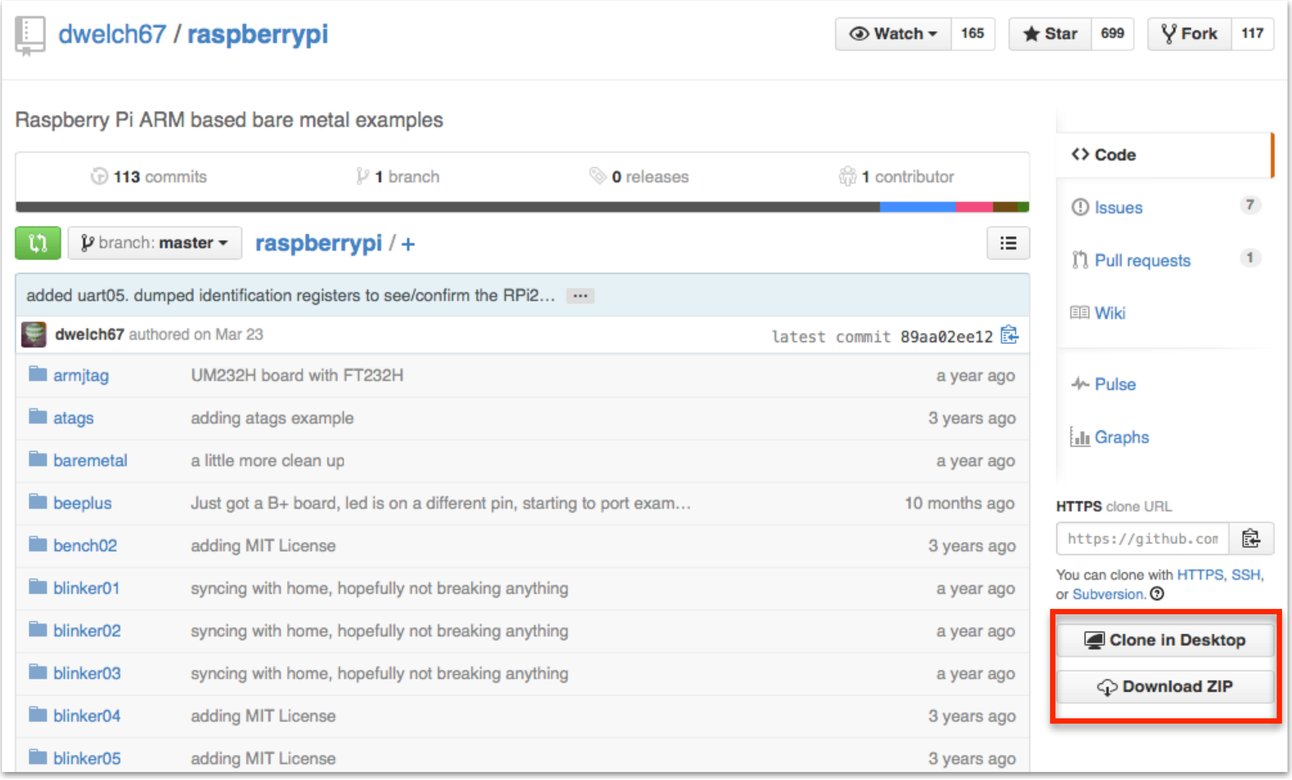
Assignment 004: Lab 4： bootloader

葛现隆 3120102146

一、 实验目的
1 理解uboot等bootloader的一般功能和基本工作原理；
2 掌握在三个平台上编写裸机程序并下载运行的方法。

二、 实验器材
• Raspberry板一块
• 5V/1A电源一个；
• microUSB线一根；
• PC（Mac OS）一台；

三、 实验步骤
1 通过Github下载raspberrpi-master文件(https://github.com/dwelch67/raspberrypi)；



dwelch67 / raspberrypi

Watch 165 | Star 699 | Fork 117

Raspberry Pi ARM based bare metal examples

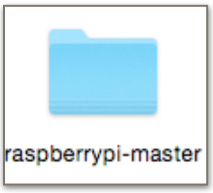
113 commits | 1 branch | 0 releases | 1 contributor

branch: master | raspberrypi / +

added uart05. dumped identification registers to see/confirm the RPi2...

