

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN - ĐIỆN TỬ



ĐỒ ÁN MÔN HỌC 1

EMBEDDED LINUX SYSTEM

SVTH:

TRẦN ANH VŨ 41304911

LÊ NHẬT THANH 41303594

GVHD:

BÙI QUỐC BẢO

TP.HCM, 2016

Lời cảm ơn

Chúng em xin chân thành cảm ơn thầy Bùi Quốc Bảo và bộ môn Điện tử Viễn thông - Khoa Điện – Điện tử Trường – DHBK – Đại học Quốc gia TP HCM đã tạo điều kiện cho chúng em có cơ hội được làm việc với thầy Bảo thông qua đồ án môn học 1. Đây là cơ hội để cho chúng em có thêm nhiều kinh nghiệm thực tế, và là một trong những bước khởi đầu vô cùng quan trọng trong sự nghiệp sau này của chúng em. Trước khi bước vào đồ án môn học 1, những kiến thức mà chúng em có được chỉ đơn thuần qua sách vở và những lần làm bài tập lớn. Nhưng qua đồ án môn học này, chúng em đã tích lũy được rất nhiều kiến thức cần thiết và quý giá cho bước khởi đầu của chúng em thêm vững chắc. Chúng em hiểu được rõ hơn mình cần phải học những gì, làm những gì để có thể phù hợp với các nhà tuyển dụng sau này, và hiểu thêm về công việc của một kỹ sư điện tử viễn thông.

Thầy đã hướng dẫn cho chúng em rất tận tình và nghiêm túc. Chúng em được làm việc nhóm thường xuyên, rèn kỹ năng teamwork cũng như kỹ năng đọc tài liệu bằng tiếng Anh. Tuy đồ án thầy giao cho chúng em rất khác với các đồ án 1 của các bạn khác, nhưng chúng em có thể hiểu được thầy muốn gì. Thầy mong muốn chúng em có một cái nhìn tổng quan hơn về công việc mình đang theo đuổi (embedded system), cũng như hiểu rõ các giai đoạn để tạo nên một sản phẩm.

Chúng em rất mong muốn được thầy Bảo hướng dẫn đồ án môn học 2 cũng như luận văn tốt nghiệp để tại em có thể tiếp thu kiến thức một cách hệ thống nhất, hiệu quả nhất và đúng chuyên môn nhất. Chúng em xin chân thành cảm ơn.

Tóm tắt đồ án

Embedded linux system là một đồ án không tạo ra một sản phẩm cụ thể, mà đồ án chuyên sâu về việc tìm hiểu tổng quan và các bước để thiết kế và xây dựng một hệ thống nhúng. Nhằm giúp cho sinh viên có cái nhìn tổng quan nhất về công việc sau này của một kỹ sư thiết kế hệ thống nhúng. Những nội dung trong tài liệu này chỉ dừng ở mức tìm hiểu và thực hiện một cách cơ bản, không thực hiện chuyên sâu. Đồ án bao gồm những chương sau:

Chương 1: Giới thiệu tổng quan về hệ thống nhúng và cách thực hiện một hệ thống nhúng.

Chương 2: Tìm hiểu và thiết kế một mạch đơn giản bằng cách thực hiện đầy đủ các bước cần thiết để thiết kế 1 hệ thống. Bao gồm các bước như sau: tìm kiếm linh kiện trên website, thiết kế schematic, thiết kế PCB.

Chương 3: Xây dựng một hệ thống embedded linux. Bao gồm các bước trong chương 2 cộng thêm các bước sau: build một hệ điều hành linux vào board mạch SoC, xây dựng webserver để điều khiển hoạt động của mạch bằng internet (Internet of Things), sau này là viết driver điều khiển các ngoại vi của mạch và phát triển Applications.

Chương 4: Tổng kết và phương hướng phát triển đồ án.

Thành phố Hồ Chí Minh, ngày 27 tháng 12 năm 2016

Lê Nhật Thanh

Trần Anh Vũ

1. Tổng quan	1
1.1. Giới thiệu về embedded linux system	1
1.2. Mục tiêu, nhiệm vụ, phạm vi đề tài	1
1.3. Tổ chức đồ án	1
2. Tìm hiểu và thiết kế	2
2.1. Cách tìm kiếm linh kiện trên website	2
2.2. Đọc datasheet của linh kiện	3
2.3. Thiết kế mạch đơn giản	3
2.3.1. Giới thiệu mạch thiết kế	3
2.3.2. Sử dụng công cụ Orcad 16.6 vẽ schematic	5
2.3.3. Sử dụng công cụ Allegro vẽ PCB	5
2.4. Phát triển mạch GM8136s	7
2.4.1. Giới thiệu mạch GM8136s	7
2.4.2. Thêm module phát triển mạch	9
3. Xây dựng một hệ thống linux embedded	10
3.1. Giới thiệu mạch Doobell_Rk3188	10
3.2. Build kernel hệ điều hành Ubuntu	11
3.2.1. Giới thiệu kernel	11
3.2.2. Các bước tạo kernel	11
3.2.3. Tạo các file hệ thống rootfs	14
3.3. Boot kernel lên SDcard và NAND flash	16
3.3.1. Boot lên SD Card	16
3.3.3. Khởi động Linux trên board RK3188	18
3.4. Điều khiển ngoại vi đơn giản (LED RGB)	18
3.5. Xây dựng web server và điều khiển ngoại vi bằng web server	20
3.5.1. Viết web server đơn giản bằng html và css	20
3.5.2. Cài đặt web server lên board	21
3.5.3. Điều khiển LED thông qua web server	23
3.5.4. Giới thiệu về openhab	23
3.6. Kết luận	25
Chương 4: Tổng kết và phương hướng phát triển đề tài	26
4.1. Đánh giá kết quả đạt được	26
4.2. Hướng phát triển đề tài	26

Chương 1: Tổng quan

1.1. Giới thiệu về embedded linux system

Embedded system (hệ thống nhúng) là một hệ thống được tích hợp cả về phần cứng và phần mềm phục vụ cho các bài toán chuyên dụng trong nhiều lĩnh vực công nghiệp, tự động hóa điều khiển, quan trắc và truyền thông. Hệ thống nhúng thường không phải là một khối riêng biệt mà là một hệ thống phức tạp nằm trong thiết bị mà nó điều khiển.

1.2. Mục tiêu, nhiệm vụ, phạm vi đề tài

Mục tiêu của đề tài là giúp cho sinh viên tiếp cận một cách tổng quan và nhanh nhất về quy trình để thực hiện một hệ thống nhúng.

1.3. Tổ chức đồ án

Đồ án được thực hiện bởi một nhóm hai người (vì có khá nhiều công việc phải làm), và phân chia nhiệm vụ đồng đều trong nhóm và do nhóm quyết định.

Chương 2: Tìm hiểu và thiết kế

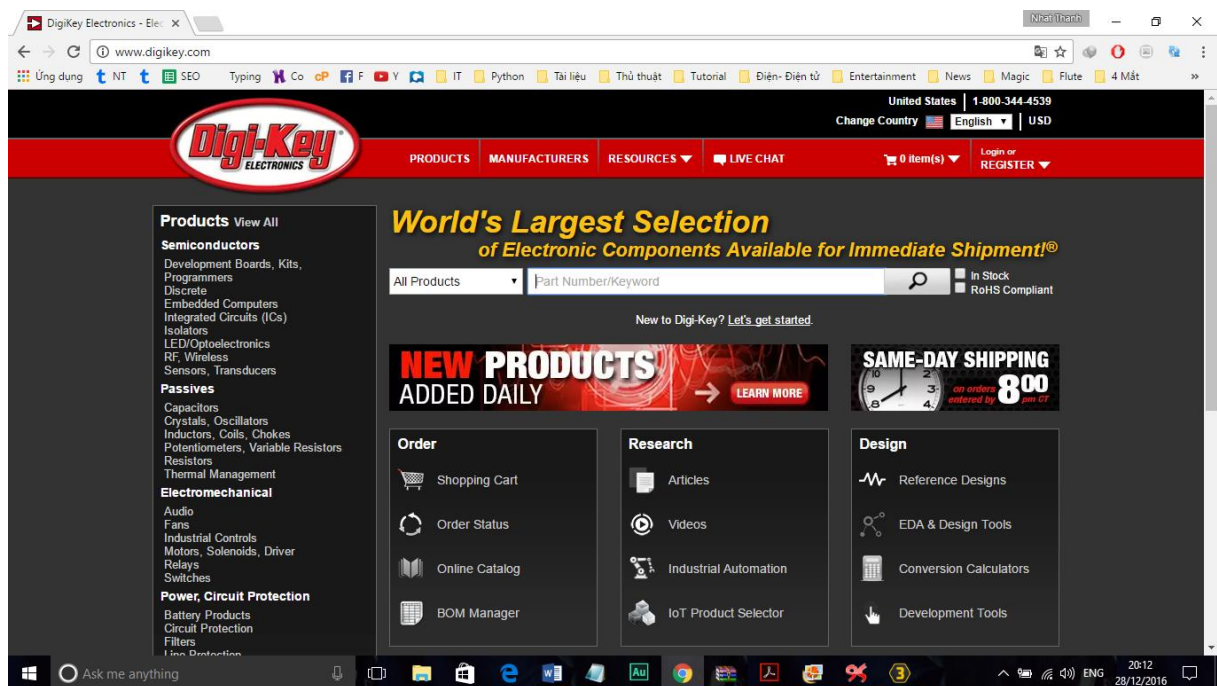
2.1. Cách tìm kiếm linh kiện trên website

