

至此，即可进入安装模式。

4.1.1.3 安装开始

进入安装模式后，安装系统的具体流程，参见 grub 的安装流程。

4.1.2 基于 grub 启动安装

grub 启动需要连接键盘和显示器，若是 BIOS 支持串口，则可以不连接键盘

和显示器（BIOS 是否支持串口，取决于 BIOS 的提供方，一般是板卡厂商）。

在 BIOS 阶段，进入选择启动设备的界面，不同的主板、BIOS，进入的方式不同，具体咨询板卡厂商或 BIOS 的提供方。然后选择启动盘作为启动设备，如图 4-2 所示。

图 4- 2 选择启动设备

进入 grub 选择界面，如图 4-3 所示。如果选择“WangHuo OS”，则将会直接运行启动盘内的系统，输入账号 root，密码 wanghuo 登入即可。如果选择“install”，则会进入安装模式。

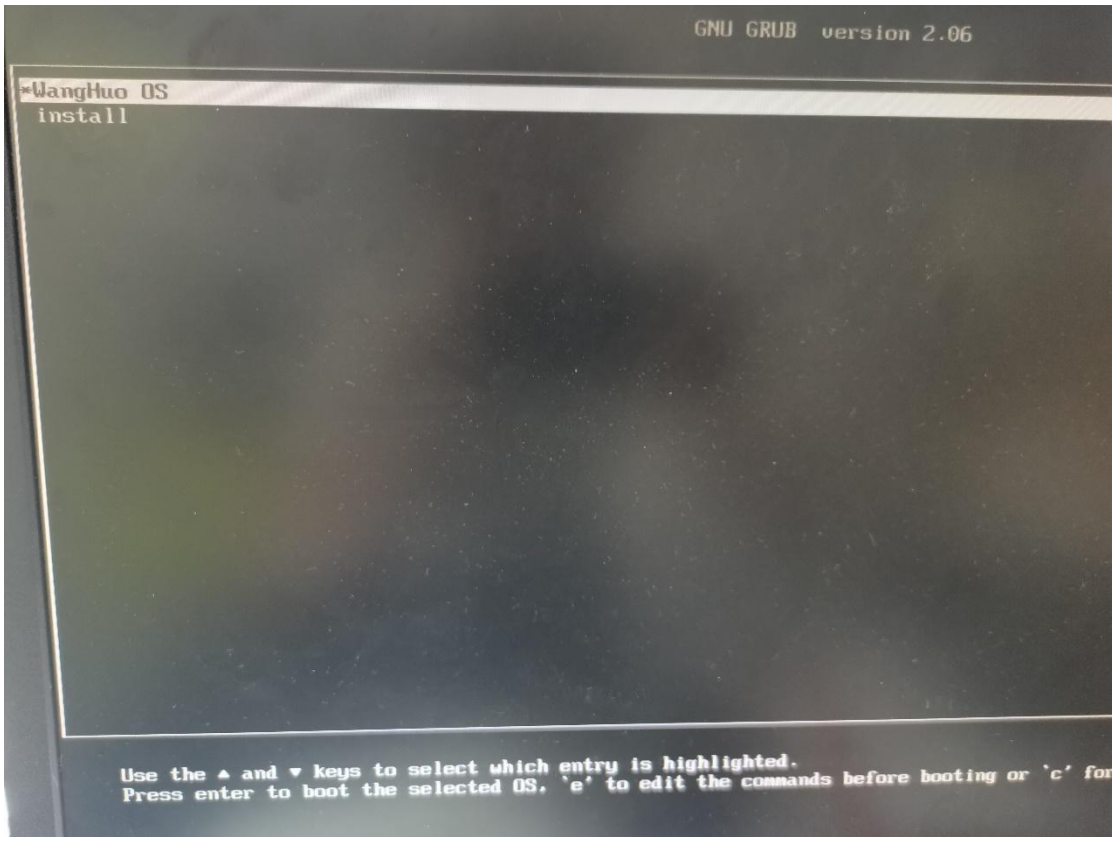

图 4- 3 grub 选择界面

进入安装模式后，提示信息如图 4-4 所示。该界面会展示目前板卡所连接的存储设备，需要键盘选择安装系统的目标盘，本章节以 nvme 设备为例。

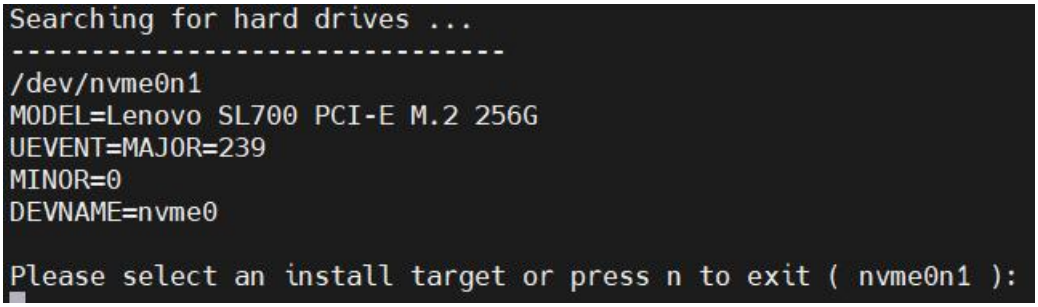

图 4- 4 选择安装目标

选择安装系统的目标盘为 nvme0n1，则进入如图 4-5 所示界面，等待安装成功即可。

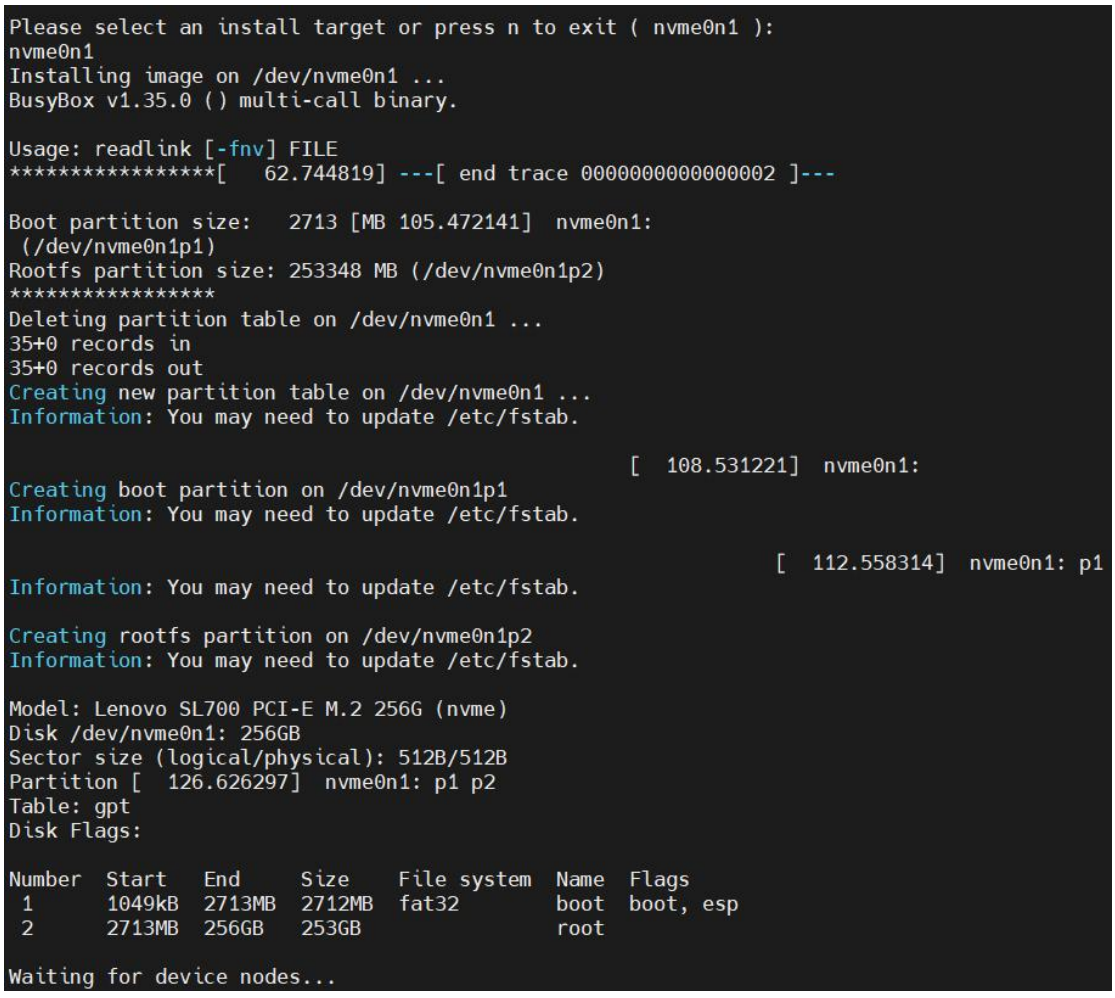

图 4- 5 安装系统

安装成功后，如图 4-6 所示。此时，根据提示，拔出启动盘，按回车键重启。

```
Formatting /dev/nvme0n1p1 to vfat...
mkfs.fat 4.2 (2021-01-31)
Formatting /dev/nvme0n1p2 to ext4...
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 61852416 4k blocks and 15466496 inodes
Filesystem UUID: 616f4d02-1dd2-11b2-a982-31e3eb7b3899
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): [  132.027702] random: crng init done
[  132.031096] random: 7 urandom warning(s) missed due to ratelimiting
done
Writing superblocks and filesystem accounting information: done

[  132.883490] EXT4-fs (nvme0n1p2): mounted filesystem with ordered data mode. Opts:
Copying rootfs files...
Copying boot files...
Preparing boot partition...
Installation successful. Remove your installation media and press ENTER to reboot.
```