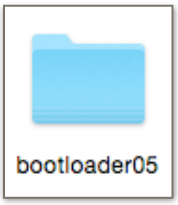
dwelch67 authored on Mar 23 | latest commit 89aa02ee12

armjtag	UM232H board with FT232H	a year ago
atags	adding atags example	3 years ago
baremetal	a little more clean up	a year ago
beeplus	Just got a B+ board, led is on a different pin, starting to port exam...	10 months ago
bench02	adding MIT License	3 years ago
blinker01	syncing with home, hopefully not breaking anything	a year ago
blinker02	syncing with home, hopefully not breaking anything	a year ago
blinker03	syncing with home, hopefully not breaking anything	a year ago
blinker04	adding MIT License	3 years ago
blinker05	adding MIT License	3 years ago



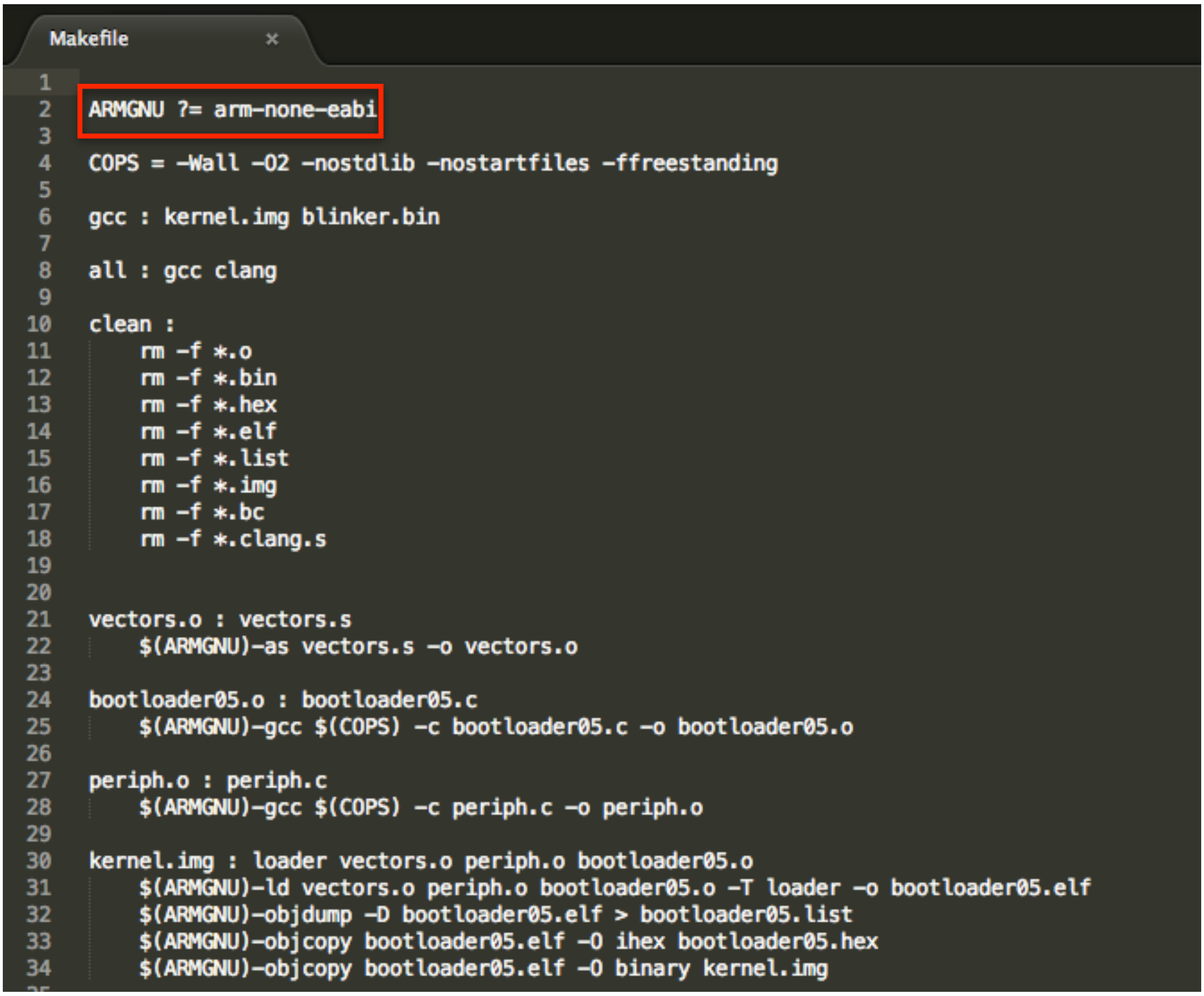
raspberrypi-master

2 尝试对bootloader05文件下文件进行编译，执行make指令，发现编译不通过，缺损arm-none-eabi-as；



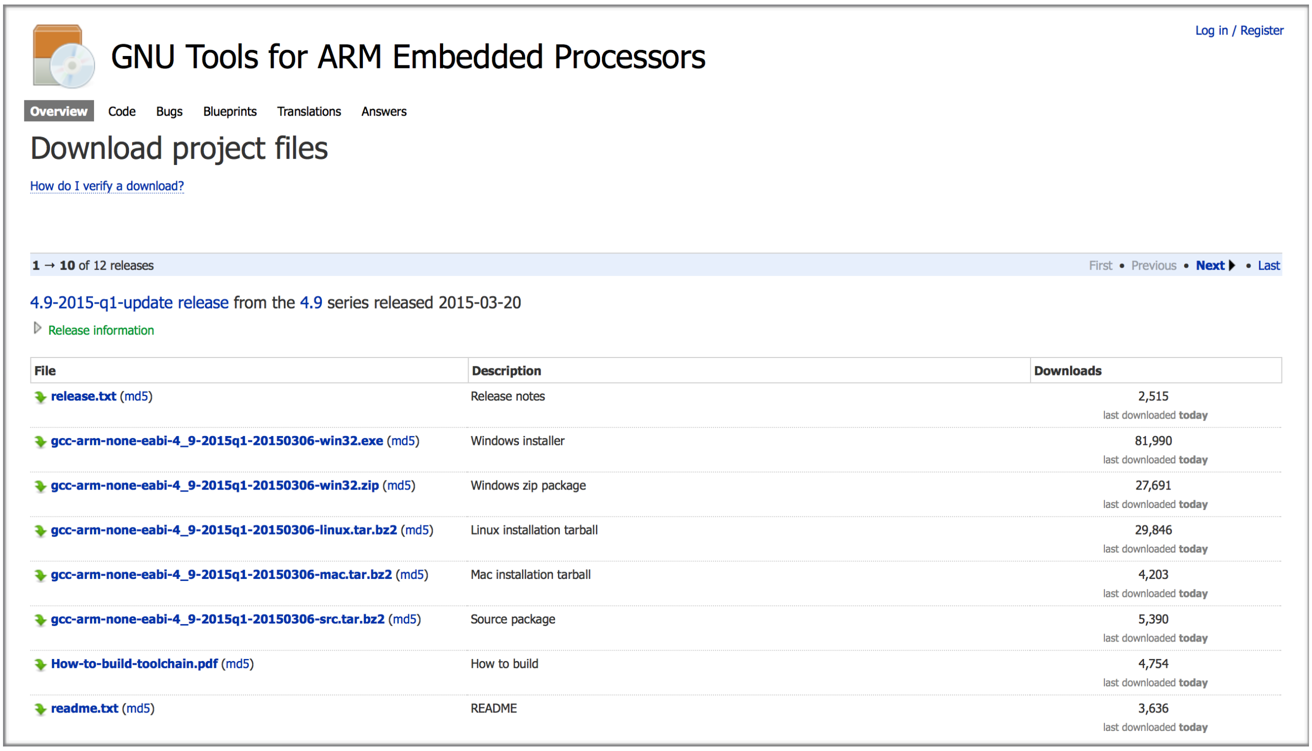
bootloader05

3 查看Makefile文件，发现需要arm-none-eabi编译；



```

ARMGNU ?= arm-none-eabi

