

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN ĐIỆN TỬ-VIỄN THÔNG**



**HỆ ĐIỀU HÀNH**

**ĐỀ TÀI: TÌM HIỂU KIT NANOPC-T3. TÌM HIỂU, BIÊN  
DỊCH KERNEL, U-BOOT, CẤU HÌNH MẠNG TRONG  
KERNEL VÀ CẤU HÌNH MẠNG TỪ XA CHO KIT**

**GVHD: TS Phạm Doãn Tĩnh**

**Nhóm thực hiện: Nhóm 5 lớp 109209**

**Nguyễn Minh Hiếu (20151336) – Điện tử 3 K60**

**Đỗ Hoàng Phước (20152920) – Điện tử 3 K60**

**Nguyễn Tiến Đạt (20140985) – Điện tử 9 K59**

**Tổng Văn Khánh (20151991) – Điện tử 7 K60**

**Nguyễn Hưng Tuyên (20154163) – Điện tử 5 K60**

**Hà Nội, tháng 5 năm 2019**

## LỜI MỞ ĐẦU

Trong các thiết bị điện tử ngày nay, hệ điều hành là phần mềm hệ thống không thể thiếu để thiết bị có thể vận hành. Các máy tính thường sử dụng hệ điều hành Windows hoặc Linux và các biến thể như Ubuntu, Debian, các điện thoại thông minh chủ yếu sử dụng hai hệ điều hành Android và iOS. Để tìm hiểu thêm về hệ điều hành cho kit, nhóm em đã chọn đề tài “Tìm hiểu kit NanoPC-T3. Tìm hiểu, biên dịch kernel U-Boot, cấu hình mạng trong kernel và cấu hình mạng từ xa cho kit” làm bài tập lớn môn hệ điều hành.

Mục tiêu của đề tài là bước đầu giúp nhóm làm quen với kit NanoPC-T3, tìm hiểu, biên dịch kernel và U-Boot, nghiên cứu cách cấu hình mạng từ xa qua wifi. Trong quá trình thực hiện khó tránh khỏi những sai sót nhất định, rất mong nhận được những ý kiến đóng góp từ thầy giáo để bọn em có thêm kinh nghiệm.

Chúng em xin cảm ơn thầy Phạm Doãn Tĩnh đã cho nhóm mượn kit NanoPC-T3 và gợi ý đề tài bài tập lớn cho nhóm em.

## MỤC LỤC

LỜI MỞ ĐẦU .....	2
Chương 1. Tổng quan kit NanoPC-T3 .....	8
1.1 Giới thiệu kit NanoPC-T3.....	8
1.2 Cấu trúc phần cứng .....	9
1.3 Các hệ điều hành tương thích .....	10
Chương 2. Cài hệ điều hành cho kit NanoPC-T3 .....	11
2.1 Chuẩn bị.....	11
2.2 Thực hiện .....	11
2.3 Sử dụng thẻ nhớ boot cho kit.....	14
Chương 3. Tìm hiểu và biên dịch kernel cho kit NanoPC-T3 .....	15
3.1 Tìm hiểu kernel.....	15
3.1.1 Khái niệm Kernel .....	15
3.1.2 Vai trò của kernel .....	15
3.1.3 Mục đích biên dịch Kernel .....	16
3.1.4 Vị trí các file kernel.....	16
3.2 Biên dịch Kernel cho kit .....	17
Chương 4. Tìm hiểu và biên dịch U-Boot cho kit NanoPC-T3 .....	20
4.1 Tìm hiểu U-Boot.....	20
4.2 Biên dịch U-Boot cho kit.....	20
Chương 5. Cấu hình kernel cho network .....	23
5.1 Các khái niệm cơ bản.....	23
5.2 Cấu hình network trong kernel .....	23

5.2.1 Networking support .....	23
5.2.2 Bộ lọc mạng Netfilter .....	23
5.2.3 Các driver mạng .....	24
5.2.4 IrDA.....	25
5.2.5 Bluetooth .....	25
5.2.6 Mạng không dây .....	26
<b>Chương 6. Phát wifi trên kit NanoPC–T3.....</b>	<b>27</b>
6.1 Trên Lubuntu .....	27
6.2 Trên Debian/Friendly Core.....	27
<b>Chương 7. Cấu hình mạng từ xa cho kit NanoPC–T3 .....</b>	<b>29</b>
7.1 Kết nối tới kit bằng giao thức SSH.....	29
7.2 Cấu hình mạng cho kit.....	30
7.2.1 Trình soạn thảo vi.....	30
7.2.2 Cấu hình mạng Domain Resolution .....	30
7.2.3 Cấu hình mạng trên Lubuntu/Debian .....	32
<b>KẾT LUẬN .....</b>	<b>34</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>35</b>

## DANH MỤC HÌNH VẼ

Hình 1.1 Tổng thể kit NanoPC-T3 .....	8
Hình 1.2 Mặt trên kit NanoPC-T3.....	9
Hình 1.3 Mặt dưới kit NanoPC-T3 .....	9
Hình 2.1 Phần mềm Win32 Disk Imager .....	11
Hình 2.2: Cửa sổ cài đặt hệ điều hành cho kit .....	12
Hình 2.3 Quá trình cài hệ điều hành Lubuntu.....	12
Hình 2.4 Giao diện hệ điều hành Lubuntu trên kit.....	13
Hình 2.5 Phiên bản hệ điều hành Lubuntu được cài .....	13
Hình 2.6 Giao diện hệ điều hành Lubuntu boot từ thẻ nhớ.....	14
Hình 3.1 Kernel Linux .....	15
Hình 3.2 File .bashrc .....	17
Hình 3.3 Cấu hình Kernel .....	18
Hình 3.4 Quá trình biên dịch kernel (bắt đầu) .....	18
Hình 3.5 Quá trình biên dịch kernel (kết thúc) .....	19
Hình 3.6 Phân vùng boot.....	19
Hình 3.7 Kiểm tra phiên bản kernel .....	19
Hình 4.1 Cấu hình U-Boot .....	20
Hình 4.2 Quá trình biên dịch U-Boot (bắt đầu) .....	21
Hình 4.3 Quá trình biên dịch U-Boot (kết thúc) .....	21
Hình 4.4 Kiểm tra phiên bản U-Boot.....	22
Hình 5.1 Networking support.....	23
Hình 5.2 Netfilter Configuration.....	23
Hình 5.3 PCI support.....	24
Hình 5.4 Network device support .....	24
Hình 5.5 Ethernet driver support.....	24
Hình 5.6 IrDA subsystem support.....	25
Hình 5.7 Các giao thức IrDA .....	25

Hình 5.8 Các driver cho Bluetooth .....	25
Hình 5.9 IEEE 802.11 Networking Stack .....	26
Hình 5.10 Các driver mạng không dây .....	26
Hình 5.11 Driver USB Network Adapter.....	26
Hình 6.1 Phát wifi trên Lubuntu .....	27
Hình 6.2 Lỗi khi thiết lập wifi-AP .....	27
Hình 7.1 Xác định địa chỉ IP của kit .....	29
Hình 7.2 Truy cập tới kit bằng SSH.....	29
Hình 7.3 Chỉnh sửa file bằng trình soạn thảo vi .....	30
Hình 7.4 File /etc/resolv.conf.....	30
Hình 7.5 File /etc/hosts .....	31
Hình 7.6 File /etc/nsswitch.conf.....	31
Hình 7.7 File /etc/network/interfaces.....	32

## **DANH MỤC BẢNG BIỂU**

Bảng 1.1 Cấu trúc phần cứng kit NanoPC-T3.....	9
--	---

## Chương 1. Tổng quan kit NanoPC-T3

### 1.1 Giới thiệu kit NanoPC-T3

Kit NanoPC-T3 được FriendlyARM thiết kế và phát triển cho người dùng chuyên nghiệp và doanh nghiệp. Nó sử dụng Samsung Octa-Core Cortex-A53 S5P6818 SoC. So với FriendlyARM NanoPC-T2, NanoPC-T3 không chỉ có tất cả các giao diện và cổng T2, mà còn có SoC mạnh hơn. Tần số động của nó có quy mô từ 400M đến 1,4GHz. NanoPC-T3 có 8G eMMC trên bo mạch, jack âm thanh, giao diện đầu vào/đầu ra video, WiFi, Bluetooth và cổng Ethernet Gbps tích hợp. Ngoài ra, NanoPC-T3 có trình quản lý năng lượng, anten sứ onboard và cổng serial debug. Kit cũng có một bộ tản nhiệt có lỗ lắp để tránh quá nhiệt. Hình 1 là tổng thể kit NanoPC-T3.



**Hình 1.1 Tổng thể kit NanoPC-T3**

NanoPC-T3 có hai giao diện camera: giao diện camera DVP và MIPI-CSI, bốn giao diện video: HDMI 1.4A, LVDS, RGB-LCD song song và giao diện MIPI-DSI. Kit hỗ trợ RTC và có các chân giao diện RTC. Kit cũng có bốn cổng USB với hai cổng loại A và hai cổng đầu pin 2,54mm.

NanoPC-T3 hỗ trợ các hệ điều hành đơn giản, ví dụ: Android5.1, Debian và UbuntuCore + Qt. Đây là một dự án nguồn mở với các giao diện và cổng phong phú. Nó được sinh ra là một sự lựa chọn cho người dùng chuyên nghiệp và doanh nghiệp.



## 1.2 Cấu trúc phần cứng

Bảng 1.1 là cấu trúc phần cứng của kit NanoPC-T3. Hình 1.2 và 1.3 là mặt trên và mặt dưới của kit.

