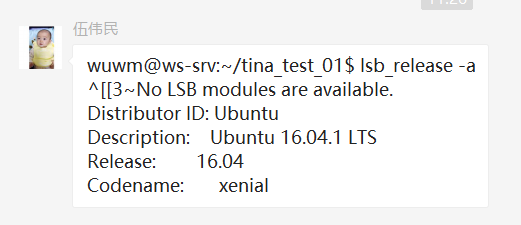
# 安装准备

Ubuntu 16.04 系统



sudo apt-get install vim

sudo apt-get install git-core

sudo apt-get install gawk

sudo apt-get install flex bison

sudo apt-get install texinfo

sudo apt-get install lib32ncurses5 lib32z1

sudo apt-get install u-boot-tools

sudo apt-get install lib32stdc++6

sudo apt-get install libncurses5-dev

sudo apt-get install zlib1g-dev

sudo apt-get install libssl-dev

//装这个的目的是为了装JSON-C 编译过程中 报错了

wget https://cmake.org/files/v3.3/cmake-3.3.2.tar.gz

tar xzvf cmake-3.3.2.tar.gz

cd cmake-3.3.2

./bootstrap

Gmake

make install

<https://github.com/json-c/json-c>

Json-c 这个安装最恶心

sudo apt install git

sudo apt install cmake

sudo apt install doxygen # optional

sudo apt install valgrind # optional

$ git clone https://github.com/json-c/json-c.git

$ mkdir json-c-build

$ cd json-c-build

$ cmake ../json-c # See CMake section below for custom arguments

$ make

$ make test

$ make USE\_VALGRIND=0 test # optionally skip using valgrind

$ make install

# Tina编译命令

编译命令

$ source build/envsetup.sh

$ lunch

$ make menuconfig

$ make kernel\_menuconfig

$ make

$ pack

之后有提示输出

/home/dj/tina\_aw3.5.1/out/c200s-F1C200s/tina\_c200s-F1C200s\_uart0.img

查找文件

grep "sys\_config.fex" ./ -nr

当前文件夹查找文件

sudo find ./ -name 'sys\_config.fex' -print

# 打补丁

Spi nand FLASH 补丁文件

解压补丁压缩包，打补丁

tar -zxvf R6\_Update\_Nand\_To\_3.6013\_2018-09-06.tar.gz

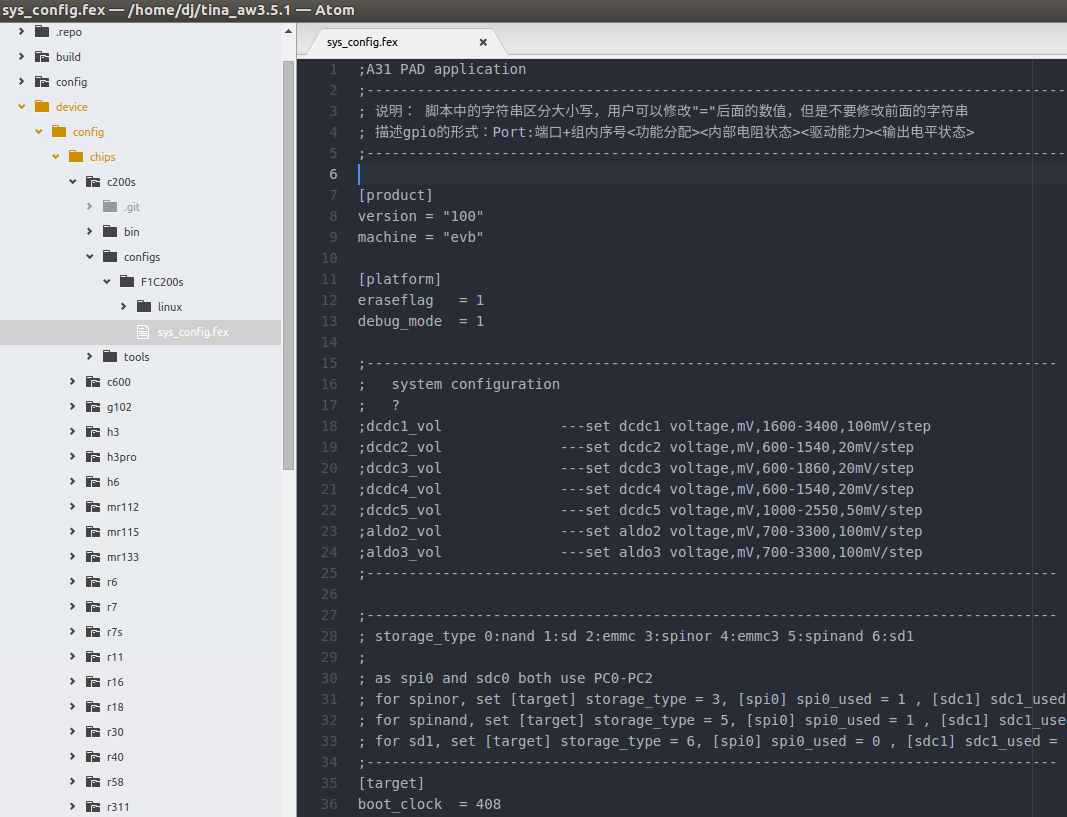
cd R6\_Update\_Nand\_To\_3.6013\_2018-09-06/