COPS = -Wall -O2 -nostdlib -nostartfiles -ffreestanding

gcc : kernel.img blinker.bin

all : gcc clang

clean :
	rm -f *.o
	rm -f *.bin
	rm -f *.hex
	rm -f *.elf
	rm -f *.list
	rm -f *.img
	rm -f *.bc
	rm -f *.clang.s


vectors.o : vectors.s
	$(ARMGNU)-as vectors.s -o vectors.o

bootloader05.o : bootloader05.c
	$(ARMGNU)-gcc $(COPS) -c bootloader05.c -o bootloader05.o

periph.o : periph.c
	$(ARMGNU)-gcc $(COPS) -c periph.c -o periph.o

kernel.img : loader vectors.o periph.o bootloader05.o
	$(ARMGNU)-ld vectors.o periph.o bootloader05.o -T loader -o bootloader05.elf
	$(ARMGNU)-objdump -D bootloader05.elf > bootloader05.list
	$(ARMGNU)-objcopy bootloader05.elf -O ihex bootloader05.hex
	$(ARMGNU)-objcopy bootloader05.elf -O binary kernel.img
```

4 下载arm-none-eabi，下载地址：https://launchpad.net/gcc-arm-embedded/+download；



GNU Tools for ARM Embedded Processors

Download project files

How do I verify a download?