**Bảng 1.1 Cấu trúc phần cứng kit NanoPC-T3**

Thông tin	Mô tả
Vi xử lý	Samsung S5P6818 Octa-Core Cortex-A53 hoạt động với xung nhịp 400MHz-1,4GHz
Điều khiển nguồn	Vi điều khiển AXP228, được tích hợp phần mềm để tắt và đèn báo khi mở
RAM	1GB 32bit DDR3
Bộ nhớ	eMMC 8GB và 1 khe thẻ nhớ SD
Ethernet	Gbps Ethernet Port (RTL8211E)
Wifi	802.11b/g/n
Bluetooth	4.0 dual mode
USB	2 cổng USB standard type A
Âm thanh	1 jack 3.5mm cho âm thanh và 1 microphone
Kết nối đầu ra	1 cổng HDMI cho đầu ra âm thanh / video số
Các chân GPIO	30 chân GPIO 2,54mm
Nguồn cung cấp	Nguồn 1 chiều 5V/2A
Nhiệt độ hoạt động	-40°C đến 80°C



**Hình 1.2 Mặt trên kit NanoPC-T3**



**Hình 1.3 Mặt dưới kit NanoPC-T3**

## 1.3 Các hệ điều hành tương thích

- UbuntuCore
  - npi-config: tiện ích cấu hình hệ thống để đặt mật khẩu, ngôn ngữ, múi giờ, tên máy chủ, SSH và tự động đăng nhập, bật/tắt I2C, spi, serial và PWM
  - Trình quản trị mạng
  - Đầu ra ghi nhật ký hệ thống từ cổng serial
  - Cửa sổ welcome với thông tin hệ thống cơ bản và trạng thái
  - Tự động đăng nhập bằng tài khoản người dùng "pi" với quyền truy cập vào npi-config
  - UART2 đã bật
  - hỗ trợ CAM500B
- Debian
  - Hỗ trợ CAM500B
- Android
  - Hỗ trợ thiết lập IP tĩnh
  - Hỗ trợ truy cập phân cứng với libfriendlyarm-things.so của FriendlyElec
  - Tiện ích iTest tích hợp để kiểm tra phân cứng

## Chương 2. Cài hệ điều hành cho kit NanoPC-T3

### 2.1 Chuẩn bị

Để cài hệ điều hành cho kit NanoPC-T3, nhóm em chuẩn bị các thứ sau:

- Kit NanoPC-T3
- 1 thẻ nhớ SD class 10 hoặc hơn
- Nguồn điện 5V/2A
- Màn hình có kết nối HDMI
- Bàn phím và chuột USB (USB hub nếu có)
- Một máy tính được cài đặt Ubuntu 64 bit phiên bản 18.04

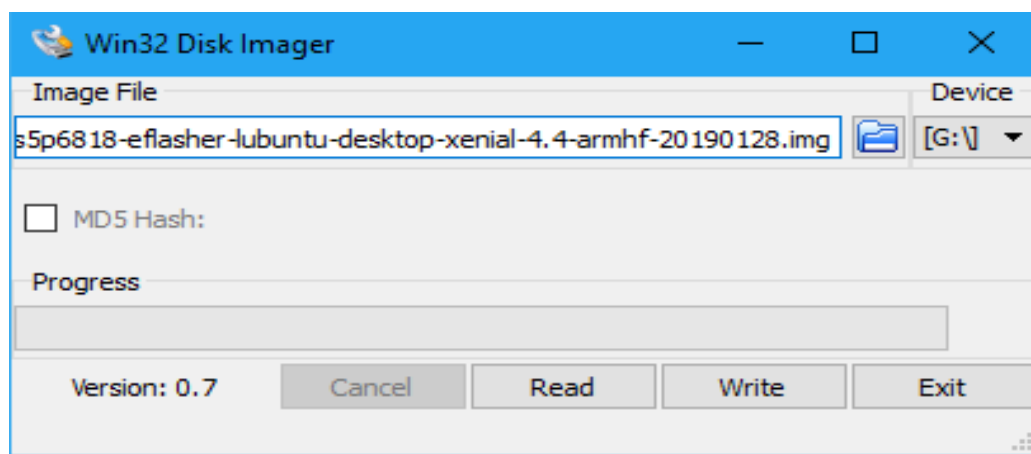
### 2.2 Thực hiện

Nhóm em cài hệ điều hành Lubuntu cho kit theo các bước sau:

- Bước 1: Tải phần mềm **Win 32 Disk Imager** và file **s5p6818-eflasher-lubuntu-desktop-xenial-4.4-armhf-20190128.img.zip** tại:

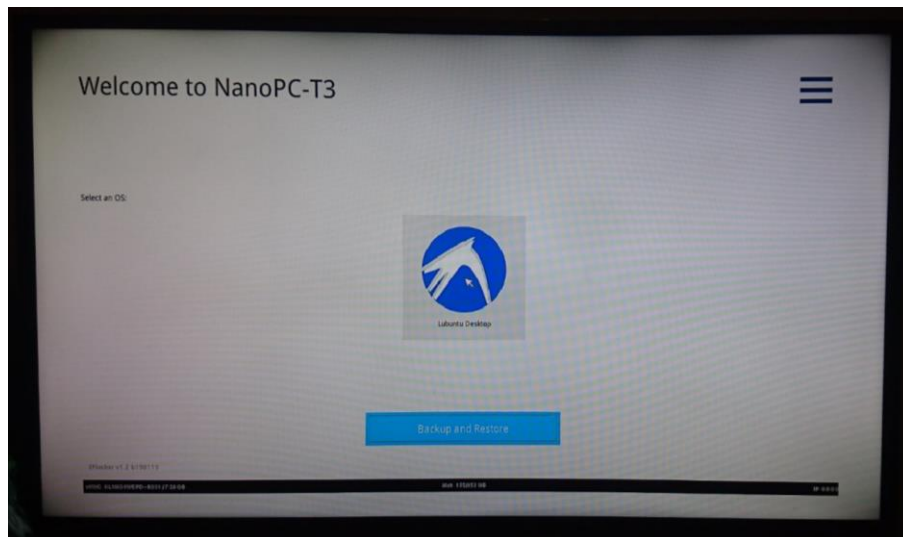
[https://drive.google.com/drive/folders/1hGZVao7vpj\\_0BtaN\\_HZPs2GNr35E0if](https://drive.google.com/drive/folders/1hGZVao7vpj_0BtaN_HZPs2GNr35E0if)

- Bước 2: Giải nén file đã tải về.
- Bước 3: Cắm thẻ nhớ vào máy tính và bật phần mềm **Win32diskimager** với quyền Administrator, sau đó chọn ổ của thẻ SD vừa cắm, chọn file vừa giải nén để ghi vào thẻ nhớ rồi ấn nút "**Write**" và chờ đến khi hoàn thành. Hình 2.1 là giao diện phần mềm Win32 Disk Imager

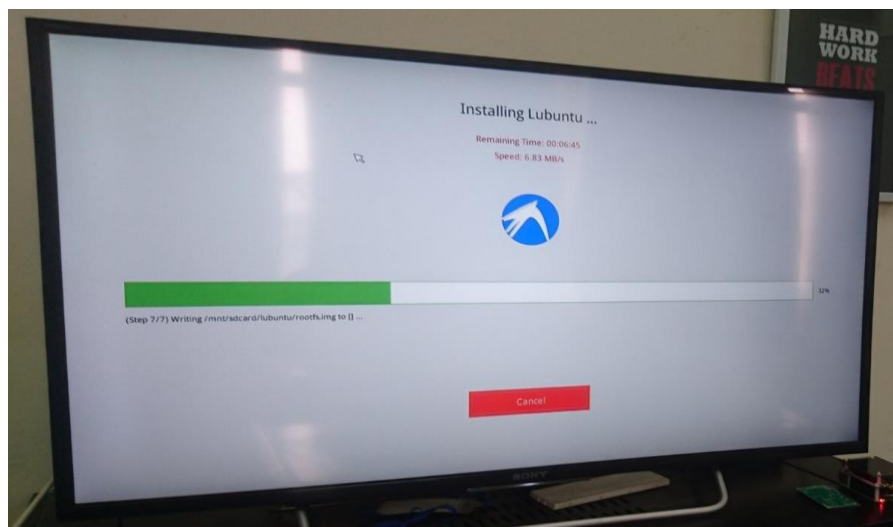


Hình 2.1 Phần mềm Win32 Disk Imager

- Bước 4: Rút thẻ nhớ ra rồi cắm vào khe thẻ nhớ trên kit, kết nối kit với một màn hình qua HDMI, sau đó ấn giữ nút "Boot" (ở bên cạnh cổng Ethernet) và gạt công tắc nguồn
- Bước 5: Một cửa sổ cài đặt hiện lên, sau khi hoàn thành hệ điều hành sẽ được cài đặt trên bộ nhớ eMMC của kit. Hình 2.2 và 2.3 là quá trình cài hệ điều hành Ubuntu cho kit