图 4- 6 安装成功

安装成功后，即可按照后续的教程进行启动。

4.2 启动

本章描述的是，当系统已经安装到 EMMC、NVME 等存储介质后的启动流程。

4.2.1 Uboot 配置

本章节主要描述飞腾平台在启动时，可能需要用到的 Uboot 配置。若是其他平台，则忽略此章节。

加载各个文件到内存，文件名、内存地址，根据实际情况修改：

```
u-boot# usb start
u-boot# fatload nvme 0:1 90100000 uImage
u-boot# fatload nvme 0:1 95000000 d2000v-demo-board.dtb
u-boot# fatload nvme 0:1 96000000 core-image-minimal-initramfs-phytium-d2000v.cpio.gz.u-boot
```

设置内核启动参数（根据实际硬件平台设置）：

```
u-boot# setenv bootargs 'console=ttyAMA1,115200 earlycon=pl011,0x28001000 rdinit=/init root=/dev/nvme0n1p2 rw'
```

启动：

```
u-boot# bootm 0x90100000 0x96000000 0x95000000
```

关于 Uboot 启动，有以下几点需要注意：

1. 所有的命令，可以根据业务需求，设置为不同的环境变量组合，实现便捷启动。
2. 加载各个文件到内存时，需要根据具体的板卡、存储设备，选择合适的命令。如内存地址，可以根据当前内存大小、内存分配情况，选择合适的内存地址。
3. 内核参数，重点关注 root=这个条目，保证其选择的设备，是刷写了系统镜像的那个设备的第二分区。
4. 启动时，bootm 命令的三个参数，要根据文件实际加载的内存地址，进行传参。
5. 之后则可以按照后续章节所示的教程进行启动。