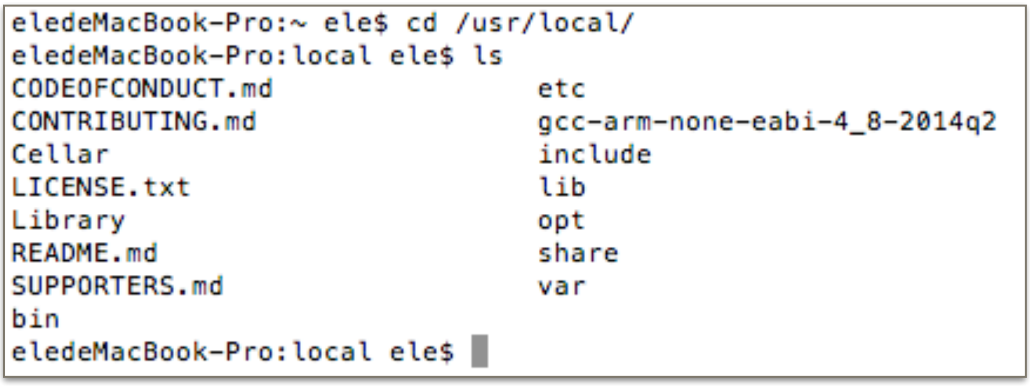
1 → 10 of 12 releases

4.9-2015-q1-update release from the 4.9 series released 2015-03-20

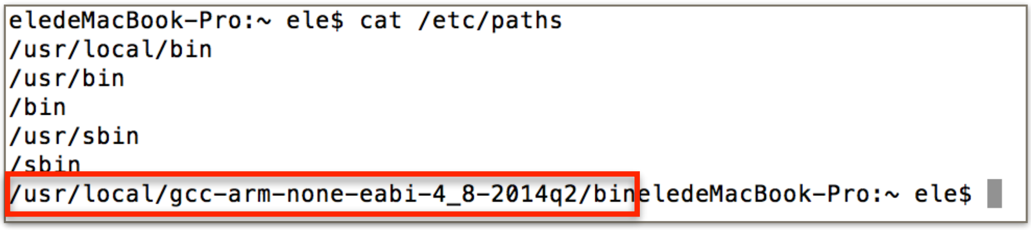
▷ Release information

File	Description	Downloads
release.txt (md5)	Release notes	2,515 last downloaded today
gcc-arm-none-eabi-4_9-2015q1-20150306-win32.exe (md5)	Windows installer	81,990 last downloaded today
gcc-arm-none-eabi-4_9-2015q1-20150306-win32.zip (md5)	Windows zip package	27,691 last downloaded today
gcc-arm-none-eabi-4_9-2015q1-20150306-linux.tar.bz2 (md5)	Linux installation tarball	29,846 last downloaded today
gcc-arm-none-eabi-4_9-2015q1-20150306-mac.tar.bz2 (md5)	Mac installation tarball	4,203 last downloaded today
gcc-arm-none-eabi-4_9-2015q1-20150306-src.tar.bz2 (md5)	Source package	5,390 last downloaded today
How-to-build-toolchain.pdf (md5)	How to build	4,754 last downloaded today
readme.txt (md5)	README	3,636 last downloaded today