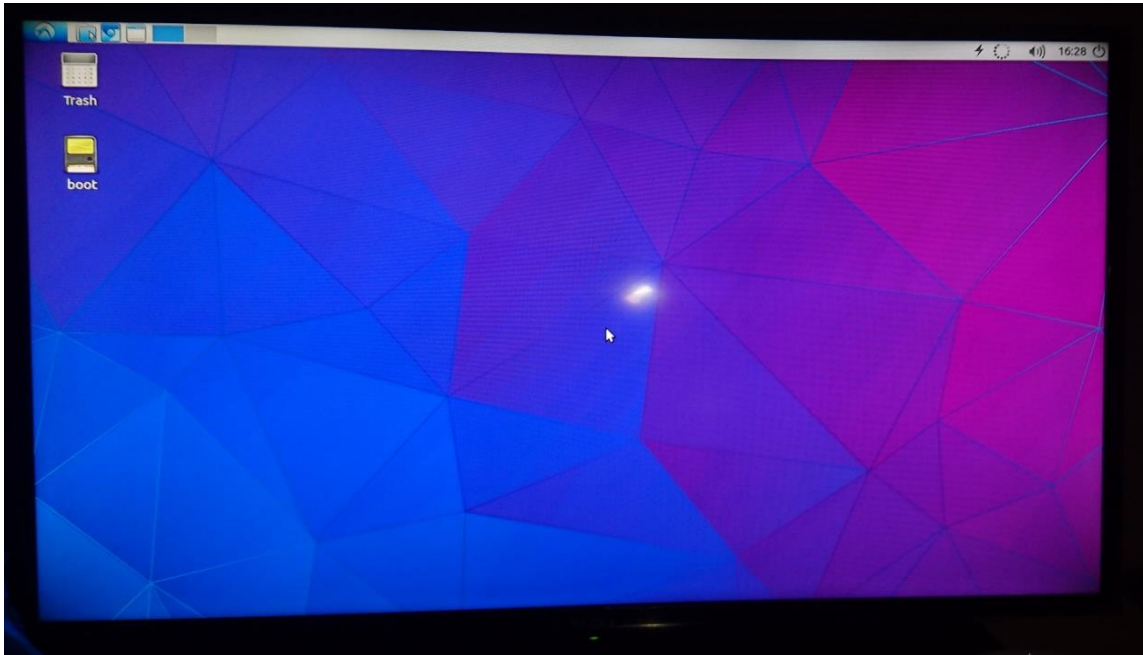


**Hình 2.2: Cửa sổ cài đặt hệ điều hành cho kit**

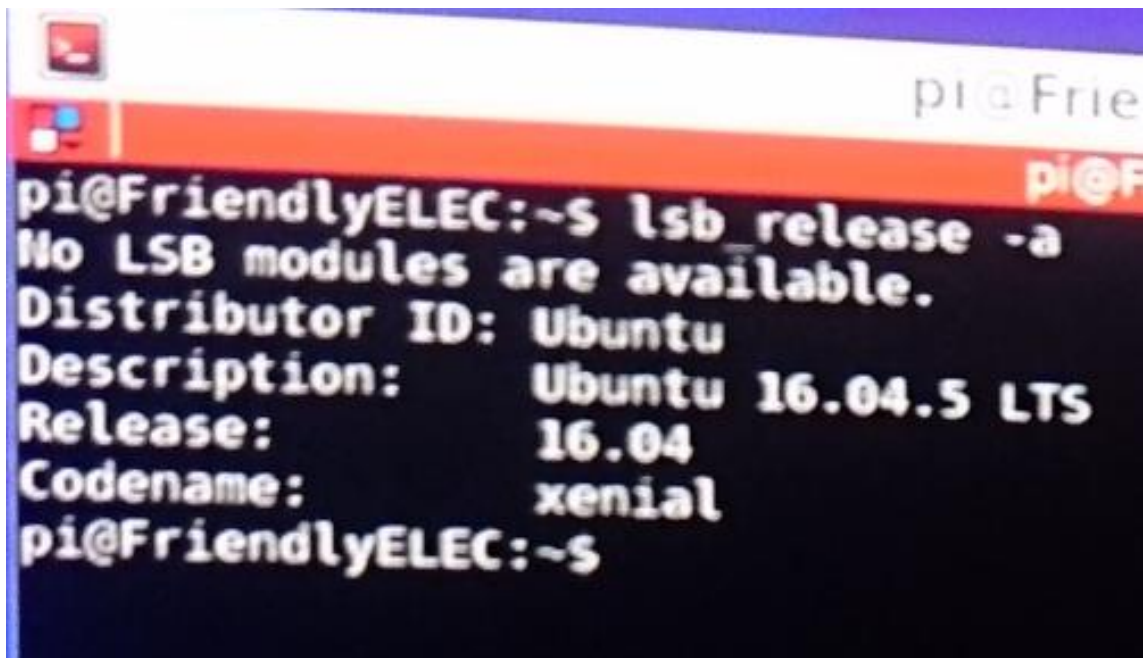


**Hình 2.3 Quá trình cài hệ điều hành Ubuntu**

- Bước 6: Sau khi cài xong thì từ đó mỗi lần bật kit sẽ khởi động hệ điều hành từ bộ nhớ eMMC. Hình 2.4 là giao diện hệ điều hành Lubuntu, hình 2.5 là phiên bản Lubuntu cài trên kit



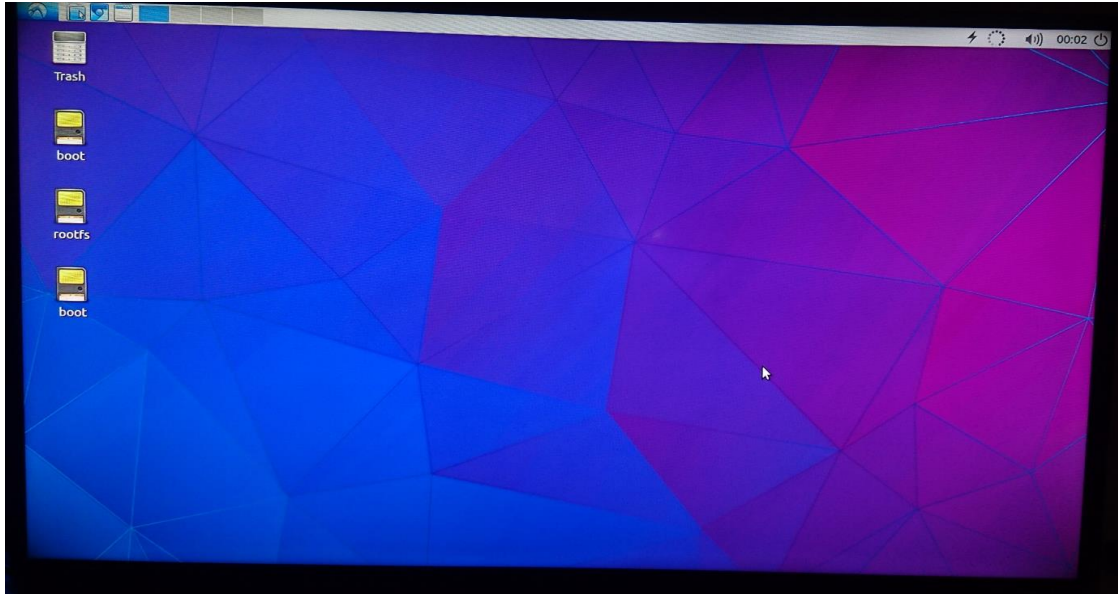
Hình 2.4 Giao diện hệ điều hành Lubuntu trên kit



Hình 2.5 Phiên bản hệ điều hành Lubuntu được cài

## 2.3 Sử dụng thẻ nhớ boot cho kit

Để boot cho kit từ thẻ nhớ, nhóm em làm giống các bước ở phần cài hệ điều hành cho kit, nhưng lúc tải file img về thì chọn file s5p6818-sd-lubuntu-desktop-xenial-4.4-armhf-20190128.img.zip. Ngoài ra còn có cách khác để ghi thẻ nhớ trên Ubuntu nhóm em sẽ trình bày ở phần biên dịch U-Boot. Hình 2.6 là giao diện hệ điều hành Lubuntu boot từ thẻ nhớ.



**Hình 2.6** Giao diện hệ điều hành Lubuntu boot từ thẻ nhớ



## Chương 3. Tìm hiểu và biên dịch kernel cho kit NanoPC-T3

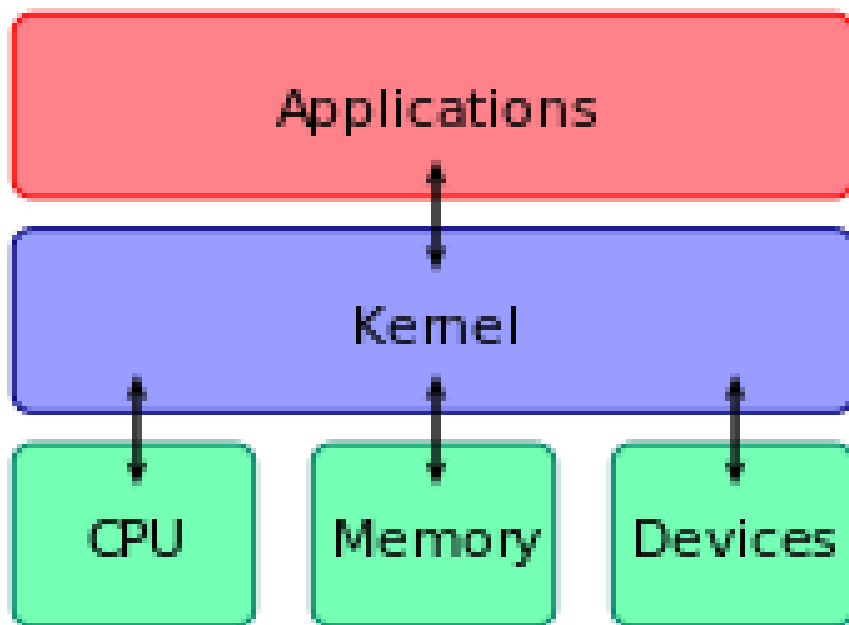
### 3.1 Tìm hiểu kernel

#### 3.1.1 Khái niệm Kernel

Khái niệm kernel ở đây nói đến những phần mềm, ứng dụng ở mức thấp (low-level) trong hệ thống, có khả năng thay đổi linh hoạt để phù hợp với phần cứng. Chúng tương tác với tất cả ứng dụng và hoạt động trong chế độ user mode, cho phép các quá trình khác – hay còn gọi là server, nhận thông tin từ các thành phần khác qua inter-process communication (IPC). Hình 3.1 là vị trí của kernel trong hệ điều hành.

#### 3.1.2 Vai trò của kernel

- Quản lý tất cả các tài nguyên phần cứng
- Cung cấp việc cài đặt đa nền tảng, kiến trúc, phần cứng độc lập
- Xử lý việc truy cập và sử dụng các thành phần phần cứng một cách đồng thời từ các nguồn tài nguyên khác nhau



Hình 3.1 Kernel Linux

### 3.1.3 Mục đích biên dịch Kernel

- Tái biên dịch kernel để chữa lỗi của kernel. Nếu các lỗi này thuộc về lỗi của kernel thì phải vá nguồn của kernel và biên dịch lại nó để sửa chữa các lỗi được công bố
- Biên dịch lại nhân để loại bỏ những driver không cần thiết qua đó nâng cao hiệu năng của nhân
- Biên dịch lại kernel để thử nghiệm một chức năng hoặc một module mới

### 3.1.4 Vị trí các file kernel

Các file kernel này, trong Ubuntu chúng được lưu trữ tại thư mục /boot và đặt tên theo vmlinuz-version. Khi bộ nhớ ảo bắt đầu được phát triển để thực hiện các tác vụ đa luồng, tiền tố vm sẽ được đặt vào đầu các file kernel để phân biệt khả năng hỗ trợ công nghệ ảo hóa. Kể từ đó, Linux kernel được gọi là vmlinux, nhưng hệ thống kernel này đã phát triển với tốc độ quá nhanh, lớn hơn so với dung lượng bộ nhớ boot chuẩn của hệ điều hành, vì vậy những file kernel này đã được nén theo chuẩn zlib – và ký tự z được thêm vào là do như vậy. Ngoài ra còn 1 số định dạng nén thường gặp khác là LZMA hoặc BZIP2, nhưng chúng vẫn được gọi chung là zImage. Các phiên bản được sắp xếp thứ tự theo định dạng A.B.C.D, trong đó A.B thường là 2.6, C đại diện cho phiên bản, và D là ký hiệu các bản vá lỗi hoặc patch.