4.2.2 串口启动

请先按章节“2.2 调试串口配置”所述，连接并配置串口。

板卡上电，当调试串口显示登录时，输入用户 root，密码 wanghuo，即可登录系统。

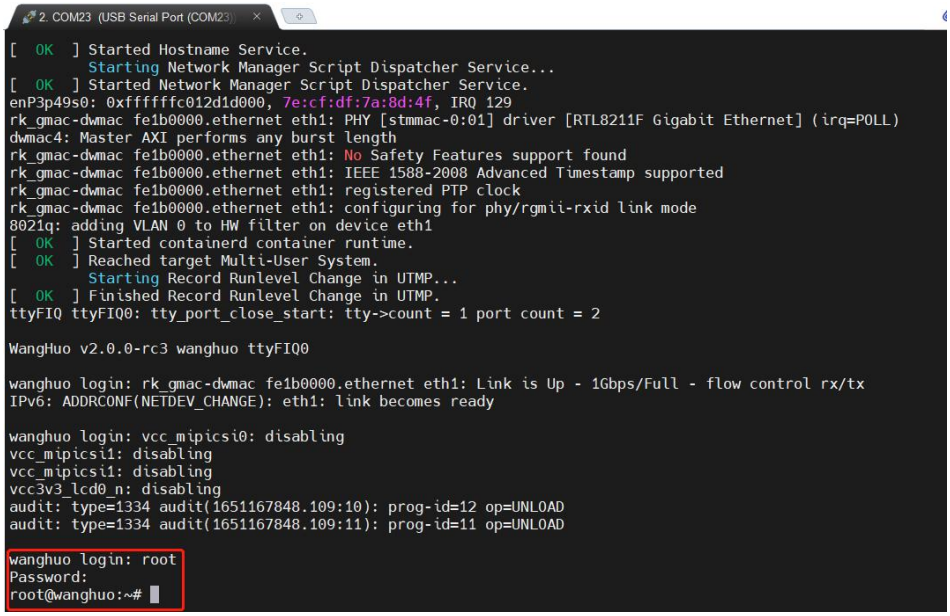

图 4- 7 登录系统

4.2.3 图形界面启动

连接鼠标、键盘、显示器到板卡，上电，待出现图形界面后，登录系统。账

号 root，密码 wanghuo。

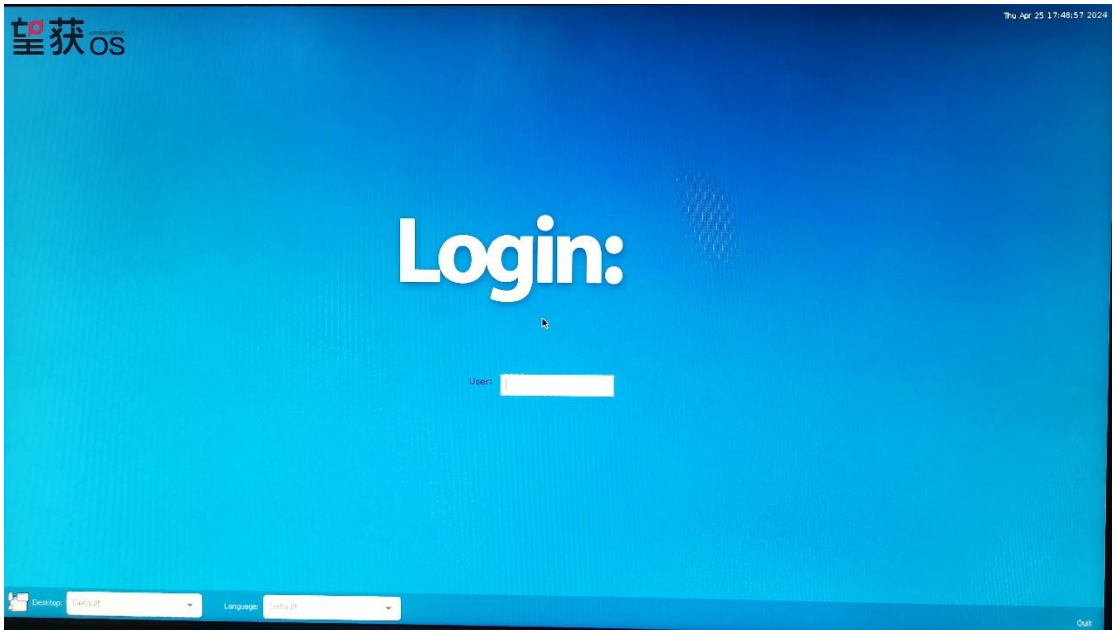

图 4- 8 登录界面

5 系统激活

未激活的 wanghuo 系统，在启动 2h 后将自动关机。

5.1 获取硬件信息

进入系统后执行 cat /proc/wh-hwinfo 可以得到设备的信息码，会得到类似图 5-1 的信息。

图 5- 1 设备信息码

5.2 获取密钥文件

联系售后服务人员，将得到的硬件信息码发送给售后服务人员以获取系统激活密钥（license.txt）。将密钥文件放到系统/etc 目录下成功激活系统。

6 网络配置

6.1 图形界面配置 DHCP

进入系统后，鼠标右键点击桌面右上角小电脑图标，选择 Edit Connections 进行编辑，如图 6-1 所示，然后弹出图 6-2。

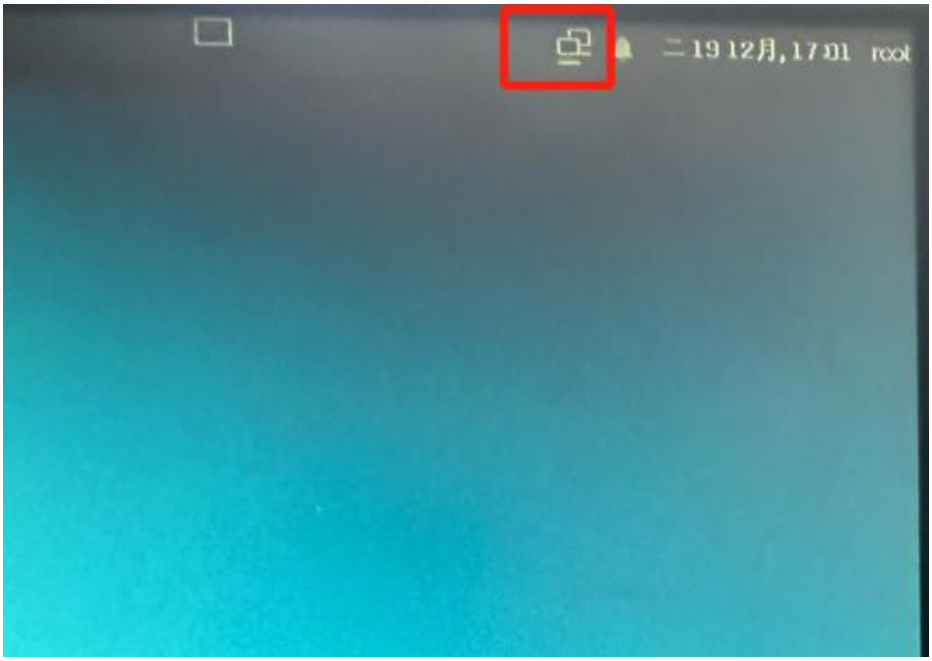

图 6- 1 打开网络设置