5 配置gcc-arm-none-eabi，将gcc-arm-none-eabi压缩包解压到/usr/local目录下，在/etc/paths文件中添加路径；

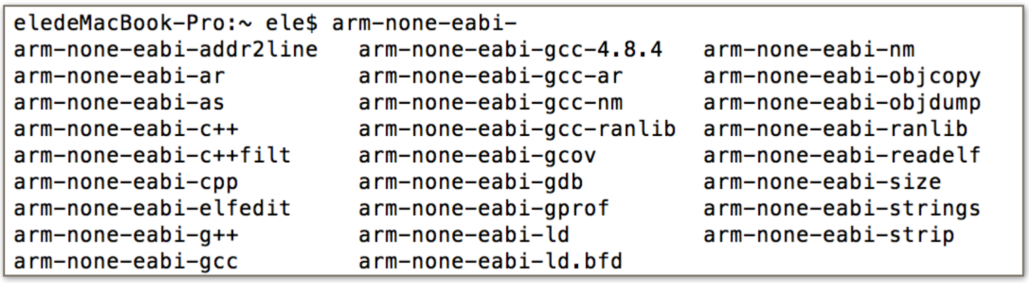


```
eledeMacBook-Pro:~ ele$ cd /usr/local/
eledeMacBook-Pro:local ele$ ls
CODEOFCONDUCT.md                etc
CONTRIBUTING.md                 gcc-arm-none-eabi-4_8-2014q2
Cellar                          include
LICENSE.txt                     lib
Library                         opt
README.md                       share
SUPPORTERS.md                   var
bin
eledeMacBook-Pro:local ele$ 
```



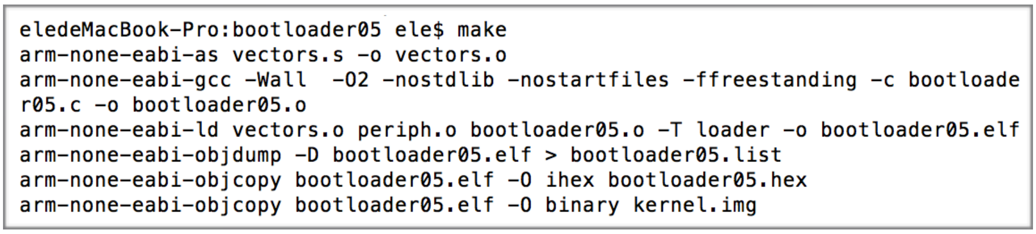
```
eledeMacBook-Pro:~ ele$ cat /etc/paths
/usr/local/bin
/usr/bin
/bin
/usr/sbin
/sbin
/usr/local/gcc-arm-none-eabi-4_8-2014q2/bineledeMacBook-Pro:~ ele$ 
```

6 重启终端，查看arm-none-eabi-*指令已添加；



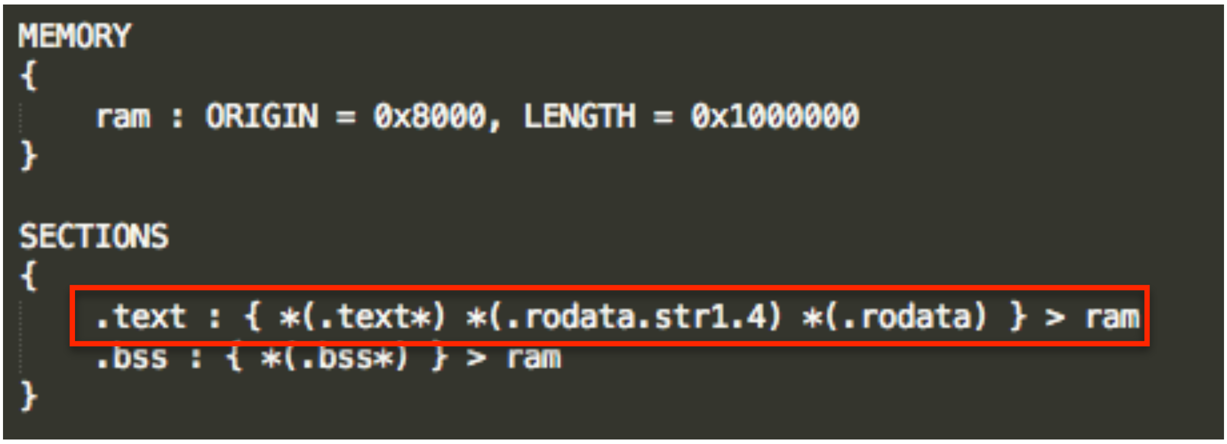
```
eledeMacBook-Pro:~ ele$ arm-none-eabi-
arm-none-eabi-addr2line   arm-none-eabi-gcc-4.8.4   arm-none-eabi-nm
arm-none-eabi-ar          arm-none-eabi-gcc-ar      arm-none-eabi-objcopy
arm-none-eabi-as          arm-none-eabi-gcc-nm      arm-none-eabi-objdump
arm-none-eabi-c++         arm-none-eabi-gcc-ranlib  arm-none-eabi-ranlib
arm-none-eabi-c++filt     arm-none-eabi-gcov        arm-none-eabi-readelf
arm-none-eabi-cpp         arm-none-eabi-gdb         arm-none-eabi-size
arm-none-eabi-elfedit     arm-none-eabi-gprof       arm-none-eabi-strings
arm-none-eabi-g++         arm-none-eabi-ld          arm-none-eabi-strip
arm-none-eabi-gcc         arm-none-eabi-ld.bfd
```