Trong thư mục /boot còn có rất nhiều file quan trọng khác như, initrd.img-version, system.map-version, và config-version. File initrd được dùng như 1 ổ đĩa RAM để giải nén và kích hoạt các file kernel thực sự, còn file system.map được dùng để quản lý bộ nhớ trước khi kernel được tải đầy đủ, và file config làm nhiệm vụ thông báo cho kernel biết những lựa chọn hoặc module nào sẽ được nạp vào quá trình hệ thống khởi động.



### 3.2 Biên dịch Kernel cho kit

Nhóm em thực hiện biên dịch kernel 4.4.49 cho kit theo các bước:

- Bước 1: Cài đặt **git**  
`sudo apt install git`
- Bước 2: Cài đặt bộ biên dịch chéo toolchain
  - Cài đặt **aarch64-linux-gcc 6.4**:

```
git clone https://github.com/friendlyarm/prebuilts.git
```

```
sudo mkdir -p /opt/FriendlyARM/toolchain
```

```
sudo tar xf prebuilts/gcc-x64/aarch64-cortexa53-linux-gnu-6.4.tar.xz -C  
/opt/FriendlyARM/toolchain/
```

- Cài đặt arm-linux-gcc 4.9.3

```
git clone https://github.com/friendlyarm/prebuilts.git
```

```
sudo mkdir -p /opt/FriendlyARM/toolchain
```

```
sudo tar xf prebuilts/gcc-x64/arm-cortexa9-linux-gnueabi-4.9.3.tar.xz -C  
/opt/FriendlyARM/toolchain/
```

- Bước 3 : Thêm các dòng sau vào cuối file .bashrc (hình 3.3)  
“export PATH=/opt/FriendlyARM/toolchain/6.4-aarch64/bin:\$PATH”  
“export GCC\_COLORS=auto”  
“export PATH=/opt/FriendlyARM/toolchain/4.9.3/bin:\$PATH”  
“export GCC\_COLORS=auto”

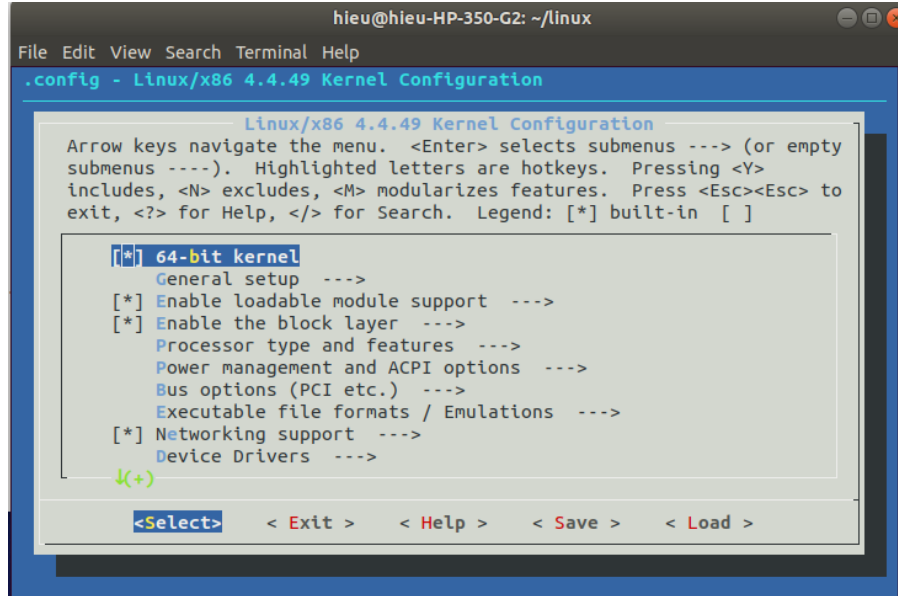
```
export PATH=/opt/FriendlyARM/toolchain/6.4-aarch64/bin:$PATH  
export GCC_COLORS=auto  
export PATH=/opt/FriendlyARM/toolchain/4.9.3/bin:$PATH  
export GCC_COLORS=auto
```

sh ▼ Tab Width: 8 ▼ Ln 121, Col 23 ▼ INS

Hình 3.2 File .bashrc

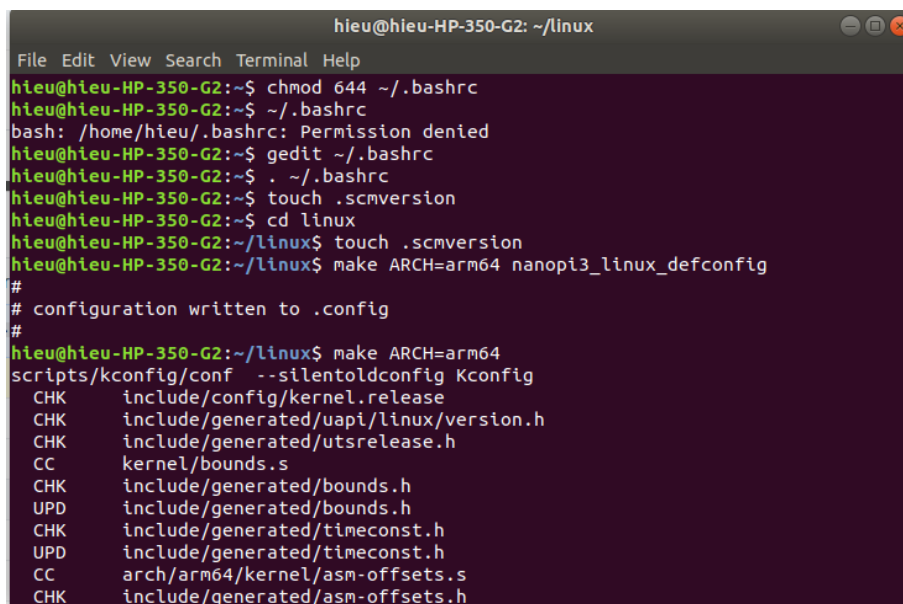
- Bước 5: Thực thi file .bashrc  
`. ~/.bashrc`
- Bước 6: Tải kernel  
`git clone https://github.com/friendlyarm/linux.git -b nanopi2-v4.4.y --depth 1`  
`cd linux`  
`touch .scmversion`

- Bước 7: Tạo file cấu hình kernel  
`make ARCH=arm64 nanopi3_linux_defconfig`
- Bước 8: Cấu hình kernel chuẩn bị biên dịch (hình 3.4)  
`make menuconfig`



Hình 3.3 Cấu hình Kernel

- Bước 9: Bắt đầu biên dịch (hình 3.5 và 3.6)  
`make ARCH=arm64`



Hình 3.4 Quá trình biên dịch kernel (bắt đầu)

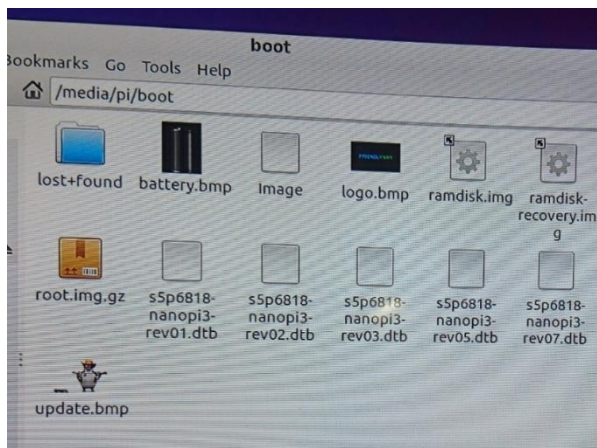
```

hieu@hieu-HP-350-G2: ~/linux
File Edit View Search Terminal Help
CC      sound/usb/caiaq/snd-usb-caiaq.mod.o
LD [M]  sound/usb/caiaq/snd-usb-caiaq.ko
CC      sound/usb/misc/snd-ua101.mod.o
LD [M]  sound/usb/misc/snd-ua101.ko
CC      sound/usb/snd-usb-audio.mod.o
LD [M]  sound/usb/snd-usb-audio.ko
CC      sound/usb/snd-usbmidi-lib.mod.o
LD [M]  sound/usb/snd-usbmidi-lib.ko
IHEX    firmware/ti_3410.fw
IHEX    firmware/ti_5052.fw
IHEX    firmware/mts_cdma.fw
IHEX    firmware/mts_gsm.fw
IHEX    firmware/mts_edge.fw
HOSTCC   firmware/ihex2fw
H16TOFW  firmware/edgeport/boot.fw
H16TOFW  firmware/edgeport/boot2.fw
H16TOFW  firmware/edgeport/down.fw
H16TOFW  firmware/edgeport/down2.fw
IHEX     firmware/edgeport/down3.bin
IHEX2FW  firmware/whiteheat_loader.fw
IHEX2FW  firmware/whiteheat.fw
IHEX2FW  firmware/keyspan_pda/keyspan_pda.fw
IHEX2FW  firmware/keyspan_pda/xircom_pgs.fw
hieu@hieu-HP-350-G2:~/linux$

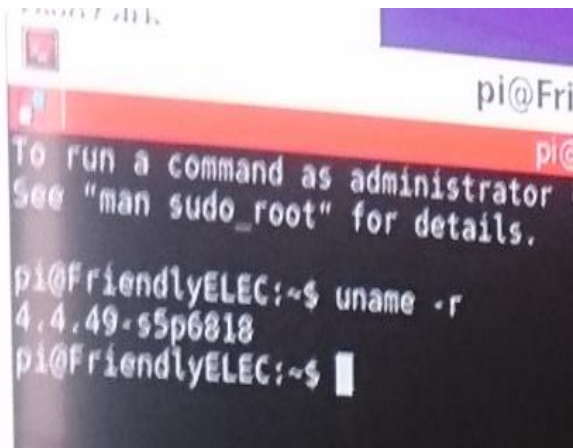
```

**Hình 3.5** Quá trình biên dịch kernel (kết thúc)

- Bước 10: Quá trình biên dịch kết thúc, nhóm em sao chép file Image ở thư mục arch/arm64/boot và các file .dtb có tên dạng s5p6818-nanopi2-rev\*.dtb ở thư mục arch/arm/boot/dts/nexell vào một chiếc USB
- Bước 11: Cắm USB vào kit NanoPC-T3 rồi sao chép các file trên vào phân vùng Boot ở trên bộ nhớ eMMC (hình 3.6)
- Bước 12: Khởi động lại kit, hoàn thành việc cập nhật kernel cho kit (hình 3.7)