Tìm kiếm linh kiện trên website là một việc rất quan trọng trong thiết kế một hệ thống nhúng. Tìm kiếm linh kiện phục vụ cho việc vẽ PCB và gia công mạch sau này. Ngoài ra còn biết được giá cả, nơi bán, hãng sản xuất như thế nào để có thể đặt mua linh kiện đúng với những gì mong muốn.

Sau đây là cách tìm kiếm linh kiện trên website: <http://www.digikey.com/>

Đây là website rất nổi tiếng trên thế giới về linh kiện điện tử. Người mua hàng có thể thoải mái lựa chọn cho mình những thứ phù hợp nhất.

Lưu ý: Nếu muốn mua linh kiện thì chúng ta cần phải thực hiện việc đăng kí tài khoản và đăng nhập vào hệ thống cũng như phải có các tài khoản online có thể thanh toán quốc tế như visa hay paypal.



Trên đây là trang chủ của digikey, khi cần tìm một linh kiện nào đó, hãy nhập Part Number hoặc từ khóa và ô tìm kiếm của một linh cần cần tìm, hãy tick vào nút In Stock để xem những linh kiện còn trong kho để có thể mua nhanh chóng. Ngoài ra chúng ta còn có thể tìm kiếm dựa vào các danh mục có sẵn trên web, và có thể lọc danh sách các linh kiện dựa vào các thông tin cần mà chúng ta cần trong mạch như tần số hoạt động, băng thông Website sẽ liệt kê cho chúng ta danh sách các linh kiện tìm được, cũng

như các thông tin khác như giá cả, hãng sản xuất, ... và có cả datasheet nữa. Những thông tin này rất cần thiết cho khách hàng. Sau khi tìm đúng linh kiện cần thiết thì click vào linh kiện đó. Tiếp đó website sẽ hiện ra các thông tin thanh toán, chúng ta chỉ cần làm theo hướng dẫn.

2.2. Đọc datasheet của linh kiện

Một kỹ năng mà bắt buộc một kỹ sư điện tử phải có chính là đọc datasheet của một linh kiện hay của một mạch nào đó. Khi linh kiện (hay mạch điện) được sản xuất ra thì nhà sản xuất sẽ đính kèm một file (thường là pdf) cho biết thông tin chi tiết về linh kiện (hay mạch điện) về cấu tạo, chức năng, hoạt động, Trên mạng hiện nay có rất nhiều website cung cấp cho chúng ta datasheet như: <http://www.digikey.com/> hay <http://www.alldatasheet.com/>.

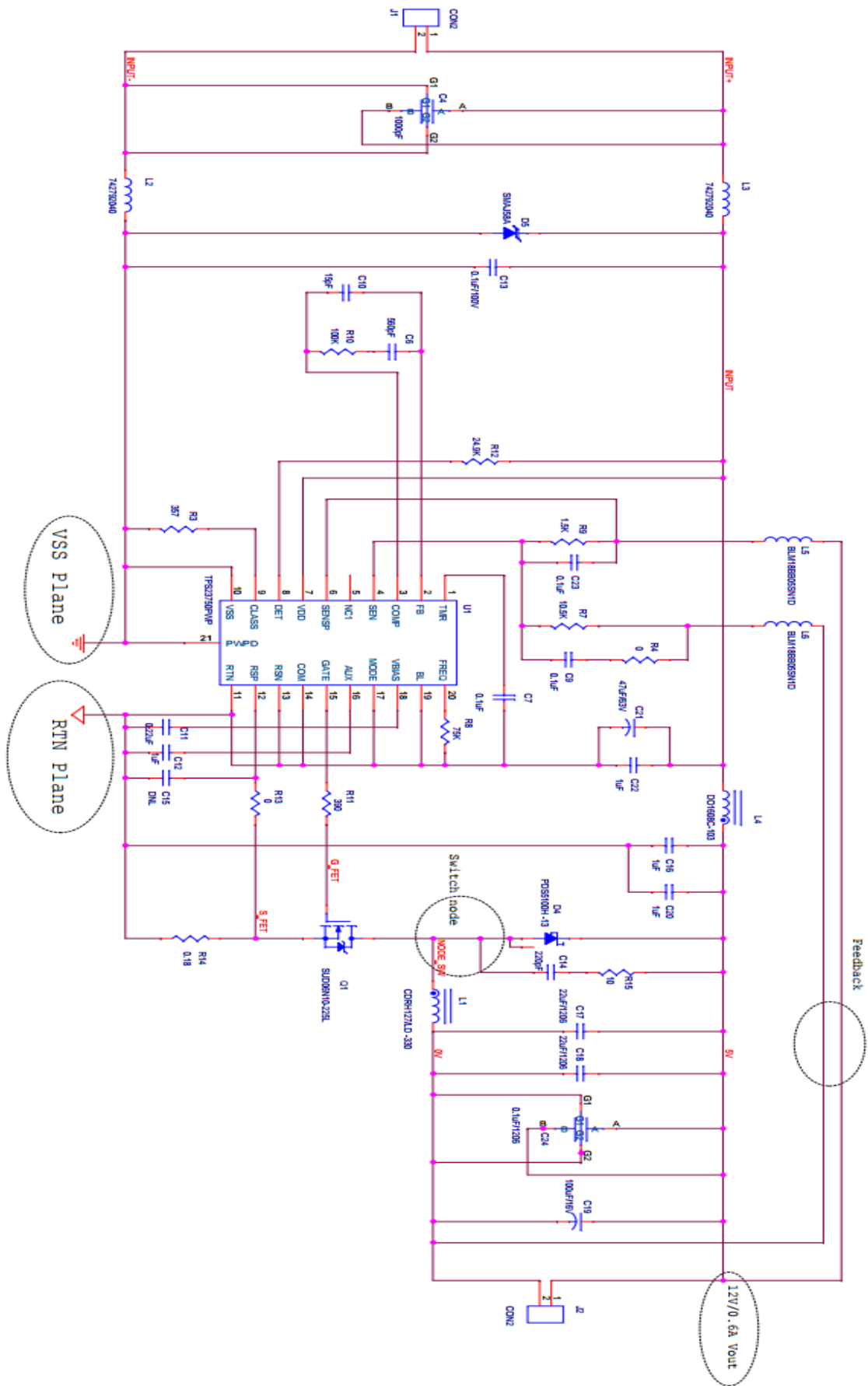
Việc đọc datasheet cung cấp cho chúng ta những thông tin kỹ thuật cần thiết của linh kiện để cho chúng ta có thể thiết kế được mạch mong muốn. Và một trong những thông tin vô cùng quan trọng nữa đó là kích thước thực tế của linh kiện hay kích thước vẽ layout của linh kiện, để chúng ta có thể thực hiện vẽ PCB (một trong những bước chính để thiết kế một mạch).

Chúng ta nên tập đọc datasheet, vì sau này ra ngoài doanh nghiệp, đọc datasheet là việc chúng ta phải làm thường xuyên. Ngoài ra còn luyện cho chúng ta cách đọc tài liệu bằng tiếng Anh, một kỹ năng mà bất kỳ doanh nghiệp nào cũng đòi hỏi.

2.3. Thiết kế mạch đơn giản

2.3.1. Giới thiệu mạch thiết kế

Dưới đây là schematic của mạch cần thiết kế PCB. Đây là một module power over Ethernet. Chúng ta sẽ thiết kế mạch này bằng công cụ Orcad và PCB Allegro 16.6 của tập đoàn Cadence. Phần thiết kế ở mục sau.