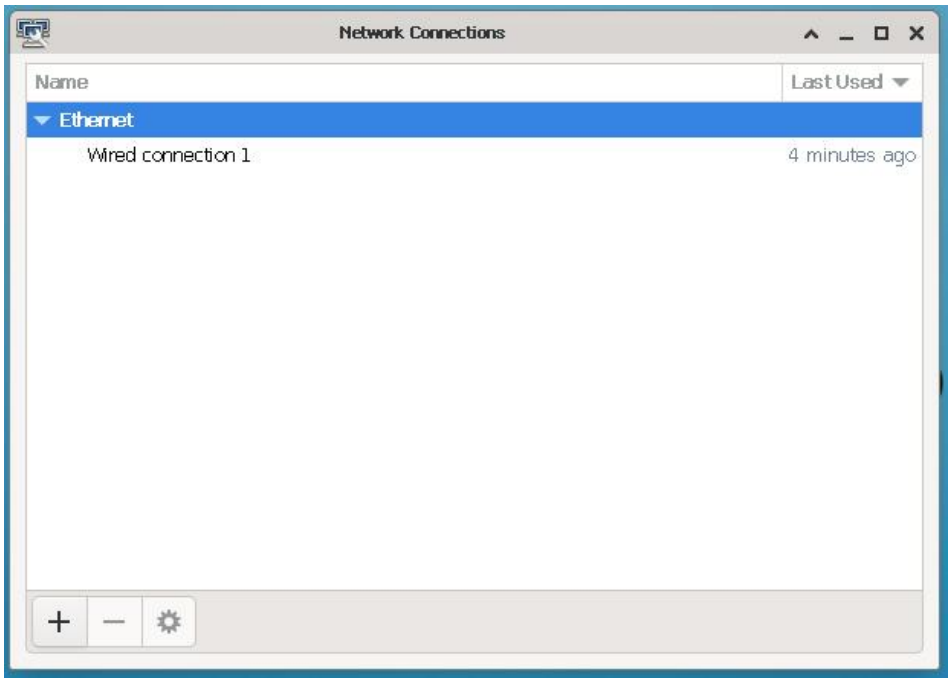

图 6- 2 编辑网络设置

双击 Wired connection 1，进入编辑界面：

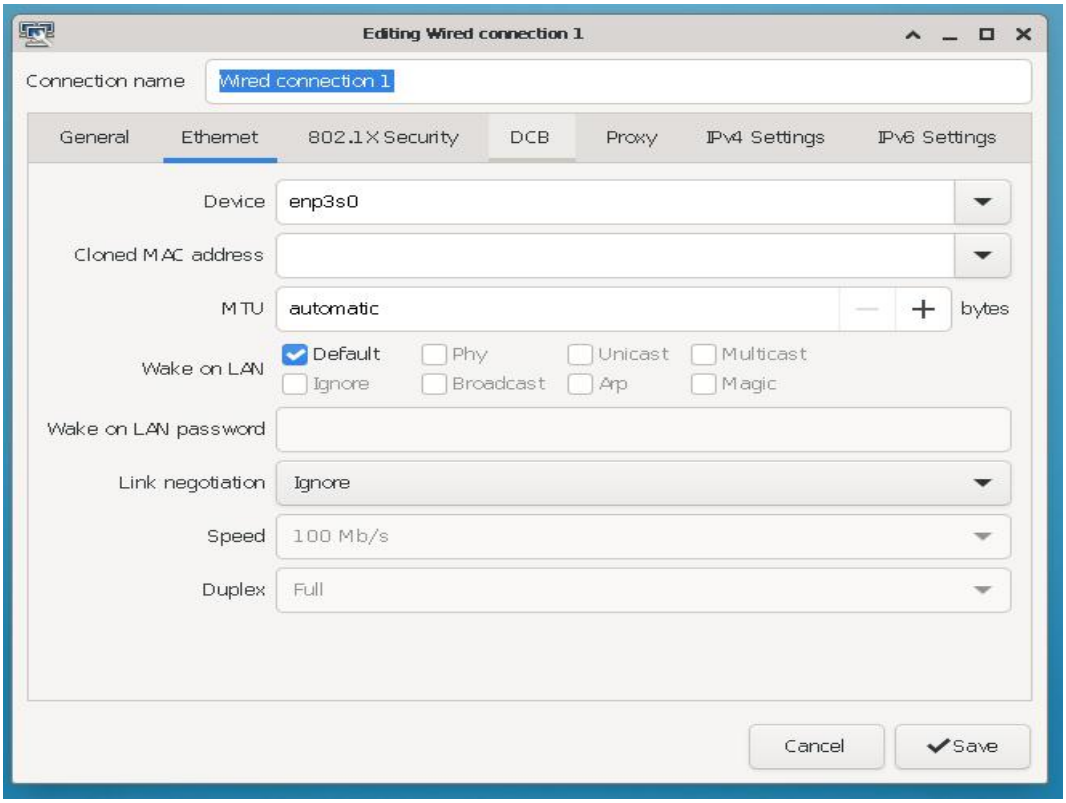

图 6- 3 编辑网络设置

选择 IPV4 Settings 选项卡，如图 6-4 所示。

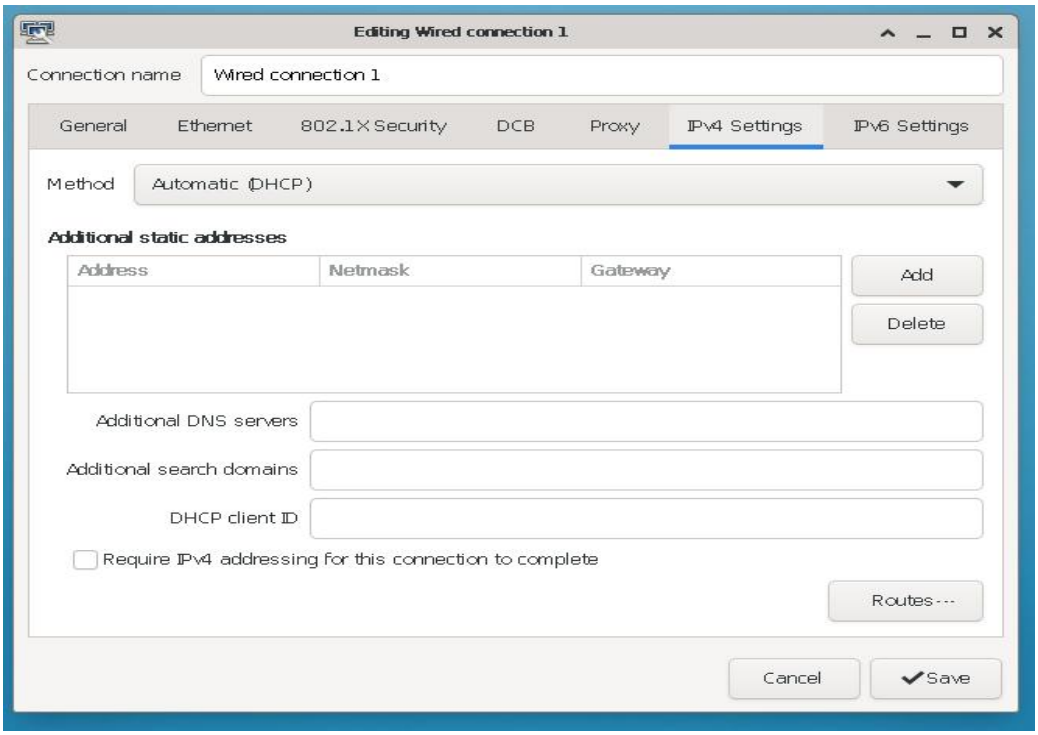

图 6- 4 IPV4 选项卡

Method 设置为 Automatic (DHCP)，注意，以前设置的 ip 需要删除，如图 6-5 所示。

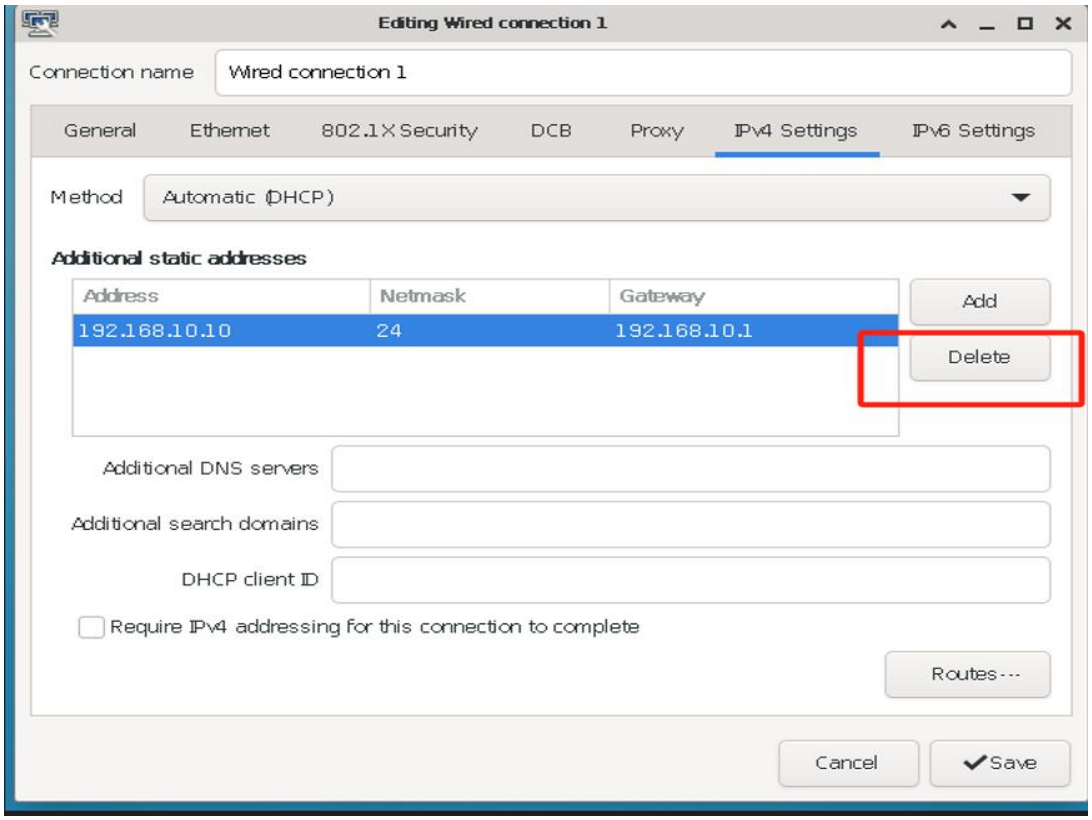

图 6- 5 删除 IP 设置

设置完成后，点击 save 进行保存，点击桌面右上角电脑图标，点击 Disconnect,

断开网络连接，如图 6-6 所示。

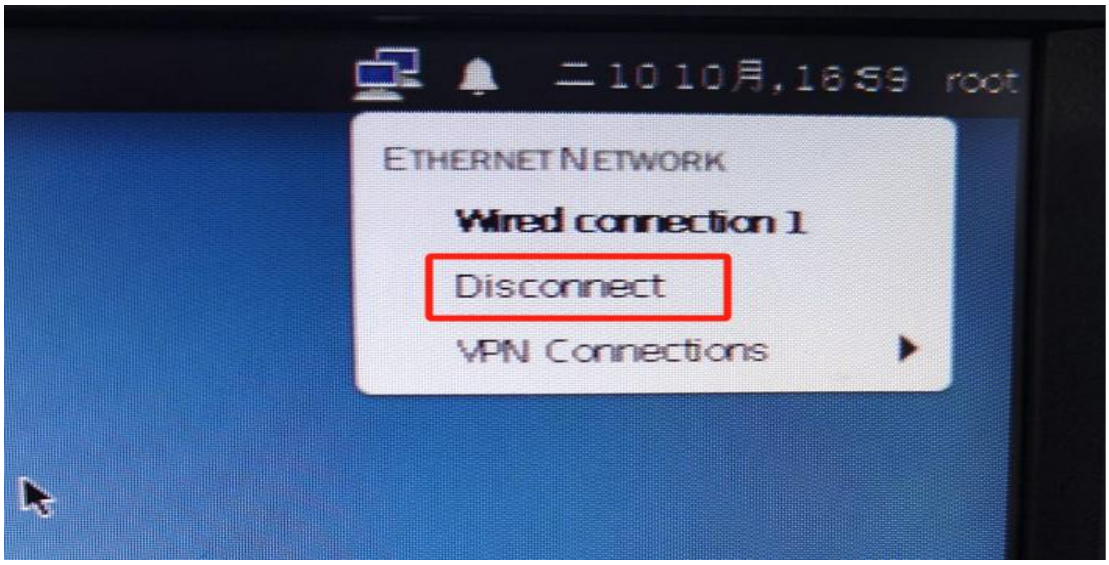

图 6- 6 断开网络

重新点击桌面右上角电脑图标，点击 Wired connect 1,重新连接网络，如图 6-7 所示。

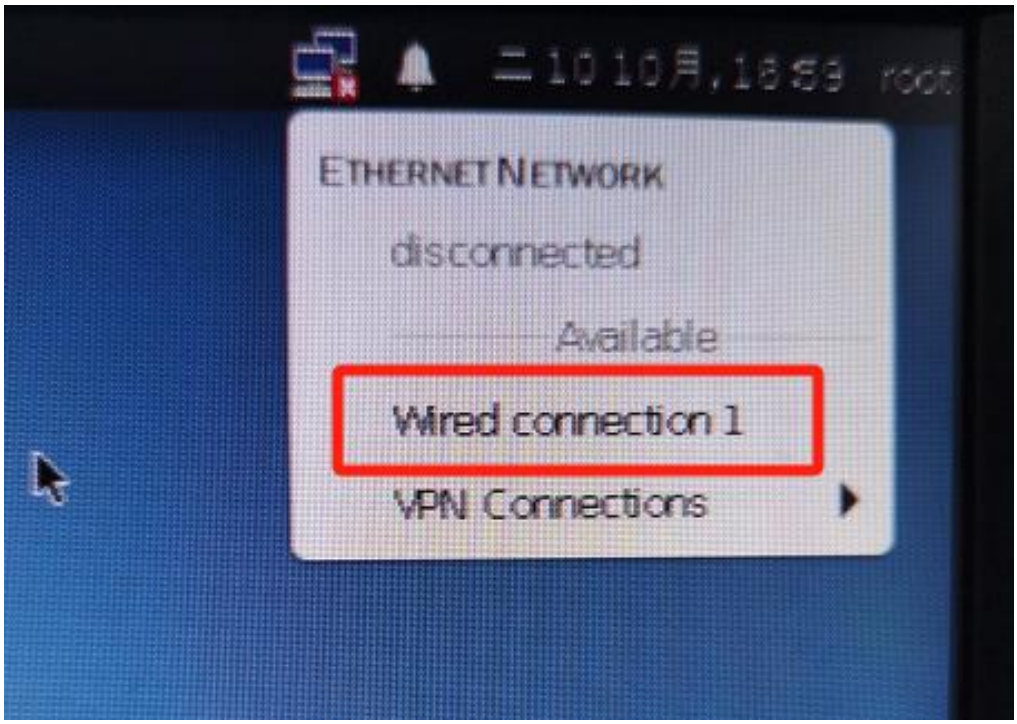

图 6- 7 重连网络

6.2 图形界面配置静态 IP