**Hình 3.6** Phân vùng boot



**Hình 3.7** Kiểm tra phiên bản kernel

## Chương 4. Tìm hiểu và biên dịch U-Boot cho kit NanoPC-T3

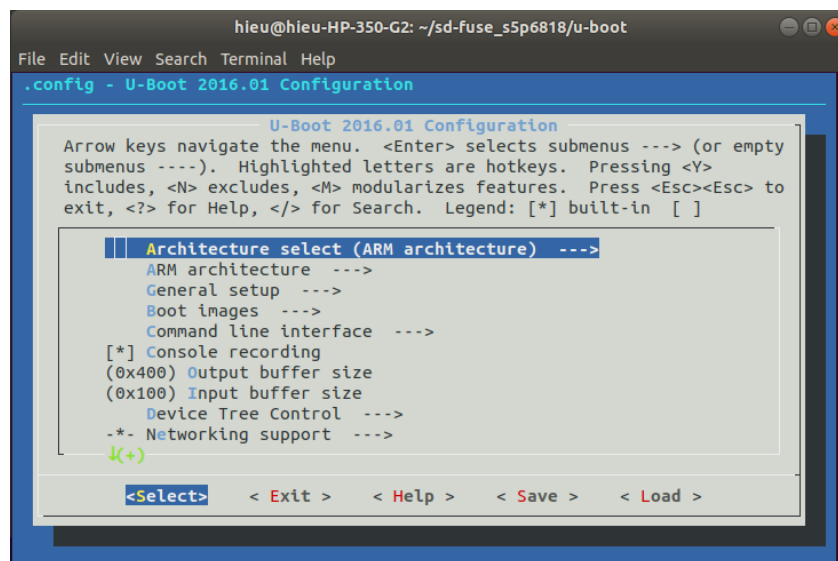
### 4.1 Tìm hiểu U-Boot

Das U-Boot (the Universal Boot Loader), thường gọi tắt là U-Boot, là boot loader chính, nguồn mở được sử dụng trong các thiết bị nhúng để đóng gói các lệnh để khởi động kernel hệ điều hành. Nó khả dụng với rất nhiều kiến trúc máy tính gồm 68k, ARM, Blackfin, MicroBlaze, MIPS, Nios, SuperH, PPC, RISC-V and x86.

### 4.2 Biên dịch U-Boot cho kit

Nhóm em thực hiện biên dịch U-Boot v2016.01 cho kit theo các bước sau:

- Bước 1: Chuẩn bị: Tải file img của hệ điều hành Lubuntu  
`git clone https://github.com/friendlyarm/sd-fuse\_s5p6818.git`  
`cd sd-fuse_s5p6818`  
`wget http://112.124.9.243/dvdfiles/S5P6818/images-for-eflasher/lubuntu-desktop-images.tgz`  
`tar xzf lubuntu-desktop-images.tgz`
- Bước 2: Tải U-boot  
`git clone https://github.com/friendlyarm/u-boot.git`  
`cd u-boot`  
`git checkout nanopi2-v2016.01`
- Bước 3: Tạo file cấu hình U-Boot  
`make s5p6818_nanopi3_defconfig`
- Bước 4: Cấu hình U-Boot chuẩn bị biên dịch (hình 4.1)  
`make menuconfig`



Hình 4.1 Cấu hình U-Boot

- Bước 5: Bắt đầu biên dịch (hình 4.2 và 4.3)  
`make CROSS_COMPILE=aarch64-linux-`

```

hieui@hieui-HP-350-G2: ~/sd-fuse_s5p6818
File Edit View Search Terminal Help
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/conf
#
# configuration written to .config
#
hieui@hieui-HP-350-G2:~/sd-fuse_s5p6818/u-boot$ export PATH=/opt/FriendlyARM/toolc
hain/6.4-aarch64/bin:$PATH
hieui@hieui-HP-350-G2:~/sd-fuse_s5p6818/u-boot$ make CROSS_COMPILE=aarch64-linux-
scripts/kconfig/conf --silentoldconfig Kconfig
CHK include/config.h
UPD include/config.h
GEN include/autoconf.mk
GEN include/autoconf.mk.dep
CHK include/config/uboot.release
UPD include/config/uboot.release
CHK include/generated/version_autogenerated.h
UPD include/generated/version_autogenerated.h
CHK include/generated/timestamp_autogenerated.h
UPD include/generated/timestamp_autogenerated.h
CC lib/asm-offsets.s
CHK include/generated/generic-asm-offsets.h
UPD include/generated/generic-asm-offsets.h
CC arch/arm/lib/asm-offsets.s

```

**Hình 4.2 Quá trình biên dịch U-Boot (bắt đầu)**

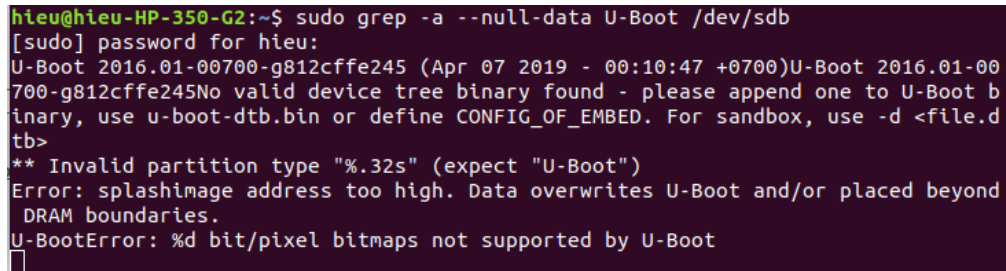
```

hieui@hieui-HP-350-G2: ~/u-boot
File Edit View Search Terminal Help
CC test/dm/cmd_dm.o
LD test/dm/built-in.o
CC examples/standalone/stubs.o
LD examples/standalone/libstubs.o
CC examples/standalone/hello_world.o
LD examples/standalone/hello_world
OBJCOPY examples/standalone/hello_world.srec
OBJCOPY examples/standalone/hello_world.bin
LDS u-boot.lds
LD u-boot
OBJCOPY u-boot.srec
OBJCOPY u-boot.bin
start=$(aarch64-linux-nm u-boot | grep __rel_dyn_start | cut -f 1 -d ' '); end=$(
(aarch64-linux-nm u-boot | grep __rel_dyn_end | cut -f 1 -d ' '); tools/relocate
-rela u-boot.bin 0x43C00000 $start $end
CFG u-boot.cfg
Firmware Image Package ToC:
-----
- Non-Trusted Firmware BL33: offset=0x60, size=0x739F0
file: 'u-boot.bin'
-----
Creating "fip-nonsecure.bin"
Creating "fip-nonsecure.img" (<-- tools/nexell/nsih/nanopi3.txt)

```

**Hình 4.3 Quá trình biên dịch U-Boot (kết thúc)**

- Bước 6: Copy file fip-nonsecure.img vào thư mục lubuntu  
`cp fip-nonsecure.img ../lubuntu/`
- Bước 7: Kiểm tra phiên bản U-Boot (hình 4.4)  
`sudo grep -a -null-data U-Boot /dev/sdb`



```

hieu@hieu-HP-350-G2:~$ sudo grep -a -null-data U-Boot /dev/sdb
[sudo] password for hieu:
U-Boot 2016.01-00700-g812cffe245 (Apr 07 2019 - 00:10:47 +0700)U-Boot 2016.01-00
700-g812cffe245No valid device tree binary found - please append one to U-Boot b
inary, use u-boot-dtb.bin or define CONFIG_OF_EMBED. For sandbox, use -d <file.d
tb>
** Invalid partition type "%.32s" (expect "U-Boot")
Error: splashimage address too high. Data overwrites U-Boot and/or placed beyond
DRAM boundaries.
U-BootError: %d bit/pixel bitmaps not supported by U-Boot

```

**Hình 4.4 Kiểm tra phiên bản U-Boot**

- Bước 8: Cắm thẻ nhớ vào kit và boot từ thẻ nhớ

## Chương 5. Cấu hình kernel cho network

### 5.1 Các khái niệm cơ bản

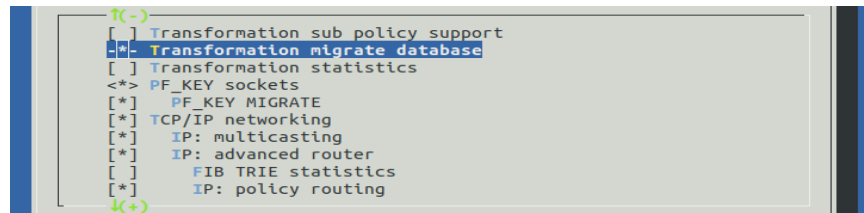
Trước tiên nhóm em tìm hiểu các khái niệm cơ bản về mạng:

- IPv4: Giao thức mạng phiên bản thứ 4. Hầu hết các máy chủ và máy tính dùng IPv4. Nó dùng 32 bit gán địa chỉ mạng, lên đến 255.255.255.255, mỗi số 8 bit
- IPv6: Giao thức mạng phiên bản thứ 6. Nó sử dụng địa chỉ 128 bit, bổ sung các tính năng dịch vụ và bảo mật mới. Linux hỗ trợ IPv6 nhưng dùng IPv4 nổi mạng
- TCP/IP (Transmission Control Protocol/Internet Protocol) : Sử dụng mô hình máy chủ – khách để liên lạc. TCP/IP xác định các gói dữ liệu truyền đi, xác minh tính toàn vẹn dữ liệu, giao thức kết nối và phân hồi, truyền lại
- MAC Address (Media Access Control) : Địa chỉ card mạng được sử dụng để liên lạc giữa các thiết bị trên mạng con. Địa chỉ MAC được xác định duy nhất mỗi node mạng, được sử dụng bởi giao thức Ethernet

### 5.2 Cấu hình network trong kernel

#### 5.2.1 Networking support

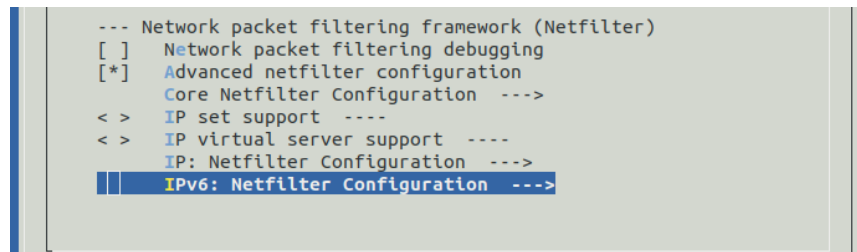
Để Linux hỗ trợ mạng, đầu tiên bắt buộc chọn Networking support, chọn Networking option, chọn TCP/IP networking (hình 5.1)



Hình 5.1 Networking support

#### 5.2.2 Bộ lọc mạng Netfilter

Netfilter là một framework để lọc và điều khiển các gói mạng đi qua máy. Nó thường được dùng khi bật tường lửa hoặc khi sử dụng máy làm proxy cho các máy khác trên mạng. Để chọn Netfilter, cần chọn Network packet filtering framework, để chỉnh sửa chọn Netfilter Configuration (hình 5.2).