2.3.2. Sử dụng công cụ Orcad 16.6 vẽ schematic

Hiện nay có rất nhiều công cụ thiết kế mạch mà nổi tiếng nhất là Orcad, Allegro của Cadence và công cụ khác là Altium. Tài liệu này sẽ dùng công cụ Orcad 16.6 để thiết kế mạch.

Lưu ý: Tài liệu này chỉ nêu ra các bước vẽ schematic của mạch cần thiết kể chứ không hướng dẫn chi tiết sử dụng công cụ.

Đầu tiên chúng ta sẽ vẽ mạch schematic như hình vẽ trên và điền đầy đủ các giá trị linh kiện và các chú thích, đặc biệt chúng ta phải đặt Pin Number và PCB footprint cho linh kiện để có thể vẽ PCB (nếu thiết kế mạch lớn thì có thể vẽ trên nhiều trang, và chúng ta nên tạo một file BOM.xlsx để liệt kê tất cả các thông tin linh kiện trong mạch như tên linh kiện, Part, Part Number, Manufacturer, PCB Footprint, link mua linh kiện, giá cả, ...).

Khi đã vẽ xong thì chúng ta tiến hành kiểm tra lỗi của mạch bằng công cụ Design rules check được tích hợp sẵn trong Orcad 16.6.

Bước tiếp theo là tạo Netlist. Bước này vô cùng quan trọng để chuyển sang bước tiếp theo là thiết kế PCB.

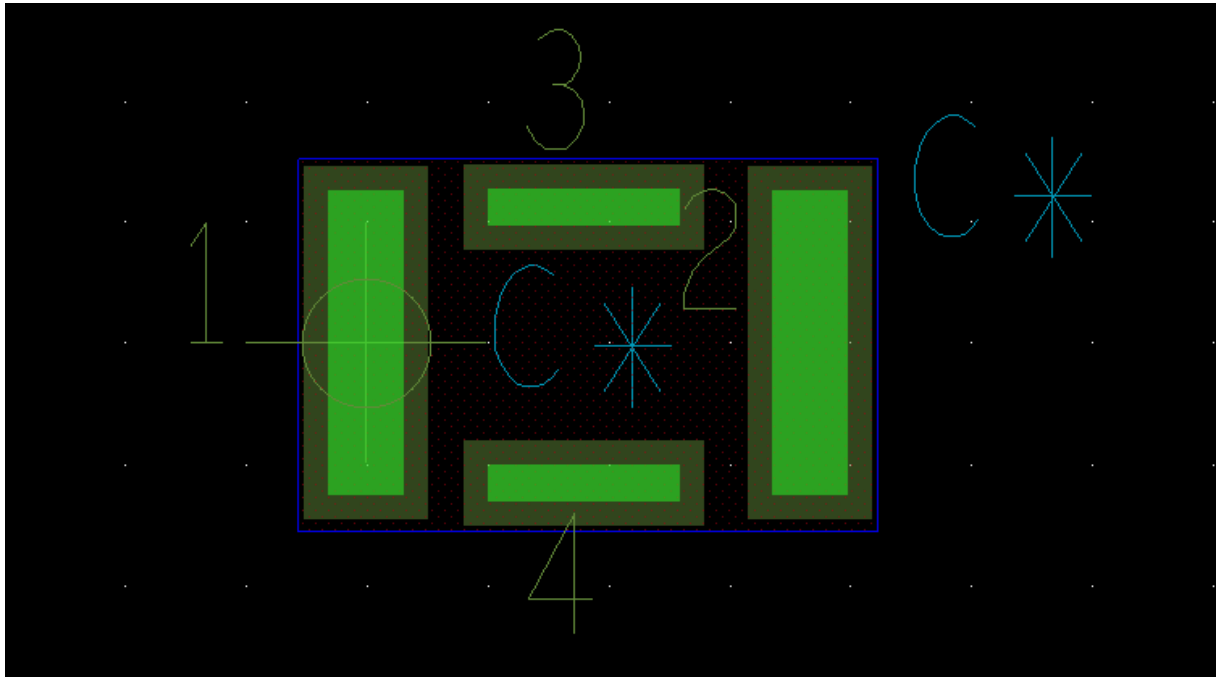
2.3.3. Sử dụng công cụ Allegro vẽ PCB

Allegro là công cụ vô cùng mạnh mẽ để thiết kế PCB, được sử dụng rất rộng rãi trong các doanh nghiệp. Sinh viên chúng ta mà biết sử dụng công cụ này là một lợi thế rất lớn để sau này có thể phỏng vấn xin việc. Và sau đây là các bước để thiết kế PCB.

Bước 1: Đọc yêu cầu thiết kế mạch mà khách hàng yêu cầu. Ví dụ yêu cầu của mạch này như sau: Kích thước module là 20x20 mm, pin header ở 2 phía đối diện, và linh kiện thông tin linh kiện dùng được thể hiện trong file BOM.xlsx do thầy cung cấp.

Bước 2: Đọc datasheet của những linh kiện trong mạch để có thiết biết kích thước layout và sau đó tiến hành vẽ footprint (trước khi tạo footprint thì phải tạo Pad cho linh kiện) cho linh kiện (linh kiện dán hay khoan lỗ). Khi đã vẽ footprint xong chúng ta phải đặt tên footprint trùng với tên PCB footprint mà chúng ta đã đặt bên schematic, còn một điều nữa là lưu ý Pin Number của các linh kiện có cực hoặc linh kiện có 3 chân trở lên.

Hình ảnh footprint smc1206_X2Y là một linh kiện dán.



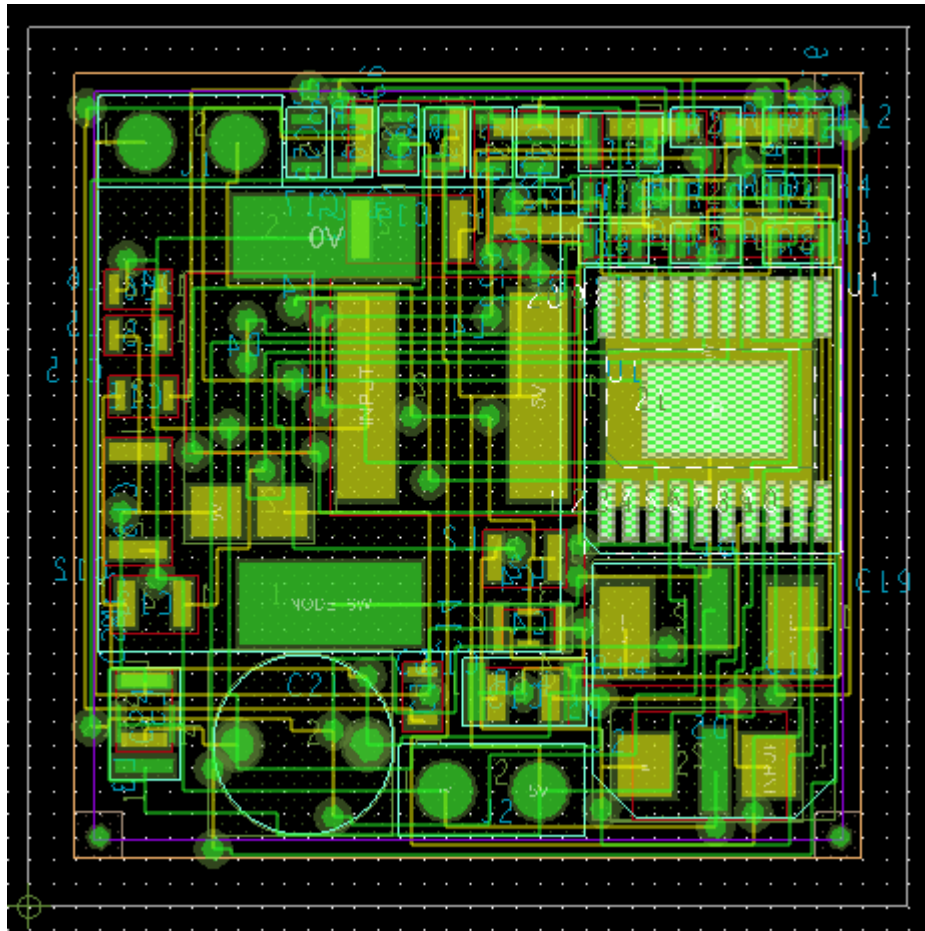
Bước 3: Tạo Board mạch nghĩa là tạo cách đường biên giới hạn cho linh kiện cũng như việc đi dây và viết các text để thể hiện thông tin project,

Bước 4: Sử dụng file netlist và các footprint đã tạo để đặt và sắp xếp linh kiện vào board đúng với kích thước yêu cầu (20x20 mm).