进入系统后，鼠标右键点击桌面右上角小电脑图标，选择 Edit Connections 进行编辑，如图 6-8 所示，然后弹出图 6-9。

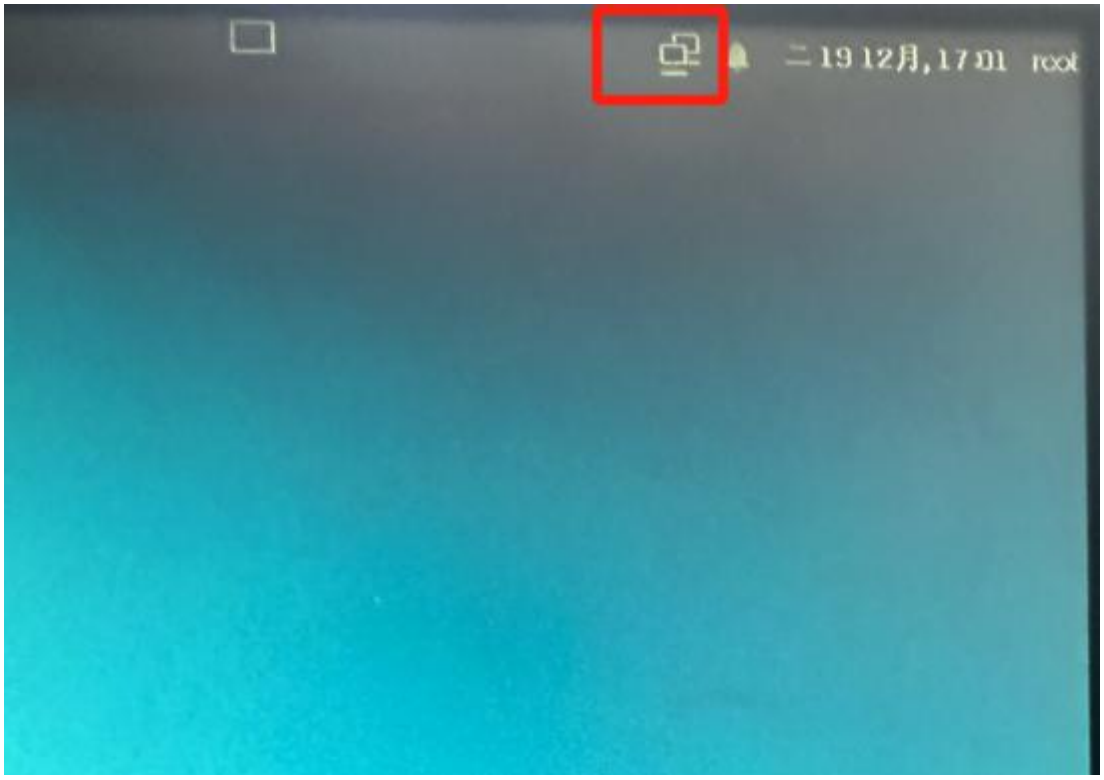

图 6- 8 打开网络设置

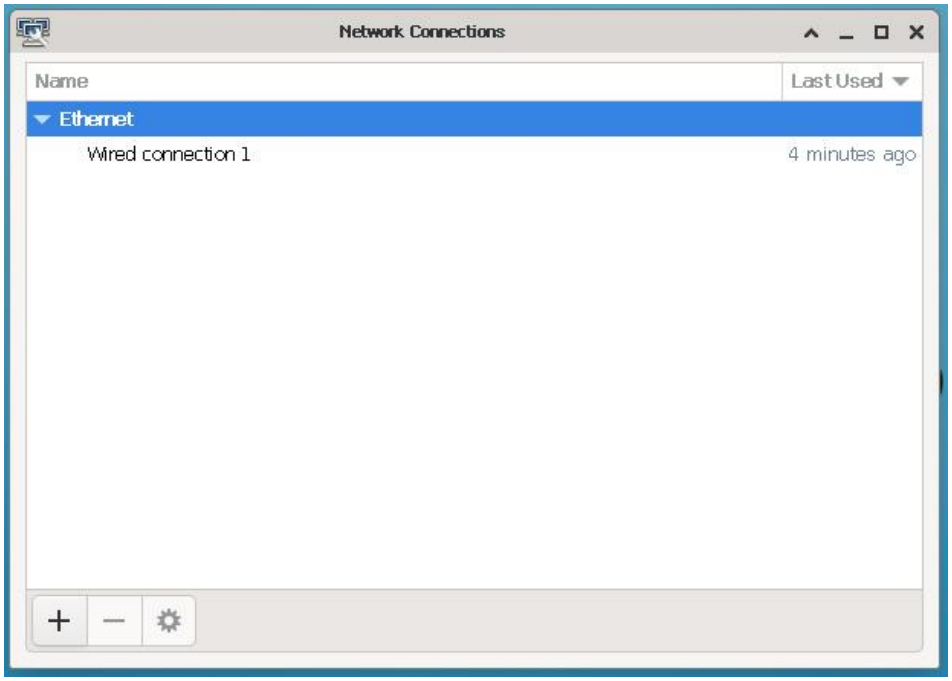

图 6- 9 编辑网络设置

双击 Wired connection 1，进入编辑界面，如图 6-10 所示。

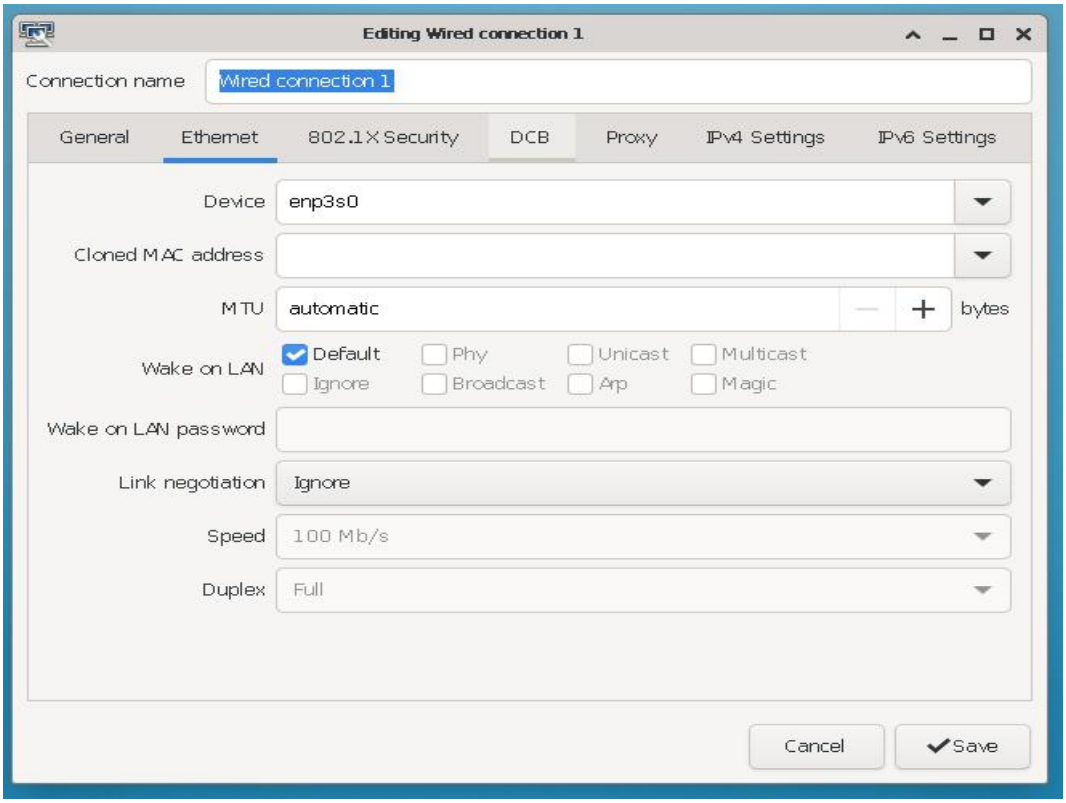

图 6- 10 编辑网络设置

选择 IPV4 Settings 选项卡，如图 6-11 所示。

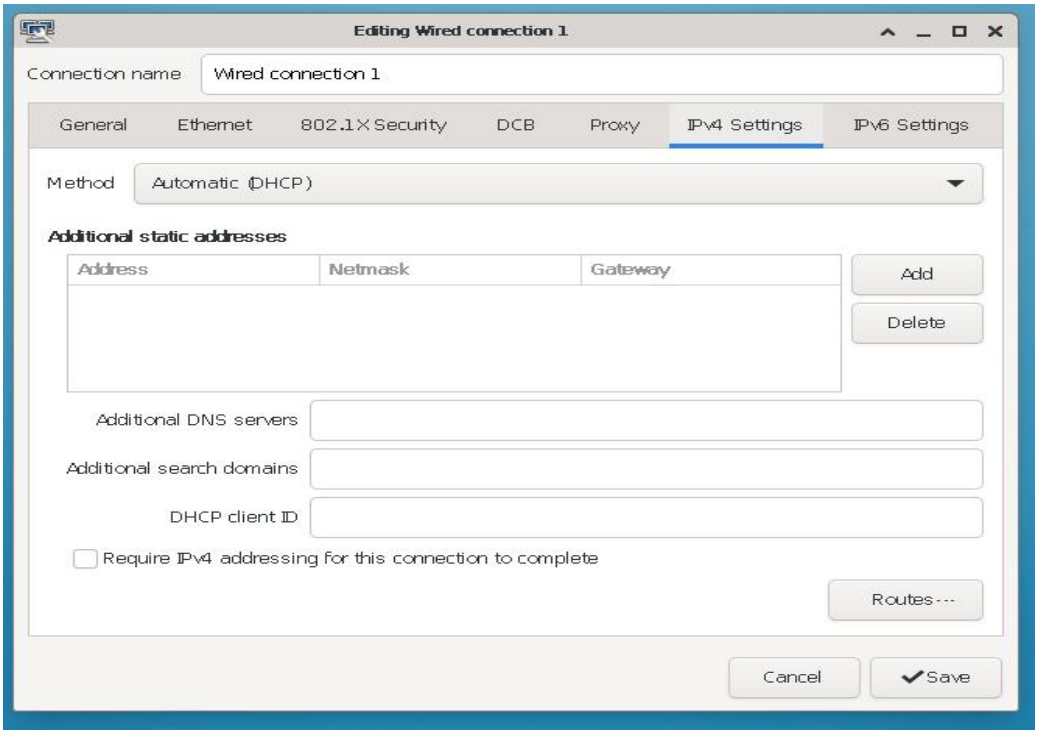

图 6- 11 IPV4 选项卡

Method 设置为 Manual，如图 6-12 所示。

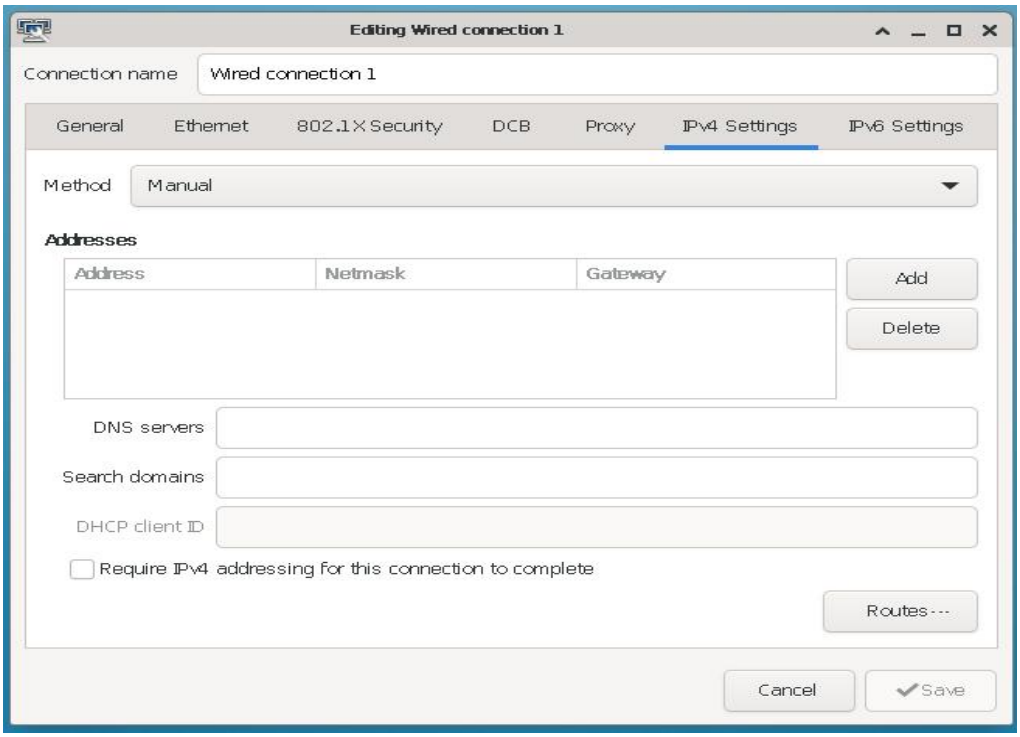

图 6- 12 设置 Method

点击 Add 按钮，添加 IP，点击 Save，如图 6-13 所示。

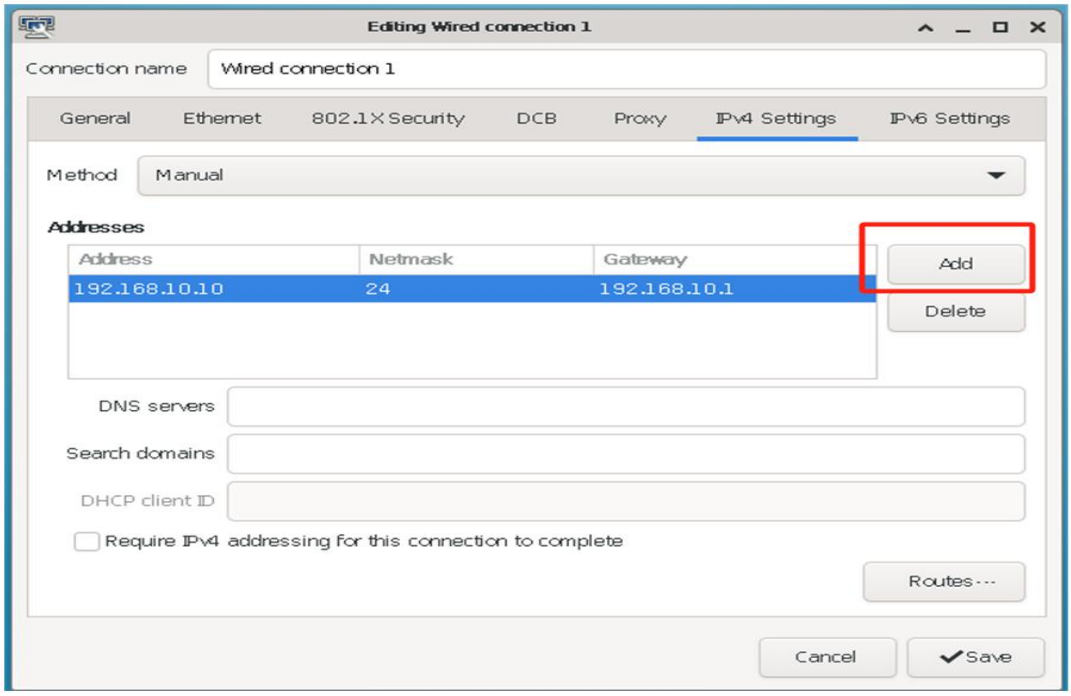

图 6- 13 配置静态 IP

点击桌面右上角电脑图标，点击 Disconnect,断开网络连接，如图 6-14 所示。

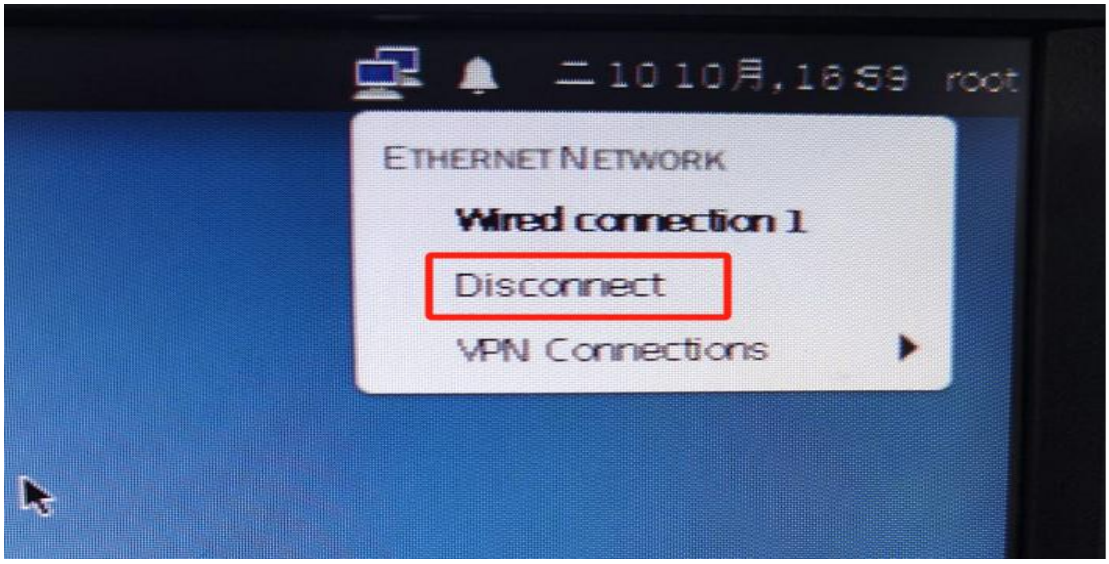

图 6- 14 断开网络