7 可正常编译；



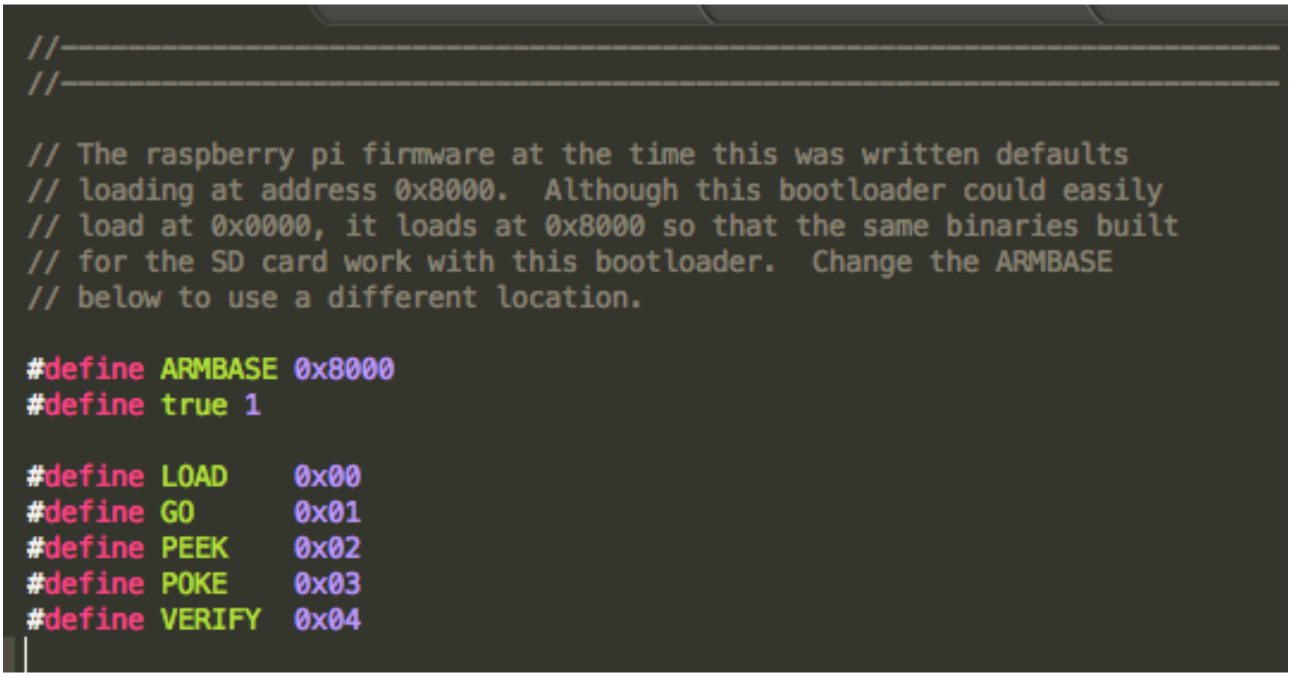
```
eledeMacBook-Pro:bootloader05 ele$ make
arm-none-eabi-as vectors.s -o vectors.o
arm-none-eabi-gcc -Wall  -O2 -nostdlib -nostartfiles -ffreestanding -c bootloade
r05.c -o bootloader05.o
arm-none-eabi-ld vectors.o periph.o bootloader05.o -T loader -o bootloader05.elf
arm-none-eabi-objdump -D bootloader05.elf > bootloader05.list
arm-none-eabi-objcopy bootloader05.elf -O ihex bootloader05.hex
arm-none-eabi-objcopy bootloader05.elf -O binary kernel.img
```