Hình 5.2 Netfilter Configuration



### 5.2.3 Các driver mạng

Linux hỗ trợ rất nhiều thiết bị mạng khác nhau, phổ biến nhất là PCI (Peripheral Component Interconnect), có thể cắm cáp Ethernet. PCI support được bật trong kernel ở Bus Option.

```
[*] PCI support
[*] ISA-style DMA support (NEW)
< > PCCard (PCMCIA/CardBus) support (NEW) ----
[ ] Mark VGA/VBE/EFI FB as generic system framebuffer (NEW)
```

Hình 5.3 PCI support

Để hỗ trợ các thiết bị mạng, cần bật Network device support trong Device Drivers (hình 5.4)

```
↑(-)
[*] Block devices --->
    Misc devices --->
< > ATA/ATAPI/MFM/RLL support (DEPRECATED) (NEW) ----
    SCSI device support --->
< > Serial ATA and Parallel ATA drivers (libata) ----
[*] Multiple devices driver support (RAID and LVM) --->
< > Generic Target Core Mod (TCM) and ConfigFS Infrastructure --
[ ] Macintosh device drivers (NEW) ----
[*] Network device support --->
[ ] Open-Channel SSD target support ----
↓(+)
```

Hình 5.4 Network device support

Để tìm các driver cho Ethernet cần vào trong Ethernet driver Support (hình 5.5).

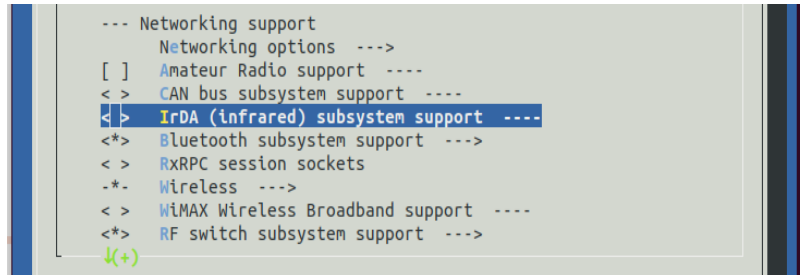
```
↑(-)
< > Virtual netlink monitoring device
    *** CAIF transport drivers ***
    Distributed Switch Architecture drivers ----
[*] Ethernet driver support --->
    -*- PHY Device support and infrastructure --->
< > Micrel KS8995MA 5-ports 10/100 managed Ethernet switch
<*> PPP (point-to-point protocol) support
<*> PPP BSD-Compress compression
<*> PPP Deflate compression
[*] PPP filtering
↓(+)
```

Hình 5.5 Ethernet driver support



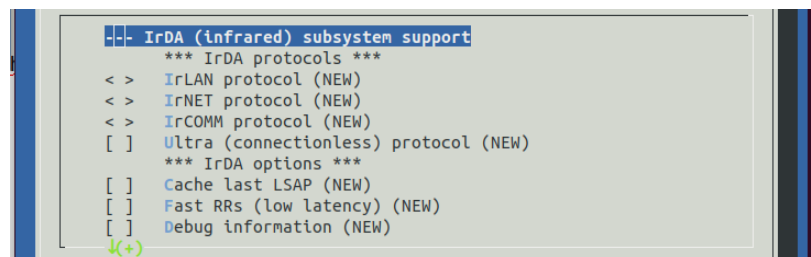
### 5.2.4 IrDA

IrDA (Infrared Data Association) là giao thức hồng ngoại được sử dụng ở một số laptop và PDAs (Personal Digital Assistant) để giao tiếp ở khoảng cách rất ngắn, thường được dùng trên các phần cứng cũ, các phần cứng mới dùng Bluetooth thay thế. Để hỗ trợ IrDA cần chọn IrDA subsystem support trong Networking support (hình 5.6).



Hình 5.6 IrDA subsystem support

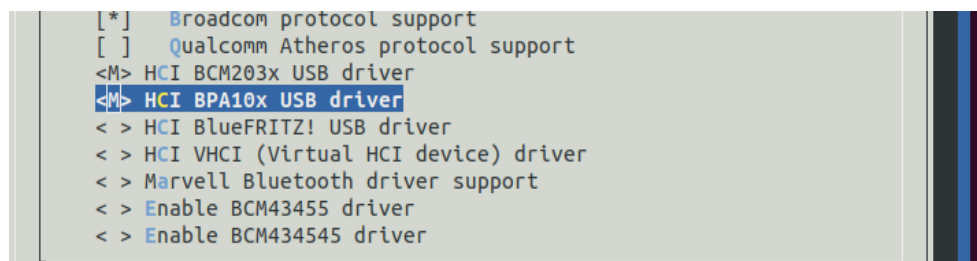
Tùy vào thiết bị giao tiếp mà ta cần chọn giao thức IrDA phù hợp (hình 5.7).



Hình 5.7 Các giao thức IrDA

### 5.2.5 Bluetooth

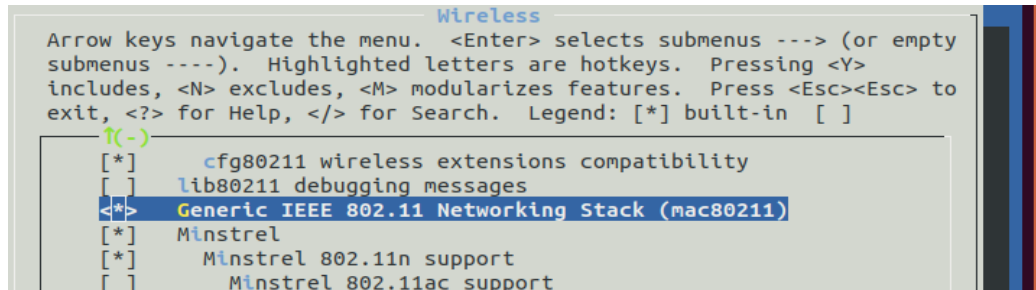
Bluetooth là công nghệ không dây được tạo ra để thay thế IrDA, hoạt động trong bán kính 10m, thường dùng ở điện thoại di động. Để hỗ trợ Bluetooth cần chọn Bluetooth subsystem support trong Networking support, cài đặt driver cho thiết bị Bluetooth trong Bluetooth device drivers. Có khá ít driver vì hầu hết các thiết bị đều tuân theo thông số kỹ thuật hoạt động của Bluetooth. Cần chọn 2 driver HCI BCM203x USB driver và HCI BPA10x USB driver để Bluetooth có thể hoạt động (hình 5.8).



Hình 5.8 Các driver cho Bluetooth

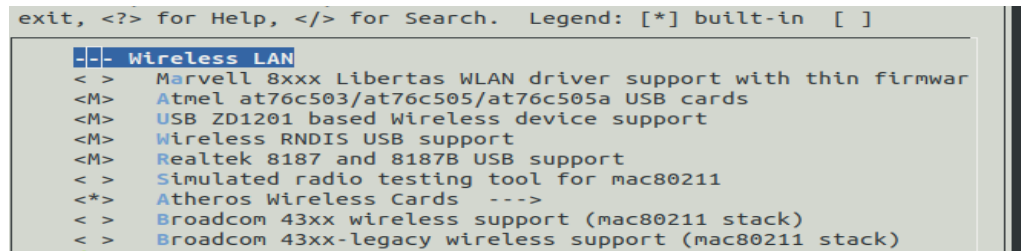
### 5.2.6 Mạng không dây

Linux hỗ trợ rất nhiều driver không dây. Để kích hoạt hỗ trợ không dây trong Linux, cần chọn cấu hình mạng IEEE 802.11 Networking Stack trong mục Wireless.



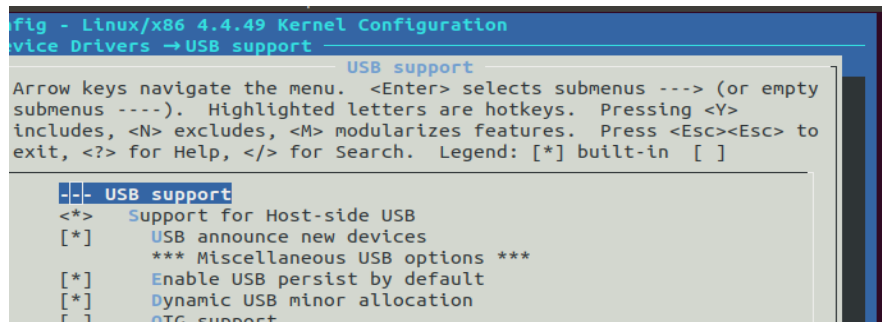
Hình 5.9 IEEE 802.11 Networking Stack

Để chỉnh sửa các driver không dây trong Network device support, chọn mục Wireless LAN (hình 5.10).



Hình 5.10 Các driver mạng không dây

Cấu hình driver USB Network Adapter có thể được chỉnh sửa trong USB Support trong Device Drivers (hình 5.11).