Bước 4: Đi dây. Đây là bước rất quan trọng trong thiết kế PCB, chúng ta phải tự đi dây bằng tay, nghĩa là không nên auto rout. Khi thiết kế mạch tần số cao, tốc độ cao thì chúng ta phải đặt biệt chú trọng đến các luật đi dây như chiều rộng dây dẫn, trở kháng, khoảng cách giữa các chân linh kiện, và những quy định về via, Chúng ta có thể thiết kế mạch nhiều lớp (trong mạch này là thiết kế 2 lớp).

Bước 5: Phủ đồng và xuất file để đi gia công mạch. Nếu là mạch nhiều lớp thì chúng ta chỉ có thể đặt gia công chứ không thể tự gia công bằng tay được.

Sau đây là mạch đã thiết kế của nhóm:



Kết luận: thiết kế PCB không phải là một chuyện dễ làm, chúng ta cần phải học tập và thực hiện rất nhiều để có thể hoàn chỉnh được một mạch PCB đúng yêu cầu.

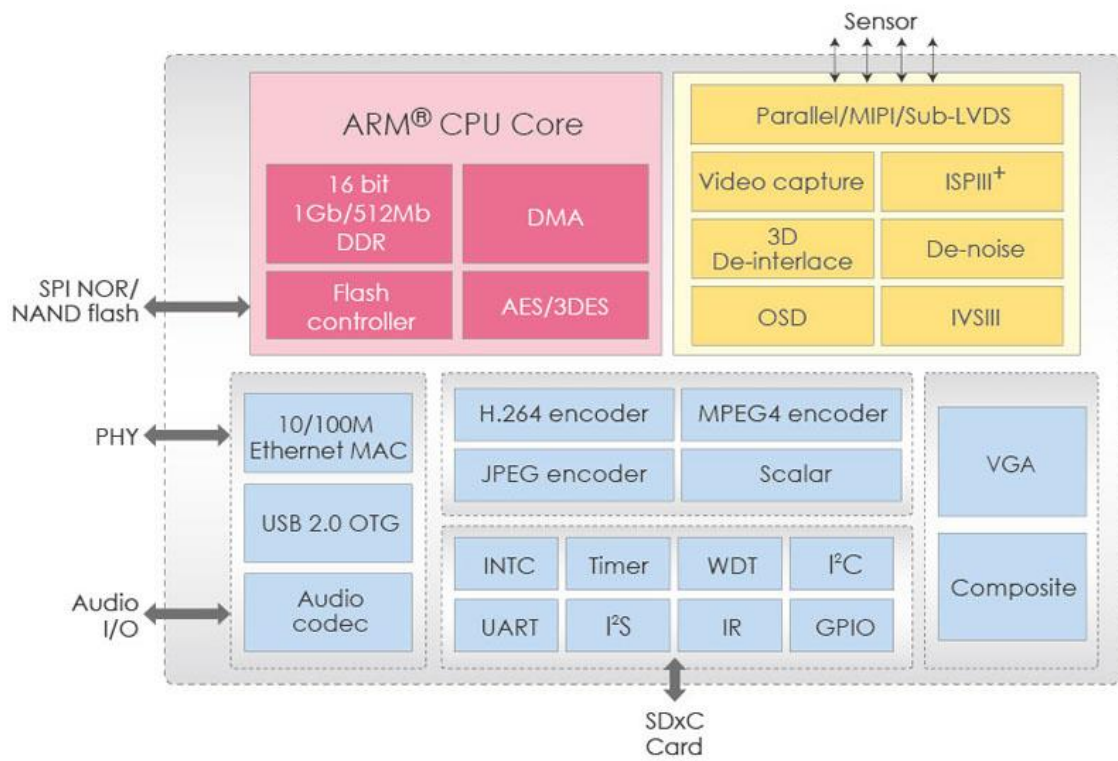
2.4. Phát triển mạch GM8136s

2.4.1. Giới thiệu mạch GM8136s

Grain Media's IP Camera SoC GM8136S embedded là một board mạch của công ty Grain Media sử dụng CPU ARM dùng để điều khiển rất nhiều ngoại vi để tạo nên một hệ thống lớn như giám sát mục tiêu qua IP Camera, car DVR, hay một hệ thống nhà thông minh.

Các công cụ mã hóa video của mạch cung cấp bộ mã hóa H.264 HP/MP/BL. Cung cấp video chất lượng cao với yêu cầu băng thông thấp. Kết nối ngoại vi rất đa dạng bao gồm quay video, hiển thị điều khiển, điều khiển mã hóa, serial I/F controllers, USB 2.0 OTG, 10/100M Ethernet controller.

Sơ đồ khối của mạch:



Các thông số của mạch như sau:

Embedded Processor	<ul style="list-style-type: none"> ARM® CPU core - 32-bit RISC, up to 600MHz
Memory Interface	<ul style="list-style-type: none"> GM8136S: SiP 1Gb DDR GM8135S: SiP 512Mb DDR Support SPI NOR/NAND flash boot
H.264 Codec	<ul style="list-style-type: none"> Support the high profile encoding <ul style="list-style-type: none"> GM8136S: 720P@60fps GM8135S: 720P@45fps
Video Input	<ul style="list-style-type: none"> MIPI, Parallel, Sub-LVDS YUV x 2 for 2-channel 720P
Video Capture	<ul style="list-style-type: none"> BT.656 input (up to 148.5 MHz) BT.1120 input (up to 148.5 MHz) Support 54MHz/108MHz and 72MHz/144MHz byte/frame interleave modes
Image Signal Processing	<ul style="list-style-type: none"> Temporal and spatial 3D-Denoise filtering Support Digital-WDR Support WDR sensor Support tone mapping Low lux sensitivity to ISO6400 Embedded IVS engine
Display Interface	<ul style="list-style-type: none"> Built-in BT.1120 digital output Support composite/parallel up to FHD output
Peripherals Interface	<ul style="list-style-type: none"> 10/100M Ethernet MAC USB 2.0 OTG USB 1.1 Device SDIO x 2 I²S I²C GPIO
Operating Voltage	<ul style="list-style-type: none"> Core: 1.1 V DDR: 1.5/1.8 V I/O: 3.3 V
Package	<ul style="list-style-type: none"> 128-pin EPADLQFP

2.4.2. Thêm module phát triển mạch

Những module và những phần đã bỏ trong mạch cũ: bỏ toàn bộ Video output, Audio in/output, bỏ USB 1.1.

Các module được thêm vào để phát triển mạch phương thức kết nối các module với mạch:

- TFT LCD: kết nối song song 8 bit (sử dụng 8 bit cao DB08 – DB15).
- RFID: giao tiếp SPI.
- FingerPrint: giao tiếp UART.
- Keypad: giao tiếp song song với MCU.