8 对bootloader05文件夹下load文件进行适当修改；



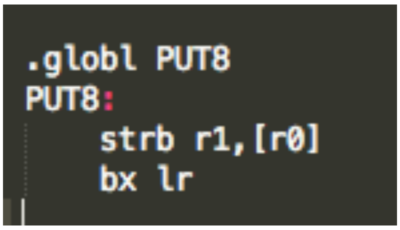
```
MEMORY
{
    ram : ORIGIN = 0x8000, LENGTH = 0x1000000
}

SECTIONS
{
    .text : { *(.text*) *(.rodata.str1.4) *(.rodata) } > ram
    .bss : { *(.bss*) } > ram
}
```

9 对bootloader05文件夹下bootloader05.c文件进行适当修改，将原有的3个控制状态改写成7个控制状态，新增Go, Verify, Peek, Poke, Load状态；
Go : 跳转到Base Address读取并且执行代码；
Verify : 检查到已读到的数据与新的bin文件不一致,并将不一致地址和数据返回；
Peek : 后跟address参数(32bits)，得到该内存地址上存的值，打印输出；
Poke : 后跟address参数(32bits)和修改data(32位)，修改address内存地址数据为data，打印结果;
LOAD : 加载二进制文件;



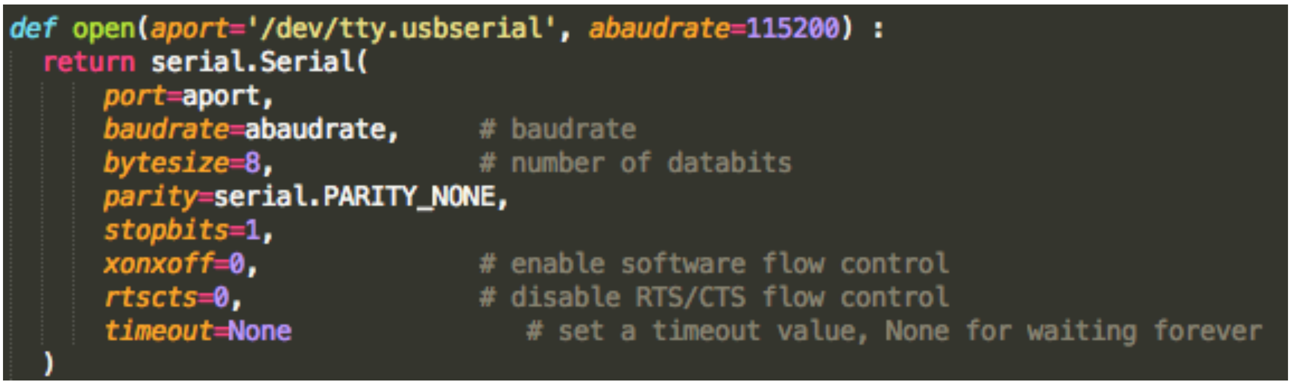
```
//------------------------------------------------------------------------
//------------------------------------------------------------------------

// The raspberry pi firmware at the time this was written defaults
// loading at address 0x8000.  Although this bootloader could easily
// load at 0x0000, it loads at 0x8000 so that the same binaries built
// for the SD card work with this bootloader.  Change the ARMBASE
// below to use a different location.

#define ARMBASE 0x8000
#define true 1

#define LOAD    0x00
#define GO      0x01
#define PEEK    0x02
#define POKE    0x03
#define VERIFY  0x04
```

10 对bootloader05文件夹下vector.s文件进行适当修改，添加GET8；

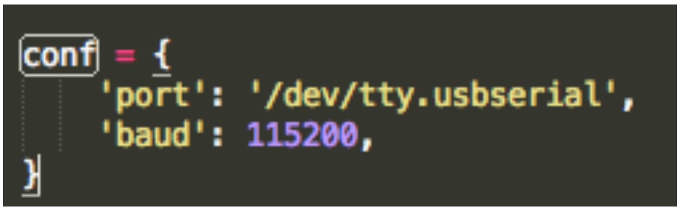


```
.globl PUT8
PUT8:
    strb r1,[r0]
    bx lr
```

11 修改xmodem-loader.py文件，修正串口，修改指令操作；

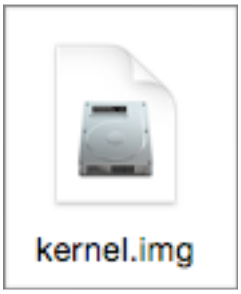


```
def open(aport='/dev/tty.usbserial', abaudrate=115200) :
  return serial.Serial(
      port=aport,
      baudrate=abaudrate,     # baudrate
      bytesize=8,             # number of databits
      parity=serial.PARITY_NONE,
      stopbits=1,
      xonxoff=0,              # enable software flow control
      rtscts=0,               # disable RTS/CTS flow control
      timeout=None               # set a timeout value, None for waiting forever
  )
```