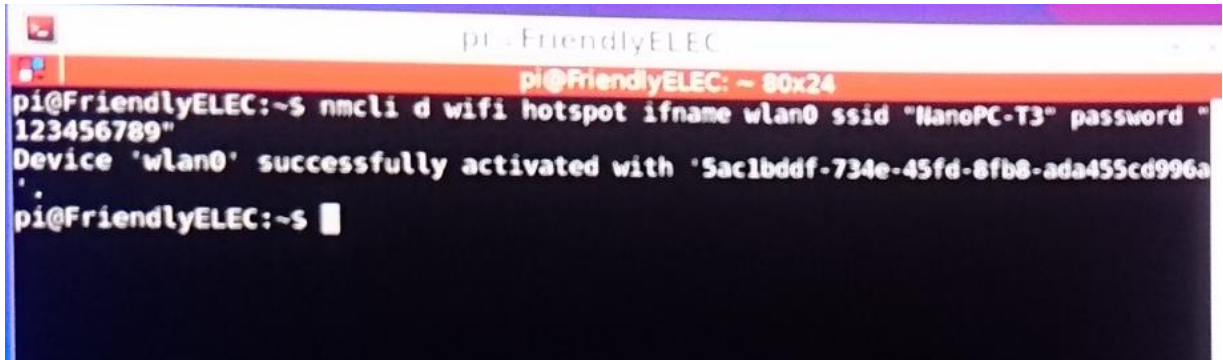
Hình 5.11 Driver USB Network Adapter

## Chương 6. Phát wifi trên kit NanoPC-T3

### 6.1 Trên Ubuntu

Trên hệ điều hành Ubuntu, mạng được quản lý bởi Network Manager Command Line Interface, vì vậy nhóm em sử dụng lệnh nmcli để phát wifi (hình 6.1)

`nmcli d wifi hotspot ifname wlan0 ssid "NanoPC-T3" password "123456789"`



```
pi@FriendlyELEC: ~$ nmcli d wifi hotspot ifname wlan0 ssid "NanoPC-T3" password "123456789"
Device 'wlan0' successfully activated with '5ac1bddf-734e-45fd-8fb8-ada455cd996a'
pi@FriendlyELEC:~$
```

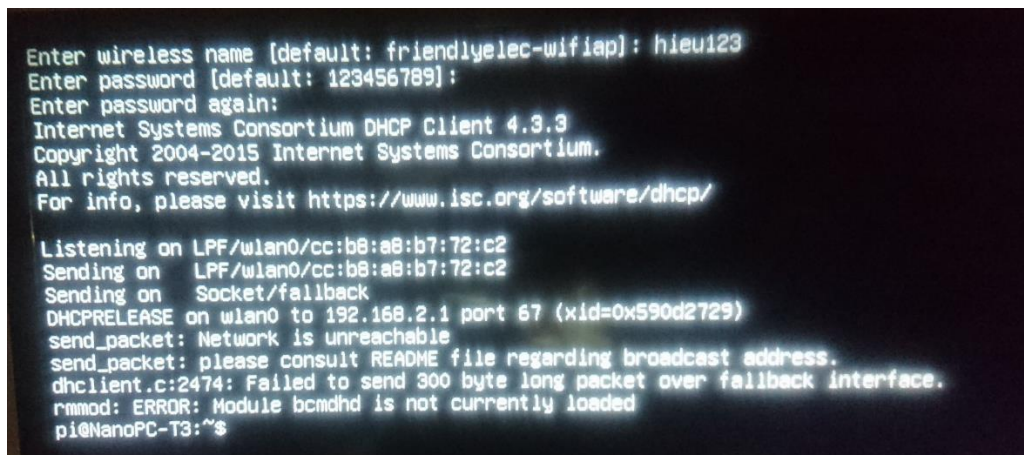
Hình 6.1 Phát wifi trên Ubuntu

### 6.2 Trên Debian/Friendly Core

Trên hệ điều hành Debian/FriendlyCore, chế độ wifi-AP được thiết lập bằng:

`sudo turn wifi-into-ap-mode yes`

Trên Friendly Core kernel 4.4 báo lỗi “Module bcmhdh is not currently loaded” (hình 6.2), vì module này đã bị thay đổi trên kernel 4.4, vì vậy nhóm em biên dịch lại kernel 3.4



```
Enter wireless name [default: friendlyelec-wifiap]: hieu123
Enter password [default: 123456789]:
Enter password again:
Internet Systems Consortium DHCP Client 4.3.3
Copyright 2004-2015 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan0/cc:b8:a8:b7:72:c2
Sending on LPF/wlan0/cc:b8:a8:b7:72:c2
Sending on Socket/fallback
DHCPRELEASE on wlan0 to 192.168.2.1 port 67 (xid=0x590d2729)
send_packet: Network is unreachable
send_packet: please consult README file regarding broadcast address.
dhclient.c:2474: Failed to send 300 byte long packet over fallback interface.
rmmmod: ERROR: Module bcmhdh is not currently loaded
pi@NanoPC-T3:~$
```

Hình 6.2 Lỗi khi thiết lập wifi-AP

- Tải kernel 3.4  
`git clone https://github.com/friendlyarm/linux-3.4.y.git`  
`cd linux-3.4.y`  
`git checkout nanopi2-lollipop-mr1`
- Biên dịch kernel  
`make nanopi3_linux_defconfig`  
`touch .scmversion`  
`make uImage`

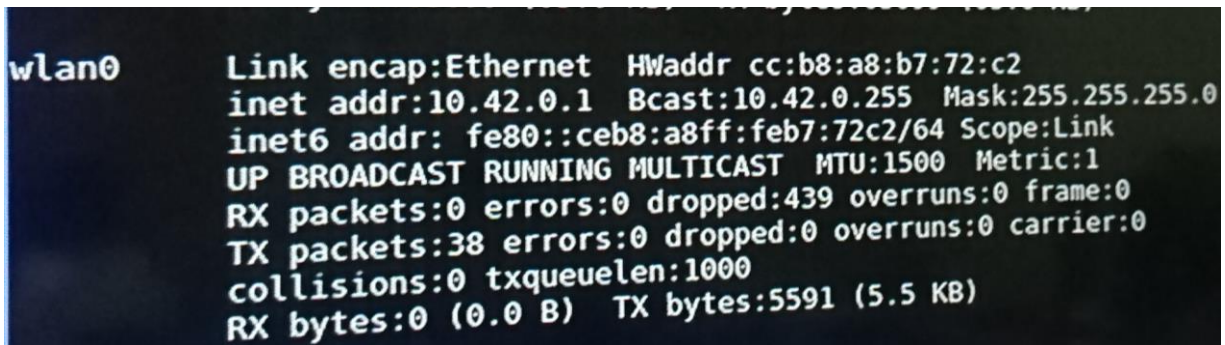
## Chương 7. Cấu hình mạng từ xa cho kit NanoPC-T3

### 7.1 Kết nối tới kit bằng giao thức SSH

SSH, hoặc được gọi là Secure Shell, là một giao thức điều khiển từ xa cho phép người dùng kiểm soát và chỉnh sửa server từ xa qua Internet. Dịch vụ được tạo ra nhằm thay thế cho trình Telnet vốn không có mã hóa và sử dụng kỹ thuật cryptographic để đảm bảo tất cả giao tiếp gửi tới và gửi từ server từ xa diễn ra trong tình trạng mã hóa. Nó cung cấp thuật toán để chứng thực người dùng từ xa, chuyển input từ client tới host, và relay kết quả trả về tới khách hàng.

Để kết nối tới kit bằng giao thức SSH qua wifi, trước tiên nhóm em xác định địa chỉ IP của kit (hình 7.1). Địa chỉ IP được xác định là: 10.42.0.1.

ifconfig

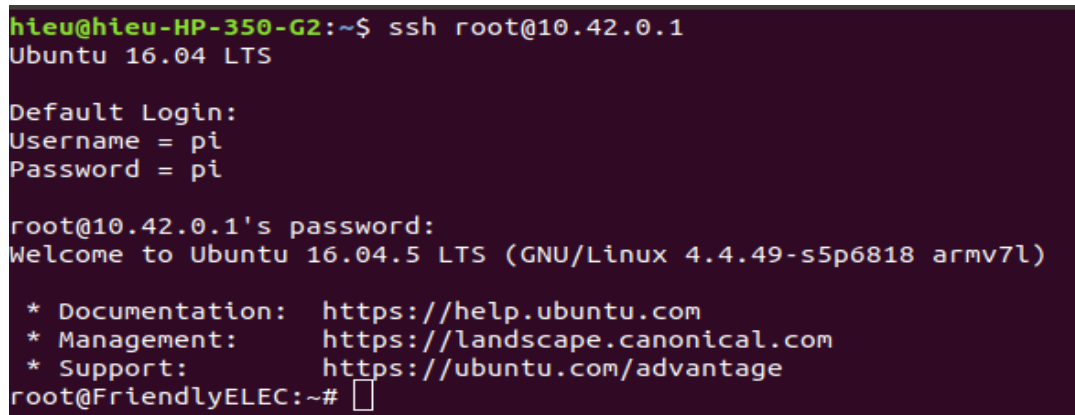


```
wlan0    Link encap:Ethernet  HWaddr cc:b8:a8:b7:72:c2
          inet addr:10.42.0.1  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::ceb8:a8ff:feb7:72c2/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:439 overruns:0 frame:0
          TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:5591 (5.5 KB)
```

Hình 7.1 Xác định địa chỉ IP của kit

Sau khi xác định được địa chỉ IP, nhóm em truy cập vào kit với chế độ root (hình 7.2). Sau khi ssh thành công thì có thể chỉnh sửa các file trong kit từ xa qua Terminal trên máy tính.

ssh root@10.42.0.1



```
hieu@hieu-HP-350-G2:~$ ssh root@10.42.0.1
Ubuntu 16.04 LTS

Default Login:
Username = pi
Password = pi

root@10.42.0.1's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.49-s5p6818 armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
root@FriendlyELEC:~#
```

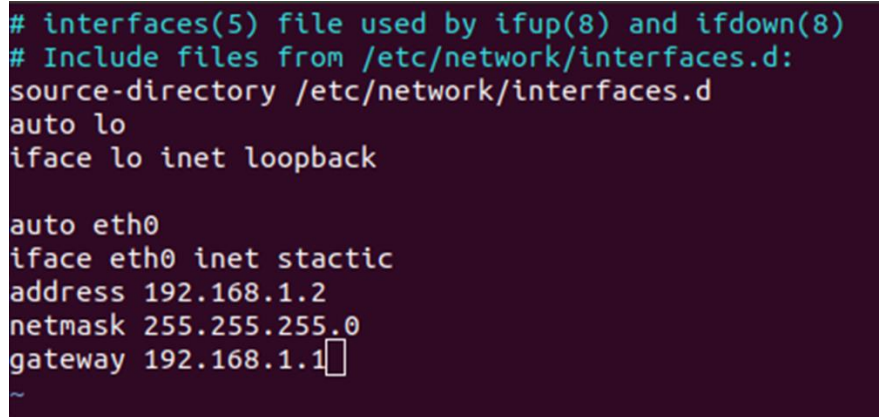
Hình 7.2 Truy cập tới kit bằng SSH

## 7.2 Cấu hình mạng cho kit

### 7.2.1 Trình soạn thảo vi

Nhóm em sử dụng trình soạn thảo vi để chỉnh sửa các file cấu hình mạng cho kit. Ví dụ để chỉnh sửa file cấu hình mạng /etc/network/interfaces (hình 7.3):