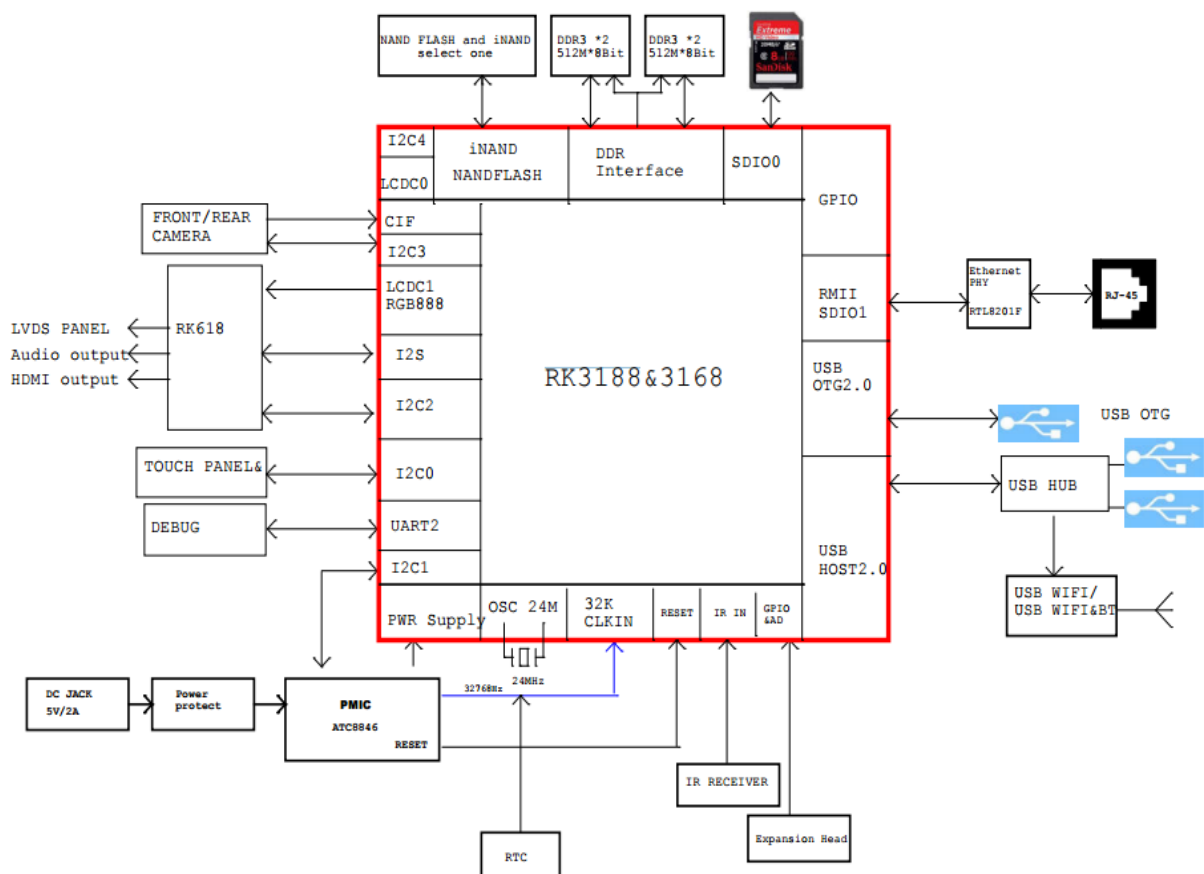
Sử dụng những dây đã bỏ 3 module trên để làm dây kết nối cho những module thêm vào. Riêng FingerPrint thì cắm thẳng vào UART2 trên board, không sử dụng thêm dây từ MCU.

Chương 3: Xây dựng một hệ thống linux embedded

3.1. Giới thiệu mạch Doobell_Rk3188

Rockchip RK3188 sở hữu CPU tốc độ cao, tần số hoạt động đạt đến 1.6GHz. CPU sở hữu 4 nhân, có thể hoạt động đa tác vụ, được xây dựng trên kiến trúc Harvard, có luồng dữ liệu cache riêng biệt, giúp cải thiện hiệu suất hoạt động tối ưu nhất. Bộ vi xử lý này được thiết kế dựa trên các chiến lược thiết kế Reduced Instruction Set Computing, cho phép cấu trúc thực hiện nhanh hơn nhiều lần. Được ứng dụng nhiều trong các ứng dụng nhúng, server, android box, TV box...

Sơ đồ khối:



Chip vi xử lý trung tâm RK3188:

- Quad-core ARM Cortex-A9 clocked up to 1.6 GHz frequency
- 28 nm HKMG process
- 512 KB L2 cache
- Mali-400 MP4 GPU, up to 600 MHz (typically 533 MHz) supporting OpenGL ES 1.1/2.0, Open G 1.1

- High performance dedicated 2D processor
- DDR3, DDR3L, LPDDR2 support
- Dual-panel display up to 2048x1536 resolution
- DDR3 Ram 512Mb x 2 8bit

Các kiểu giao tiếp hỗ trợ: I2C, I2S, GPIO, UART, USB, Ethernet

Các ngoại vi: LCD, RFID, Finger Printer, SD card, Keypad...

3.2. Build kernel hệ điều hành Ubuntu

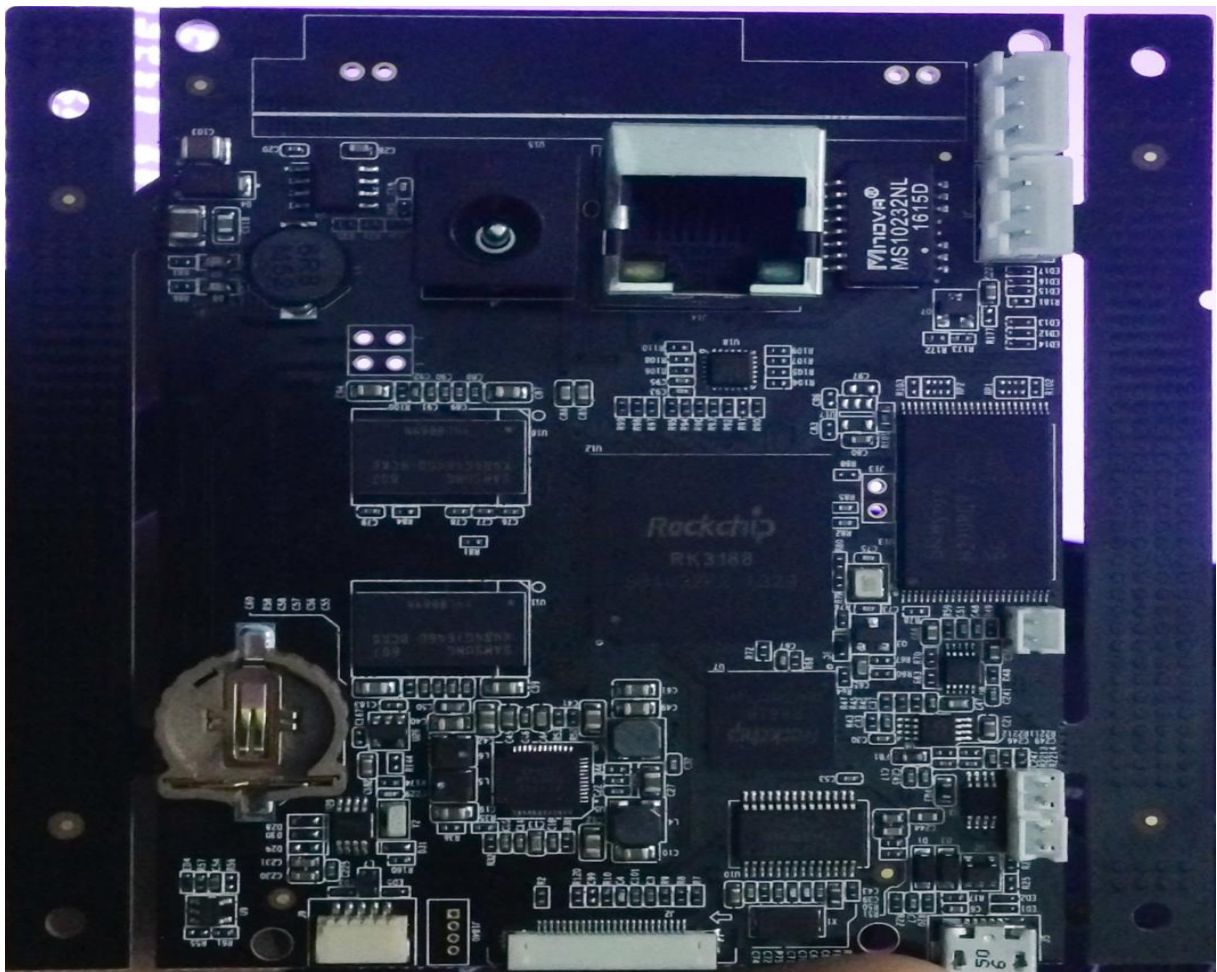
3.2.1. Giới thiệu kernel

GIT – Nói một cách đại thể, nhân Linux (kernel) là lớp phần mềm nằm giữa phần cứng và các chương trình ứng dụng chạy trên một máy tính. Kernel được lập trình bằng ngôn ngữ C và được Linus Torvalds phát triển.