./install.sh ../tina/

回到 tina 目录

muboot //编译 uboot

# SPI NAND



1.目前只支持一个型号<MX35LF1GE4AB, 3V, 1Gb, v1.5.pdf>



[target]

boot\_clock = 408

**storage\_type = 5**

burn\_key = 0

打开 spi0，关闭 sdc1

[spi0]

**spi0\_used = 1**

spi0\_cs\_number = 1

spi0\_cs\_bitmap = 1

spi0\_cs0 = port:PC1<2><1><default><default>

spi0\_sclk = port:PC0<2><default><default><default>

spi0\_mosi = port:PC3<2><default><default><default>

spi0\_miso = port:PC2<2><default><default><default>

[sdc1]

**sdc1\_used = 0**

sdc1\_detmode = 4

sdc1\_buswidth = 1

;sdc1\_clk = port:PC00<3><1><2><default>

;sdc1\_cmd = port:PC01<3><1><2><default>

;sdc1\_d0 = port:PC02<3><1><2><default>

sdc1\_det =

sdc1\_use\_wp = 0

sdc1\_wp =

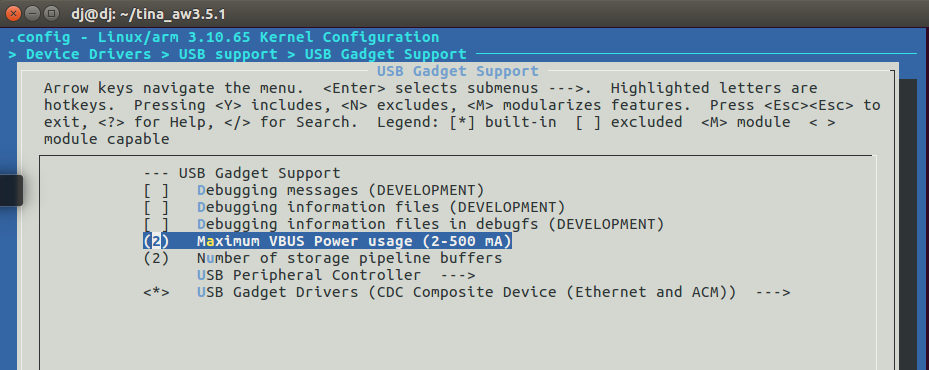
sdc1\_isio = 1

sdc1\_regulator = "none"

# 烧写方法



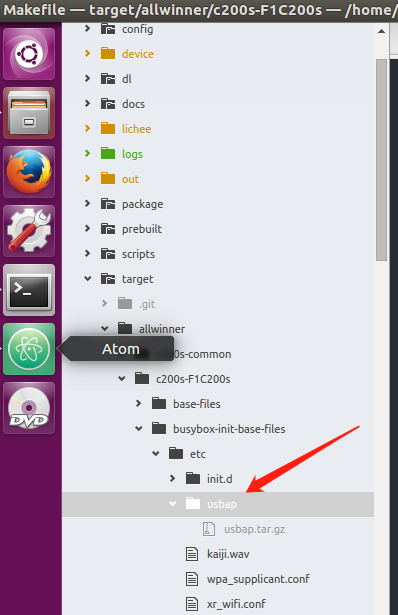
添加usb网卡支持



生成这么几个驱动文件

cdc-wdm.ko cdc\_ether.ko cdc\_ncm.ko usbap.tar.gz usbnet.ko

新建一个这个目录，然后吧内容打包进去，不打包编译不过去，然后烧写进去



**然后启动的时候提示你 驱动加载成功了**