`vi /etc/network/interfaces`



```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1
```

Hình 7.3 Chỉnh sửa file bằng trình soạn thảo vi

### 7.2.2 Cấu hình mạng Domain Resolution

Các file sau cấu hình hệ thống để tên máy chủ có thể truy cập khi dùng ssh:

- File /etc/resolv.conf (hình 7.4): file cấu hình tên host để xác định máy chủ chịu trách nhiệm name resolution (DNS). Nội dung file này có dạng:

nameserver XXX.XXX.XXX.XXX – Địa chỉ IP của tên server thứ nhất

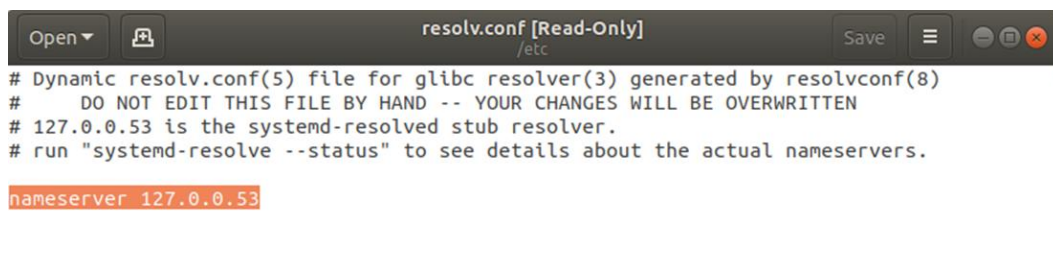
nameserver XXX.XXX.XXX.XXX – Địa chỉ IP của tên server thứ hai

Một ví dụ cho nội dung file này là:

nameserver 8.8.8.8

nameserver 8.8.8.4

Trong đó 8.8.8.8 là DNS chính của google, còn 8.8.8.4 là DNS phụ của google



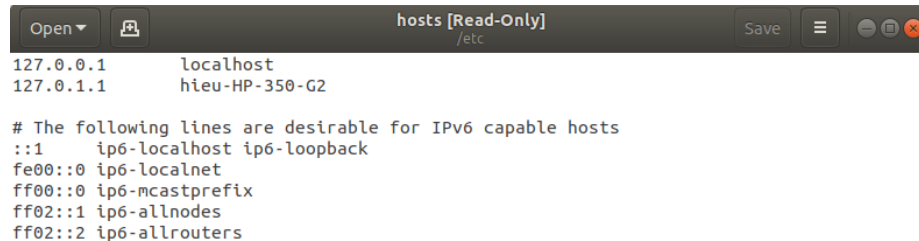
```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "systemd-resolve --status" to see details about the actual nameservers.

nameserver 127.0.0.53
```

Hình 7.4 File /etc/resolv.conf



- File /etc/hosts (hình 7.5): Quyết định cục bộ tên nút thành địa chỉ IP rõ ràng. Nội dung file này có dạng:  
127.0.0.1      tên\_node.tên\_miền.com localhost.localdomain localhost  
XXX.XXX.XXX.XXX      tên\_node



```

127.0.0.1      localhost
127.0.1.1      hieu-HP-350-G2

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

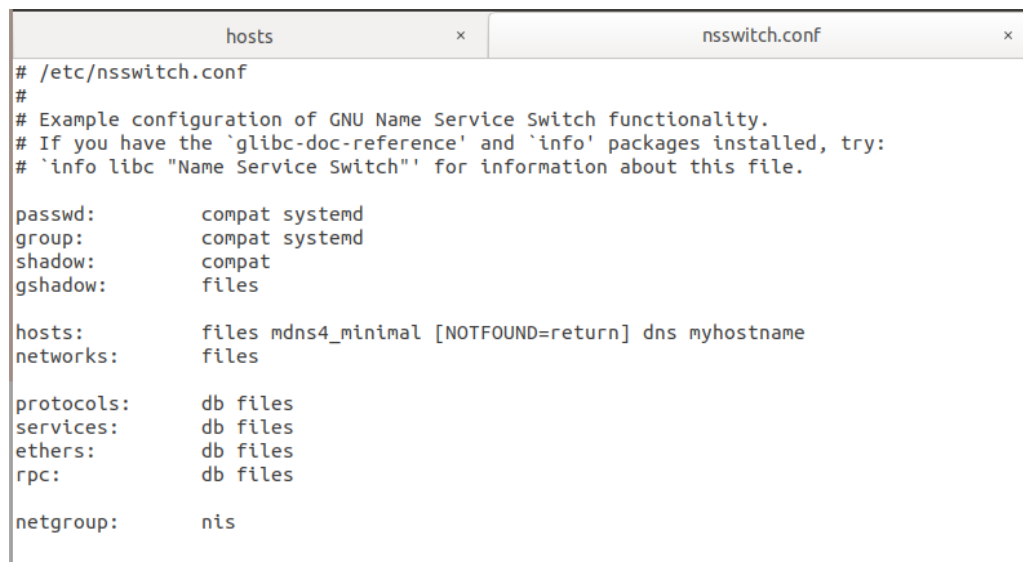
```

**Hình 7.5 File /etc/hosts**

Để thêm máy chủ (ví dụ abc.com), ta ghi dòng lệnh sau vào file

XXX.XXX.XXX.XXX abc.com superserver

- File /etc/nsswitch.conf (hình 7.6): file name service switch cấu hình chuyển đổi tên dịch vụ và cơ sở dữ liệu hệ thống, xác định mức ưu tiên của name resolvers



```

# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:          compat systemd
group:           compat systemd
shadow:          compat
gshadow:         files

hosts:           files mdns4_minimal [NOTFOUND=return] dns myhostname
networks:        files

protocols:       db files
services:        db files
ethers:          db files
rpc:             db files

netgroup:        nis

```

**Hình 7.6 File /etc/nsswitch.conf**

### 7.2.3 Cấu hình mạng trên Lubuntu/Debian

Trên các hệ điều hành Lubuntu/Debian, các file cấu hình IP nằm trong /etc/network/interfaces (hình 7.7). Nội dung file này có dạng:

```
# Card mạng loopback

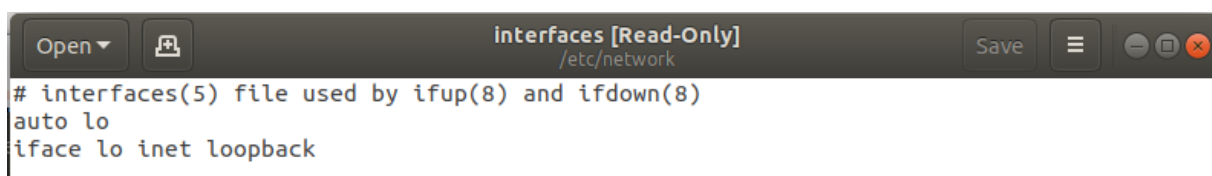
auto lo

iface lo inet loopback


auto wlan0 # Card mạng chính

iface wlan0 inet static

address xxx.xxx.xxx.xxx # Địa chỉ IP
network xxx.xxx.xxx.xxx # Địa chỉ mạng
netmask xxx.xxx.xxx.xxx # subnet mask
broadcast xxx.xxx.xxx.xxx # Địa chỉ broadcast
gateway xxx.xxx.xxx.xxx # Gateway mặc định
dns-nameservers xxx.xxx.xxx.xxx # Tên server
```



**Hình 7.7 File /etc/network/interfaces**

Ví dụ sau thiết lập IP tĩnh

```
auto lo

iface lo inet loopback


auto eth0

iface eth0 inet static

address 208.88.34.106
```



```
netmask 255.255.255.248
```

```
broadcast 208.88.34.111
```

```
network 208.88.34.104
```

```
Gateway 208.88.34.110
```

Ví dụ sau thiết lập IP động bằng giao thức DHCP (Dynamic Host Configuration Protocol)

```
auto lo # Giao diện loopback
```

```
iface lo inet loopback
```

```
auto eth0 # Giao diện Ethernet thứ nhất
```

```
iface eth0 inet dhcp
```

```
auto eth1 # Giao diện Ethernet thứ hai
```

```
iface eth1 inet dhcp
```

```
auto eth2 # Giao diện Ethernet thứ ba
```

```
iface eth2 inet dhcp
```

```
auto ath0 # Giao diện mạng không dây ath thứ nhất
```

```
iface ath0 inet dhcp
```

```
auto wlan0 # Giao diện mạng không dây wlan thứ nhất
```

```
iface wlan0 inet dhcp
```

## **KẾT LUẬN**

Trong đề tài này, nhóm đã cài hệ điều hành Lubuntu cho kit NanoPC-T3, biên dịch kernel và U-Boot, cấu hình mạng trong kernel và cấu hình mạng từ xa cho kit. Dù còn gặp nhiều khó khăn và lỗi, nhóm em đã cùng nhau nghiên cứu và tìm cách khắc phục, tích lũy thêm kinh nghiệm về hệ điều hành.

## TÀI LIỆU THAM KHẢO

- [1] <http://wiki.friendlyarm.com/wiki/index.php/NanoPC-T3>
- [2] [https://github.com/friendlyarm/sd-fuse\\_s5p6818](https://github.com/friendlyarm/sd-fuse_s5p6818)
- [3] [https://en.wikipedia.org/wiki/Das\\_U-Boot](https://en.wikipedia.org/wiki/Das_U-Boot)
- [4] [https://www.linuxtopia.org/online\\_books/linux\\_kernel/kernel\\_configuration/](https://www.linuxtopia.org/online_books/linux_kernel/kernel_configuration/)
- [5] <http://www.yolinux.com/TUTORIALS/LinuxTutorialNetworking.html>