3.2.2. Các bước tạo kernel

- Yêu cầu hệ thống

Board Radxa Rock chip RK3188



Desktop hoặc laptop chạy Linux 32/64bit

Cable micro usb

- Chuẩn bị môi trường

Trên Desktop Linux ta cần chuẩn bị môi trường để làm việc, bật Terminal và chạy các lệnh:

```
Sudo apt-get install build-essential lzop libncurses5-dev libssl-dev
```

```
Sudo apt-get install libc6:i386
```

Cài đặt gói git

```
Sudo apt-get install git
```

Tải và cài đặt gói của toolchain

```
git clone -b kitkat-release --depth 1
```

```
https://android.googlesource.com/platform/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6
```

- Build kernel

Tải source code

```
git clone -b radxa-stable-3.0 https://github.com/radxa/linux-rockchip.git
```

Build kernel:

Truy cập vào folder của source code:

```
cd linux-rockchip
```

Cấu hình cho LCD

```
make radxa_rock_pro_linux_lvds_defconfig
```

Cấu hình cho kernel


```
make radxa_rock_linux_defconfig
```

Biên dịch tạo file ảnh kernel.img

```
make -j8 kernel.img
```

Nén file ảnh kernel.img tạo file zkernel.img

```
make -j8 zkernel.img
```

- Build kernel modules

Build kernel modules

```
mkdir modules
```

```
make INSTALL_MOD_PATH=./modules modules modules_install
```

Sau khi chạy, ta có kernel modules và firmwave

- Tạo file image

Tạo initramfs

```
cd ..
```

```
git clone https://github.com/radxa/initrd.git
```

```
make -C initrd
```

Cài gói mkbooting:

```
git clone https://github.com/neo-technologies/rockchip-mkbooting.git
```

```
cd rockchip-mkbooting
```

```
make
```

```
sudo make install
```

```
cd ..
```

Tạo boot.img

```
mkbootimg --kernel linux-rockchip/arch/arm/boot/Image --ramdisk initrd.img -o boot.img
```

3.2.3. Tạo các file hệ thống rootfs

- Cài đặt gói cần thiết

```
sudo apt-get install qemu-user-static binfmt-support
```

- Tạo file image

```
dd if=/dev/zero of=rootfs.ext4 bs=1M count=1024  
mkfs.ext4 -F -L linuxroot rootfs.ext4  
sudo mount -o loop rootfs.ext4 /mnt  
wget http://releases.linaro.org/14.10/ubuntu/trusty-images/alip/linaro-trusty-alip-20141024-684.tar.gz  
sudo tar zxvf linaro-*.tar.gz -C /mnt  
cd /mnt  
sudo mv binary/*.  
sudo rmdir binary
```

Chép file modules đã build từ phần trên

```
sudo mkdir -p /mnt/lib/modules  
sudo cp -r /path/to/linux-rockchip/modules/lib/modules/3.0.36+ /mnt/lib/modules
```

- Chuẩn bị quyền root

```
sudo cp /usr/bin/qemu-arm-static /mnt/usr/bin  
sudo modprobe binfmt_misc  
sudo mount -t devpts devpts /mnt/dev/pts
```

```
sudo mount -t proc proc /mnt/proc
```

```
sudo chroot /mnt
```

- Tiếp tục cấu hình hệ thống

Chọn shell mặc định

```
rm /bin/sh && ln -s /bin/bash /bin/sh
```

Chuẩn bị cho cài đặt

```
apt-get update
```

```
apt-get install ssh vim usbutils wpasupplicant wireless-tools
```

Thêm các script vào file names

```
vim /usr/local/bin/mtd-by-name.sh
```

Thêm vào các dòng sau

```
#!/bin/sh -e  
  
# mtd-by-name link the mtdblock to name  
  
# radxa.com, thanks to naobsd  
  
rm -rf /dev/block/mtd/by-name/  
  
mkdir -p /dev/block/mtd/by-name  
  
for i in `ls -d /sys/class/mtd/mtd*[0-9]`; do  
  
name=`cat $i/name`  
  
tmp=""`echo $i | sed -e 's/mtd/mtdblock/g'`"  
  
dev=""`echo $tmp | sed -e 's/\sys\class\mtdblock\dev/g'`"  
  
ln -s $dev /dev/block/mtd/by-name/$name  
  
done
```

Thêm vào rc.local

```
vim /etc/rc.local
```

Thêm dòng sau vào trước “exit0”

```
/usr/local/bin/mtd-by-name.sh
```

Thoát khỏi

```
exit
```

```
sync
```

```
sudo umount /mnt
```

3.3. Boot kernel lên SDcard và NAND flash

3.3.1. Boot lên SD Card

- Tạo U-boot cho SD Card

```
git clone -b u-boot-rk3188-sdcard https://github.com/radxa/u-boot-rockchip.git
cd u-boot-rockchip
make rk30xx
./pack-sd.sh
```

- Hiệu chỉnh thông số

```
wget http://dl.radxa.com/rock/images/parameter/parameter_linux_sd
rkrcrc -p parameter_linux_sd parameter.img
```

- Boot lên SD Card

Xóa SD card

```
sudo dd if=/dev/zero of=/dev/sdx bs=1M count=1
```

Tạo phân vùng với fdisk

```
export START_SECTOR=65536  
  
sudo fdisk /dev/sdx << EOF  
  
n  
  
p  
  
l  
  
$START_SECTOR  
  
w  
  
EOF
```

Viết lên SD card

```
sudo dd if=u-boot-sd.img of=/dev/sdx conv=sync seek=64  
  
sudo dd if=parameter.img of=/dev/sdx conv=sync seek=$((0x2000))  
  
sudo dd if=boot.img of=/dev/sdx conv=sync seek=$((0x2000+0x2000))  
  
sudo dd if=rootfs.ext4 of=/dev/sdx1 conv=sync
```

3.3.2. Boot lên NAND

Cắm Board kết nối với máy tính thông qua cổng nối tiếp.

Vào chế độ Recovery: giữ nút SW3 (Recovery) và bấm nút SW2 (Reset).

Flash bootloader

```
sudo ./upgrade_tool ul RK3188Loader\ (L)_V2.08.bin
```

Flash thông số

```
sudo ./upgrade_tool di -p CMDPARM
```

Flash boot.img

```
sudo ./upgrade_tool di -b boot.img
```

Flash file systems

```
sudo ./upgrade_tool di linuxroot rootfs.ext4
```

3.3.3. Khởi động Linux trên board RK3188

Kết nối board và máy tính thông qua cổng nối tiếp trên máy tính.

Từ Ubuntu trên Desktop khởi động Terminal.

Cài đặt picocom.

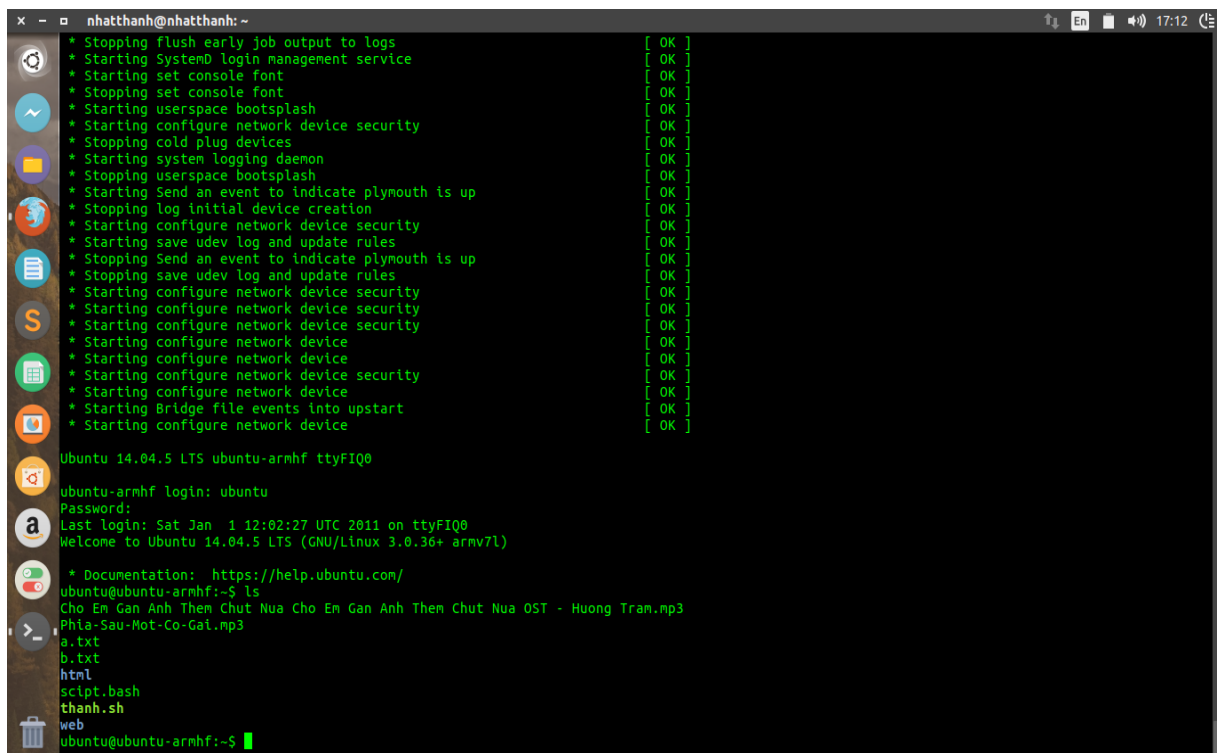
```
sudo apt-get install picocom
```

Gõ lệnh.

```
sudo picocom /dev/ttyUSB0 -b115200
```

Cấp nguồn cho board.