重新点击桌面右上角电脑图标，点击 Wired connect 1，重新连接网络，如图 6-15 所示。

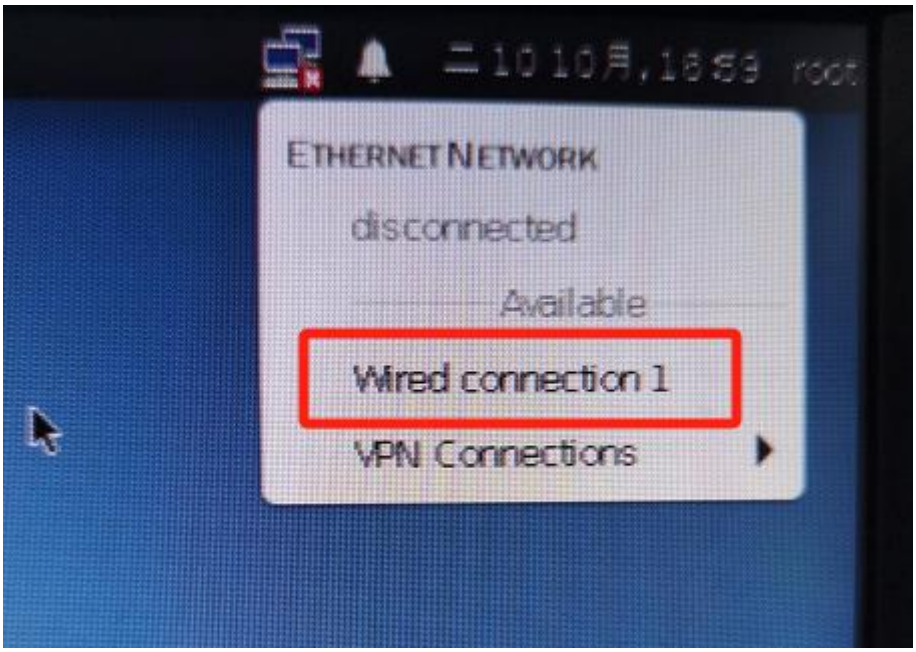

图 6- 15 重连网络

6.3 命令行配置静态 IP

默认接口为自动获取，不会生成配置文件，执行"nmcli connection show"查看当前连接：

图 6- 16 查看当前连接

如果修改当前已存在的连接，则需执行类似如下命令（具体命令需要根据实际需求进行修改）：nmcli c m <连接名称> ipv4.method manual ipv4.addr 192.168.1.30/24 ipv4.gateway 192.168.1.1 ipv4.dns 192.168.1.1

如果网卡当前没有连接，则需要新建连接：nmcli connection add con-name <连接名称> ifname <接口名称，如 eth1> type 'ethernet' ipv4.method 'manual' ipv4.addresses "192.168.1.30/24" ipv4.gateway 192.168.1.1 ipv4.dns 192.168.1.1 以上命令会自动生成配置文件并生效，下次修改 IP 可以直接修改同名配置文件（在/etc/NetworkManager/system-connections/目录下）。