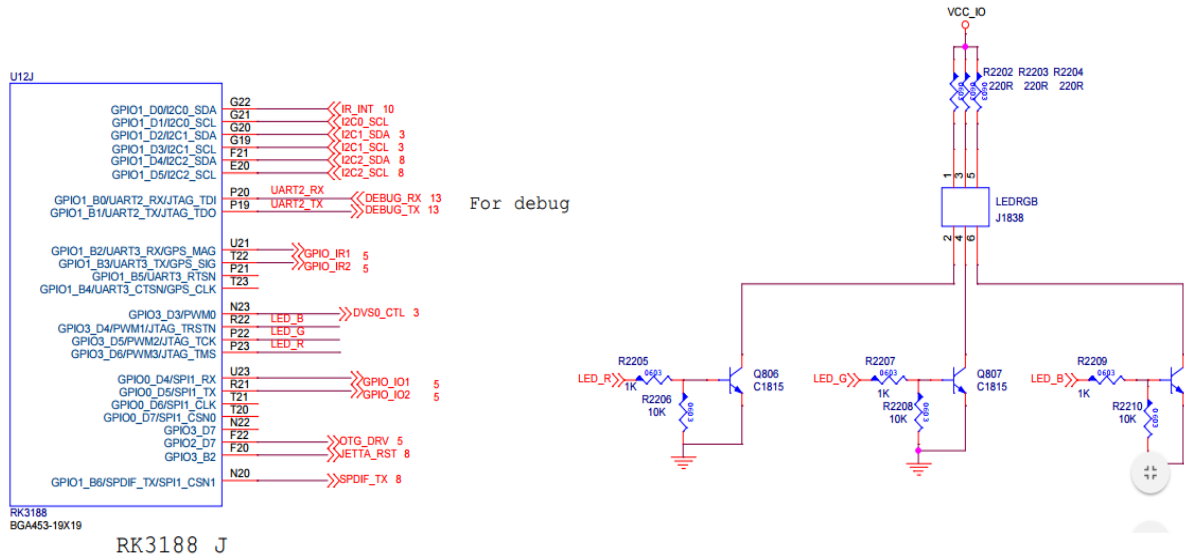
Sau khi khởi động, nhập user và password, ta đã vào được Ubuntu trên board.



```
nhatthanh@nhatthanh: ~  
* Stopping flush early job output to logs [ OK ]  
* Starting SystemD login management service [ OK ]  
* Starting set console font [ OK ]  
* Stopping set console font [ OK ]  
* Starting userspace bootplash [ OK ]  
* Starting configure network device security [ OK ]  
* Stopping cold plug devices [ OK ]  
* Starting system logging daemon [ OK ]  
* Stopping userspace bootplash [ OK ]  
* Starting Send an event to indicate plymouth is up [ OK ]  
* Stopping log initial device creation [ OK ]  
* Starting configure network device security [ OK ]  
* Starting save udev log and update rules [ OK ]  
* Stopping Send an event to indicate plymouth is up [ OK ]  
* Stopping save udev log and update rules [ OK ]  
* Starting configure network device security [ OK ]  
* Starting configure network device security [ OK ]  
* Starting configure network device security [ OK ]  
* Starting configure network device security [ OK ]  
* Starting configure network device security [ OK ]  
* Starting configure network device security [ OK ]  
* Starting Bridge file events into upstart [ OK ]  
* Starting configure network device [ OK ]  
Ubuntu 14.04.5 LTS ubuntu-armhf ttyFIQ0  
ubuntu-armhf login: ubuntu  
Password:  
Last login: Sat Jan 1 12:02:27 UTC 2011 on ttyFIQ0  
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.0.36+ armv7L)  
* Documentation: https://help.ubuntu.com/  
ubuntu@ubuntu-armhf:~$ ls  
Cho Em Gan Anh Them Chut Nua Cho Em Gan Anh Them Chut Nua OST - Huong Tran.mp3  
Phia-Sau-Mot-Co-Gai.mp3  
a.txt  
b.txt  
html  
script.bash  
thanh.sh  
web  
ubuntu@ubuntu-armhf:~$
```

3.4. Điều khiển ngoại vi đơn giản (LED RGB)

Chương 3



Dựa vào sơ đồ nguyên lý, ta xác định được các đèn Red, Green, Blue được gắn với các GPIO lần lượt là GPIO3_D6, GPIO3_D5, GPIO3_D4

Theo kernel 3.0.36

GPIO3_D4 = 284

GPIO3_D5 = 285

GPIO3_D6 = 286

Ta có được các đèn Red, Green, Blue ứng với các pin 286 285 284.

Ta thực hiện ví dụ: chớp tắt led Green bằng Shell

```
#!/bin/sh
# enable the gpio 285 -> green led
echo 285 > /sys/class/gpio/export
# set the direction to output
echo "out" > /sys/class/gpio/gpio285/direction
while true;
do
echo 0 > /sys/class/gpio/gpio285/value #led on
sleep 1
```

```
echo 1 > /sys/class/gpio/gpio285/value #led off  
sleep 1  
done
```

3.5. Xây dựng web server và điều khiển ngoại vi bằng web server

3.5.1. Viết web server đơn giản bằng html và css

HTML và CSS là các công nghệ cơ bản để xây dựng các trang Web: HTML (html và xhtml) cho cấu trúc, CSS cho phong cách và bố trí, bao gồm cả WebFonts. Tìm các nguồn tài nguyên cho trang web thiết kế tốt cũng như các công cụ hữu ích.

- Giới thiệu HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản, dùng để tạo ra các trang web.

HTML bao gồm một tập hợp các thẻ dùng để:

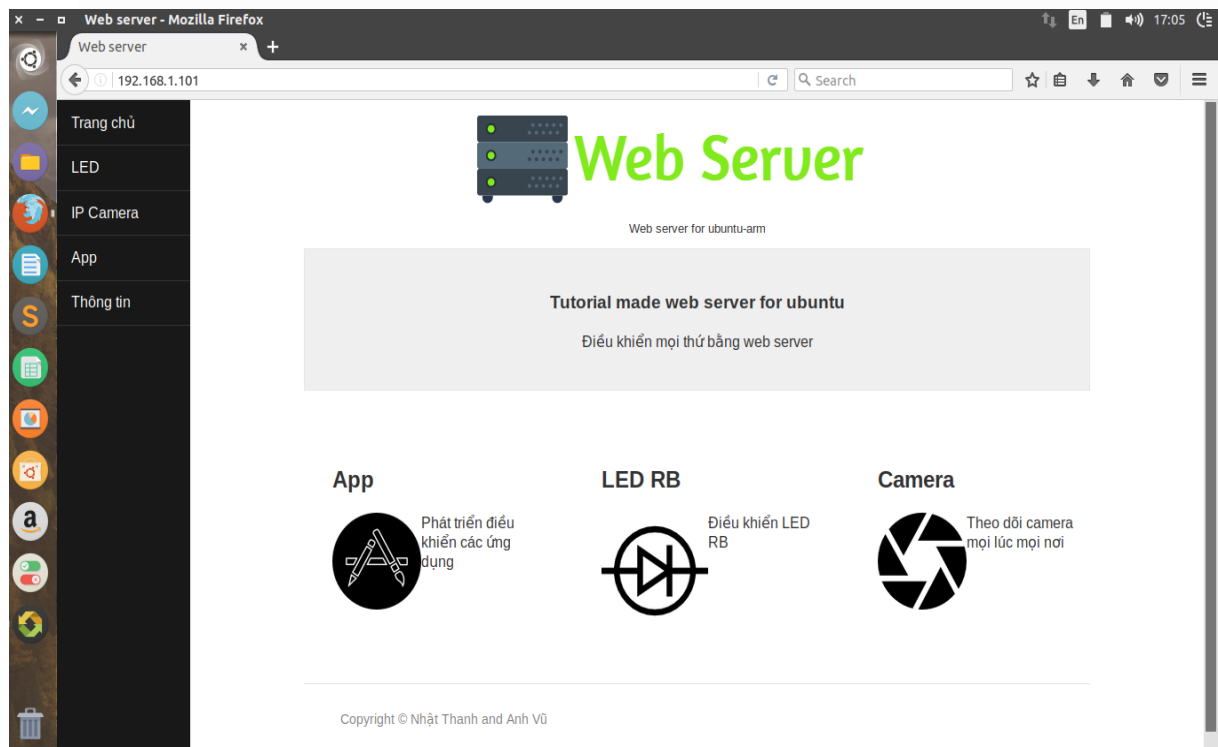
- Định nghĩa cấu trúc của trang web.
- Định dạng nội dung của trang web.
- Tạo các siêu liên kết để liên kết đến những trang web khác.
- Chèn âm thanh, hình ảnh, video, vào trang web

- Giới thiệu CCS

CSS (Cascading Style Sheets) là một ngôn ngữ dùng để định dạng cho các phần tử HTML (Ví dụ: chỉnh kích cỡ chữ, chỉnh font chữ, màu chữ, màu nền, hình nền, đường viền,...) dựa trên các cặp thuộc tính: giá trị thuộc tính

Với việc sử dụng CSS, ta có thể định dạng ra các phần tử HTML thật đặc biệt và chuyên nghiệp.

- Webserver đã hoàn thành



3.5.2. Cài đặt web server lên board

Cài đặt Apache Server

Tên phần mềm Apache trên Ubuntu sẽ là apache2 nên chúng ta sẽ cài đặt với lệnh:

```
Sudo apt-get install apache2
```

Cấu trúc của thư mục cấu hình Apache trên Ubuntu

conf-available/ – Thư mục này sẽ chứa các file thiết lập cấu hình sẵn của Apache trên Ubuntu, nhưng các thiết lập trong đây sẽ chưa được áp dụng vì Ubuntu không load thiết lập cấu hình trong thư mục này.

conf-enabled/ – Thư mục chứa các file thiết lập cấu hình của Apache trên Ubuntu đang được bật. Hãy hiểu là nếu thư mục này có một liên kết tượng trưng (symlink) qua một file module nào đó bên thư mục conf-available thì nó sẽ được bật.

mods-available/ – Thư mục chứa các file từng module của Apache trên Ubuntu nhưng chưa được bật.

mods-enabled/ – Thư mục chứa các file từng module của Apache trên Ubuntu đang được bật.

site-available/ – Thư mục chứa file cấu hình VirtualHost của Apache trên Ubuntu nhưng chưa được bật.

site-enabled/ – Thư mục chứa file cấu hình VirtualHost của Apache trên Ubuntu đang được bật.

apache2.conf – File cấu hình Apache trên Ubuntu.

envvars – File thiết lập các biến với giá trị sẵn để sử dụng trong các file cấu hình.

magic – File thiết lập của module mod_mime_magic trên Apache.

ports.conf – File cấu hình cổng mạng của Apache (mặc định là port 80).

Thêm Domain vào Apache

Tạo thư mục chứa

```
mkdir -p /home/webserver_name.com/public_html
```

```
mkdir -p /home/webserver_name.com/log
```

Sau đó chúng ta cần copy file /etc/apache2/sites-available/000-default.conf ra một file mới chứa cấu hình của domain cần thêm vào ở /etc/apache2/sites-available/webserver_name.com.conf.

```
cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/webserver_name.com.conf
```

Chỉnh sửa lại file config, thay

ServerName – Domain website cần thêm vào.

ServerAlias – Sử dụng một tên domain khác thay thế, hay còn gọi là Parked Domain nếu bạn đã từng sử dụng qua cPanel đó.

DocumentRoot – Đường dẫn tới thư mục chứa dữ liệu website của domain này mà ta đã tạo ở trên.

ErrorLog – Đường dẫn tới thư mục log đã tạo ở trên cho domain này.

CustomLog – Tương tự ErrorLog nhưng sẽ lưu log lại các lượt truy cập với file là access.log.

Gõ lệnh để kích hoạt

```
a2ensite webserver_name.com
```

Khởi động lại

```
service apache2 restart
```

Đẩy webserver vào board

Kết nối Board và Desktop Linux vào chung 1 mạng LAN

Kiểm tra IP của Desktop Linux và IP của Board.

Sử dụng lệnh để truy cập từ Desktop vào Board.

```
sudo sftp ubuntu@<địa chỉ ip của board>
```

Dùng lệnh get và put để đưa file từ Desktop qua Board. Đưa toàn bộ file html css của web vào thư mục /var/www/html

Truy cập vào web server bằng cách gõ IP của thiết bị trên trình duyệt.

3.5.3. Điều khiển LED thông qua web server

3.5.4. Giới thiệu về openhab

Openhab là phần mềm tích hợp nhiều hệ thống tự động hóa khác nhau, cung cấp giao diện cho người dùng thống nhất.

- Các tính năng của Openhab:

Có thể hoạt động trên nhiều hệ điều hành khác nhau (Linux, window, macOS)

Cho phép người dùng tích hợp nhiều thứ trong một.

Tốc độ hoạt động cao, đáp ứng tốt nhu cầu của người dùng

Đi kèm các trang web dựa trên giao diện khác nhau cho các bản Android, IOS

Mã nguồn mở

Được duy trì bởi một cộng đồng lớn

Dễ dàng mở rộng và tích hợp thêm hệ thống mới

Cung cấp các APIS cho các hệ thống khác.

- Cài đặt Openhab

Ta sẽ install OpenHAB trong thư mục /opt/openhab:

```
sudo mkdir /opt/openhab
```

```
cd /opt/openhab
```

Download OpenHAB ở trang download page của OpenHAB dùng version mới nhất (là 1.8.2 tại thời điểm bài viết). Nếu bạn download version mới hơn thì chỉ cần thay “1.8.2” trong các command dưới đây thành version đó là được. Trước tiên ta sẽ download file core runtime, giải nén và xóa file zip:

```
Sudo wget https://bintray.com/artifact/download/openhab/bin/distribution-1.8.2-runtime.zip
```

```
Sudo unzip distribution-1.8.2-runtime.zip
```

```
Sudo rm distribution-1.8.2-runtime.zip
```

Tiếp đến là các addons, ta sẽ download hết và bỏ vào folder riêng. Sau đó ta sẽ copy những addon nào cần mà thôi, giảm thời gian khởi động của OpenHAB:

```
Sudo mkdir addons_repo
```

```
Cd addons_repo
```

```
Sudo wget https://bintray.com/artifact/download/openhab/bin/distribution-1.8.2-addons.zip
```

```
Sudo unzip distribution-1.8.2-addons.zip
```

```
Sudo rm distribution-1.8.2-addons.zip
```

Trước mắt ta chỉ cần addon binding cho MQTT:

```
Cd /opt/openhab
```

```
Sudo cp addons_repo/org.openhab.binding.mqtt-1.8.2.jar
```

```
addons/org.openhab.binding.mqtt-1.8.2.jar
```

3.6. Kết luận

Sau khi hoàn thành các bước trên thì chúng ta sẽ hiểu được quá trình xây dựng một hệ thống nhúng và đã có được một hệ thống hoàn chỉnh. Tuy chỉ điều khiển LED đơn giản nhưng đó là tiền đề để điều khiển các ngoại vi lớn hơn mà board hỗ trợ.

Chương 4: Tổng kết và phương hướng phát triển đề tài

4.1. Đánh giá kết quả đạt được

Mặc dù chưa hoàn thành 100% đồ án được giao, nhưng nhóm chúng em cũng đã hoàn thành được khoảng 80% công việc. Và đã nắm được khá nhiều kiến thức từ đồ án này, cũng như được thực hành khá nhiều trên board, quen thuộc với hệ điều hành linux.

4.2. Hướng phát triển đề tài

Đề tài có rất nhiều hướng phát triển như phát triển các ứng dụng IoT, phát triển hệ thống thông minh lớn hơn, kết nối nhiều ngoại vi hơn.

Tài liệu tham khảo

- [1] Cadence (2008). *Allegro PCB Editor*. Cadence Design Systems Inc, USA.
- [2] Kraig Mitzner (2009). *Complete PCB Design Using OrCad Capture and PCB editor*. Elsevier Inc, USA.
- [3] Jon Duckett (2011). *HTML & CSS design and build websites*. John Wiley & Sons Inc, Indianapolis, Indiana.
- [4] Radxa team (2013). *<http://wiki.radxa.com/>*. Shenzhen, China.