7 软件包在线安装

注：以下命令均需要 root 权限。

1. 更新软件包列表和依赖项：dnf update
2. dnf install <package1> <package2> … …

其他常用命令：dnf search <包名>

8 时间与时区

8.1 时间设置

date 命令用于显示或设置系统的日期和时间。

date -s ‘2024-01-01’可以用来设置日期；date -s ’13:08:55’可以用来设置时间；也可以时间日期一同设置，用空格分开。

8.2 时区设置

timedatectl 是一个用于管理 Linux 系统时间和日期的命令行工具。

查看时区列表：timedatectl list-timezones

设置时区：timedatectl set-timezone Asia/Shanghai（Asia/Shanghai 根据实际需要填写）

8.3 网络时间同步设置

修改 timesyncd.conf 文件，修改内容如图 8-1 红框所示。

vim /etc/systemd/timesyncd.conf

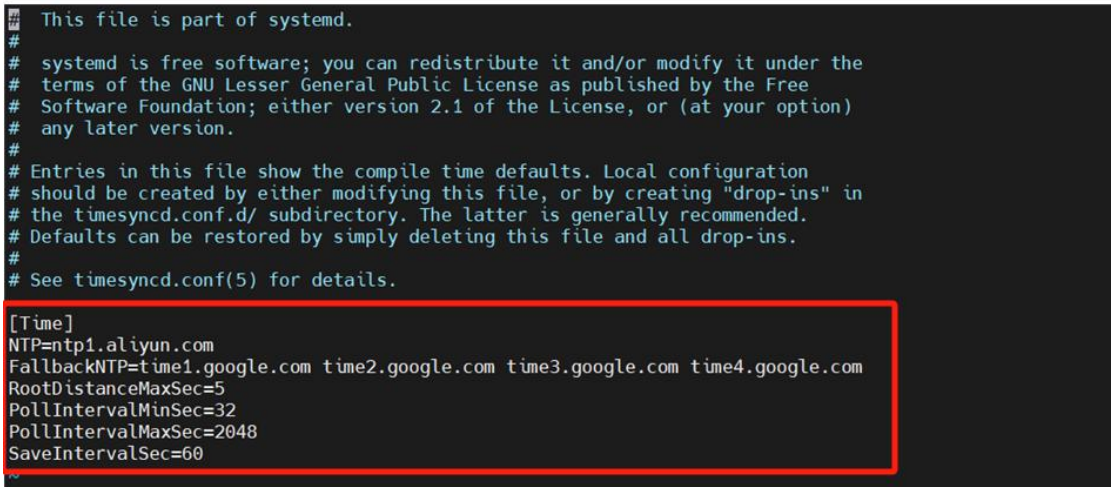

图 8- 1 网络时间同步设置

9 系统备份与还原

9.1 功能概述

- 备份：将当前系统进行备份，最多备份一份，多次备份将会覆盖上次备份。
- 还原：将备份的系统，还原为当前系统。
- 恢复出厂设置：将板卡恢复到出厂时的系统状态。
- 在线升级：将系统包（update.tar.gz）作为新的系统执行。

9.2 功能说明

所有功能，均需执行完特定脚本后，重启生效；执行脚本前，需要将工作目录切换到/etc/startup-wanghuo；执行脚本时，使用 root 用户。

9.2.1 备份

执行命令./backup.sh backup，重启后，将自动备份当前的系统。

9.2.2 还原

执行命令./backup.sh restore_backup，重启后，会把系统还原为已经备份的系统。

9.2.3 恢复出厂

执行命令./backup.sh restore_factory，重启后，会把系统还原出厂时的系统。

9.2.4 升级

将系统升级包 update.tar.gz 放置在与 backup.sh 脚本同一级目录，然后执行./backup.sh update。在 backup.sh 的同级目录下，有一个配置文件 regular.conf，它的作用是，列出了升级时，不被升级包所覆盖的目录或者文件。

比如，在使用系统时，修改了 sshd 的相关配置文件/etc/ssh/sshd_config。在系统升级时，为了避免/etc/ssh/sshd_config 被系统包里的文件所覆盖，可以把该文件路径，填写到 regular.conf 里。

9.2.5 其他说明

删除备份：./backup.sh -r backup
该命令需要慎用，会导致已备份的系统丢失。

清理升级缓存：./backup.sh -r update
有时在执行./backup.sh update 时，会提示删除缓存文件，此时执行此命令即可。

10 开机自启动方法

10.1 开机自启动方法

开机自启动的，可以是脚本、二进制程序、shell 命令等，将其写入/etc/startup-wanghuo/wanghuo-start.sh 该文件即可。

11 系统实时性和压力测试

11.1 实时性测试

cyclictest 是一个用于评估实时系统性能的工具。它主要用于测量系统的时钟精度和实时性能，尤其是在嵌入式系统和实时应用程序中很有用。

下载安装 cyclictest 测试工具，在大多数 Linux 发行版中，cyclictest 通常包含在"rt-tests"软件包中。安装成功后，执行以下命令进行测试。

./cyclictest -a2 -n -p90 -i1000 --policy=fifo --latency=0 -d0 -D 1h

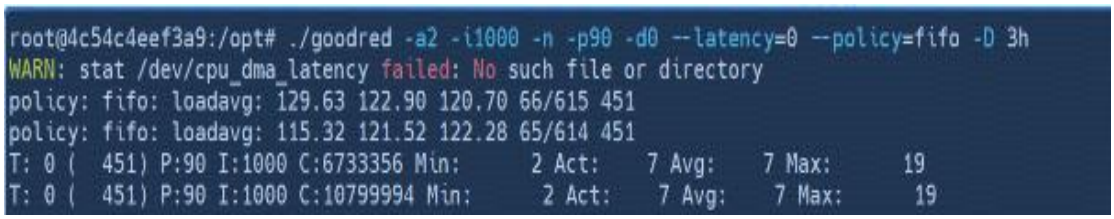

图 11- 1 cyclictest

11.2 压力测试

stress 是一个 Linux 系统上的压力测试工具，用于模拟系统负载，以测试系统在高负载条件下的稳定性和可靠性。

./stress -c 4 -m 10 --vm-bytes 1G -t 60m

图 11- 2 stress

11.3 实时性调优

11.3.1 禁用显卡

使用 lspci 查看显卡的 PCIe 编号，一般会包含 VGA、Graphics 等字样。

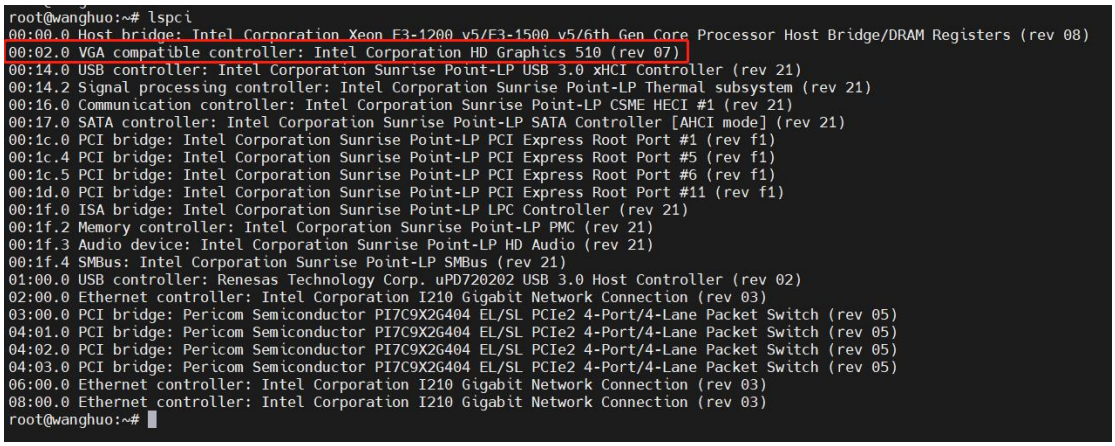

图 11- 3 lspci 命令

然后执行该命令：echo 1 >/sys/devices/pci0000\:00/0000\:00\:02.0/remove。注意，PCIe 编号要根据实际情况填写。

禁用后，poweroff 和 reboot 命令需要加-f 选项才能正常执行。

11.3.2 关闭图形界面

11.3.2.1 Runlevel 介绍

在 Linux 中，Runlevel（运行级别）是指操作系统启动时所处的状态或模式。它定义了哪些系统服务和进程在特定的运行级别下启动或停止。

Linux 中常见的运行级别如下：

0：系统关机。该运行级别意味着系统将关闭所有服务并停止运行。

1：单用户模式。该运行级别提供最小的功能，仅允许一个用户登录系统。通常用于系统修复或维护任务。

2：多用户文本模式。此运行级别下，系统启动到命令行界面，允许多个用户登录并执行命令行操作。这是一个无图形界面的运行级别。

3：多用户文本模式（网络）。类似于运行级别 2，但此级别还启动了网络服务，允许进行远程登录或进行网络相关操作。

4：保留给用户定义。此运行级别未预先配置，留给用户根据需要自定义。

5：多用户图形模式。该运行级别启动了图形界面，允许多个用户登录到具有图形环境的系统。

6：系统重启。在此运行级别下，系统将重新启动。

11.3.2.2 切换运行级别

系统默认启动为图形化界面，runlevel 级别为 5，如图 11-4 所示。

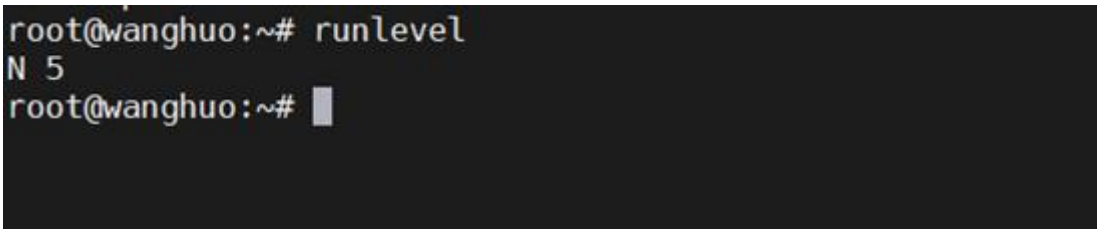

图 11- 4 图形界面的 runlevel

使用 init 3 命令，将图形化切换到命令行模式。

11.3.2.3 更改默认运行级别

系统启动后，默认为图形化界面，运行级别为 5。执行命令 systemctl set-default multi-user.target，默认运行级别为将被设置为 3（重启后生效）。

11.3.3 添加隔离核

11.3.3.1 飞腾版本系统添加隔离核

1. 连接串口设备，开机后，按回车键打断 uboot 自启动，进入 uboot 命令行，输入 printenv，打印信息如图所示。

```
FT2004C#printenv
arch=arm
baudrate=115200
board=FT2004C
board_name=FT2004C
boot_ft=run bootcmd_ft;run load_fdt_ft;run load_kernel_ft;bootm 0x90100000 -:- 0x90000000
boot_lists=usb_root nvme_root tftp
boot_nvme_ramfs=run nvme_ramfs_load;setenv bootargs $nvme_ramfs_bootargs;bootm 0x90100000 0x96000000 0x95000000
boot_nvme_root=run nvme_root_load;setenv bootargs $nvme_root_bootargs;bootm 0x90100000 - 0x95000000
boot_tftp=run tftp_load;setenv bootargs $tftp_bootargs;bootm 0x90100000 0x96000000 0x95000000
boot_usb=run usb_load;setenv bootargs $usb_bootargs;bootm 0x90100000 0x96000000 0x95000000
boot_usb_root=run usb_root_load;setenv bootargs $usb_root_bootargs;bootm 0x90100000 -:- 0x95000000
bootcmd=run distro_bootcmd
bootcmd_ft=setenv bootargs 'console=ttyAMA1,115200 earlycon=pl011,0x28001000 root=/dev/sda5 rw rootwait'
bootdelay=1
cpu=armv8
distro_bootcmd=for medium in $boot_lists;do run boot_$medium;done
eth1addr=98:0e:24:62:72:eb
ethaddr=98:0e:24:62:72:ea
fdtcontroladdr=fae54878
gatewayip=192.168.0.1
ipaddr=192.168.0.111
load_fdt_ft=ext4load scsi 0:1 0x90000000 dtb/ft2004c-devboard-dsk.dtb
load_kernel_ft=ext4load scsi 0:1 0x90100000 uImage
netmask=255.255.255.0
nvme_ramfs_bootargs=console=ttyAMA1,115200 earlycon=pl011,0x28001000 rdinit=/sbin/init rw isolcpus=2,3 FROM=nvme
nvme_ramfs_load=fatload nvme 0:1 90100000 uImage;fatload nvme 0:1 95000000 ft2004.dtb;fatload nvme 0:1 96000000 core.cpio.gz.u-bo
ot
nvme_root_bootargs=console=ttyAMA1,115200 earlycon=pl011,0x28001000 init=/sbin/init rw root=/dev/nvme0n1p2 FROM=nvme_root
nvme_root_load=fatload nvme 0:1 90100000 uImage;fatload nvme 0:1 95000000 ft2004.dtb
serverip=192.168.0.99
stderr=uart1@28001000
stdin=uart1@28001000
stdout=uart1@28001000
tftp_bootargs=console=ttyAMA1,115200 earlycon=pl011,0x28001000 rdinit=/sbin/init rw isolcpus=2,3 FROM=tftp
tftp_load=tftpboot 90100000 uImage;tftpboot 95000000 ft2004.dtb;tftpboot 96000000 core.cpio.gz.u-boot
usb_bootargs=console=ttyAMA1,115200 earlycon=pl011,0x28001000 rdinit=/sbin/init rw isolcpus=2,3 FROM=usb
usb_load=usb start;fatload usb 0:1 90100000 uImage;fatload usb 0:1 95000000 ft2004.dtb;fatload usb 0:1 96000000 core.cpio.gz.u-bo
ot
usb_root_bootargs=console=ttyAMA1,115200 earlycon=pl011,0x28001000 init=/sbin/init root=/dev/sda2 rw rootwait FROM=usb_root
usb_root_load=usb start;fatload usb 0:1 90100000 uImage;fatload usb 0:1 95000000 ft2004.dtb
```

图 11- 5 uboot printenv 输出

2. 使用 setenv，修改隔离核资源配置，在 usb_bootargs 变量后添加“isolcpus=2,

3”，2，3 代表需要的隔离核，根据实际需求替换。以图 11-5 为例追加，输入：setenv usb_bootargs “console=ttyAMA1,115200 earlycon=pl011,0x28001000 rdinit=/sbin/init rw isolcpus=2,3 FROM=usb”，输入 saveenv 进行保存，重启机器。

```
FT2004C#setenv usb_bootargs "console=ttyAMA1,115200 earlycon=pl011,0x28001000 rdinit=/sbin/init rw isolcpus=2,3 FROM=usb"
FT2004C#
```

图 11- 6 设置隔离核

3. 使用 cat /proc/cmdline 查看隔离核是否生效，出现 isolcpus=2,3，则为生效

11.3.3.2 X86 版本系统添加隔离核

1. 系统启动后，进入/boot/EFI/BOOT 目录，对 grub.cfg 文件进行编辑。
2. 找到以 linux /bzImage root=开头的行，在该行末尾添加“isolcpus=2”，退出保存，重启机器。具体数值根据具体的需求进行设置。

```
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
default=boot
timeout=5
menuentry 'boot'{
linux /bzImage root=PARTUUID=1f86f4a2-bf6c-4e57-8fde-eda01c8dc693 rootwait  rootfstype=ext4  rootwait console=ttyS0,115200 console=tty0 isolcpus=2
initrd /core-image-minimal-initramfs-intel-corei7-64.cpio.gz
}
~
```

图 11- 7 设置隔离核

3. 使用 cat /proc/cmdline 查看隔离核是否生效，出现 isolcpus=2，则为生效。

11.3.3.3 瑞芯微版本系统添加隔离核

1. 系统启动后，进入/boot/extlinux 目录，对 extlinux.conf 文件进行编辑。添加”isolcpus=2,3”，2，3 代表需要的隔离核，根据实际需求进行修改。

```
# Generic Distro Configuration file generated by OpenEmbedded
LABEL WangHuo
        KERNEL /boot/Image
        FDT /boot/rk3588-toybrick-x0-linux.dtb
        APPEND root=/dev/mmcblk0p2 rw rootwait rootfstype=ext4 earlycon=uart8250,mmio32,0xfeb50000 console=ttyFIQ0 isolcpus=2,3 r
rqchip.gicv3_pseudo_nmi=0
~
```

图 11- 8 修改 extlinux.conf

2. 重启机器，使用 cat /proc/cmdline 查看隔离核是否生效，出现 isolcpus=2,3，则为生效。

11.3.3.4 ZYNQ 版本系统添加隔离核

1. 进入系统，列出系统所有块设备，找到系统启动分区(挂载点/为根文件系统，

/的上一个分区即为启动分区)。

```
root@wanghuo:~# lsblk
NAME         MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
mtdblock0     31:0    0   32M  0 disk
mmcblk0      179:0    0   30G  0 disk
├─mmcblk0p1  179:1    0    1G  0 part
└─mmcblk0p2  179:2    0   29G  0 part /
mmcblk1      179:8    0  7.3G  0 disk
mmcblk1boot0 179:16   0 31.9M  1 disk
mmcblk1boot1 179:24   0 31.9M  1 disk
```

图 11- 9 查找启动分区

2. 将系统启动分区挂载到/mnt，编辑里面的 uENV.txt

```
root@wanghuo:~# cd /mnt/
root@wanghuo:/mnt# ls
BOOT.bin  milianke-system.dtb  uEnv.txt  uImage
root@wanghuo:/mnt# vim uEnv.txt
```

图 11- 10 打开 uENV.txt

3. 找到以 bootargs 开头的一行，在该行末尾添加“isolcpus=1”，1 代表需要隔离的核，可根据实际情况修改。

```
machine_name=milianke
kernel_image=uImage
kernel_load_address=0x2080000
devicetree_image=milianke-system.dtb
devicetree_load_address=0x2000000
bootargs=earlyprintk console=ttyPS0,115200 root=/dev/mmcblk0p2 rw rootwait isolcpus=1
loadkernel=fatload mmc 0 ${kernel_load_address} ${kernel_image}
loaddtb=fatload mmc 0 ${devicetree_load_address} ${devicetree_image}
bootkernel=run loadkernel && run loaddtb && bootm ${kernel_load_address} - ${devicetree_load_address}
uenvcmd=run bootkernel
~
```

图 11- 11 添加 isolcpus

4. 退出保存，重启机器，使用 cat /proc/cmdline 查看隔离核是否生效，出现 isolcpus=1，则为生效。

12 其他

更多资源参见 https://www.onewos.com/或联系 wanghuo@ucas.com.cn 获取。