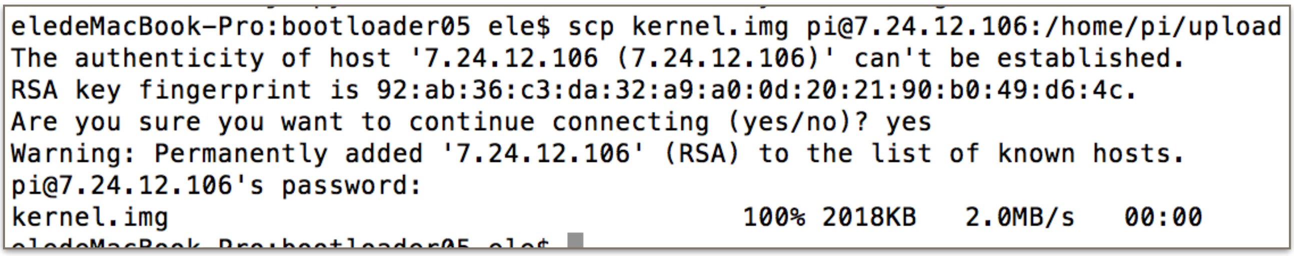


```
conf = {
    'port': '/dev/tty.usbserial',
    'baud': 115200,
}
```

12 再次make进行编译，获得kernel.img；

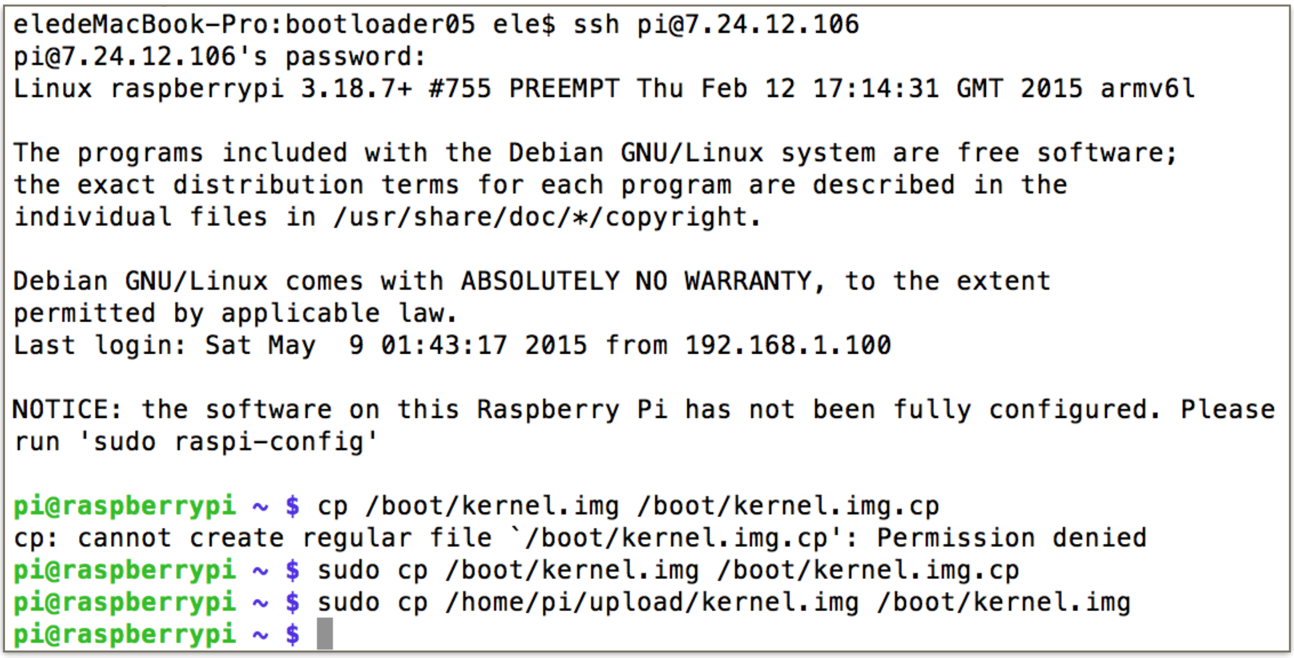


13 链接raspberry到路由器，通过scp指令将文件传递给raspberry；



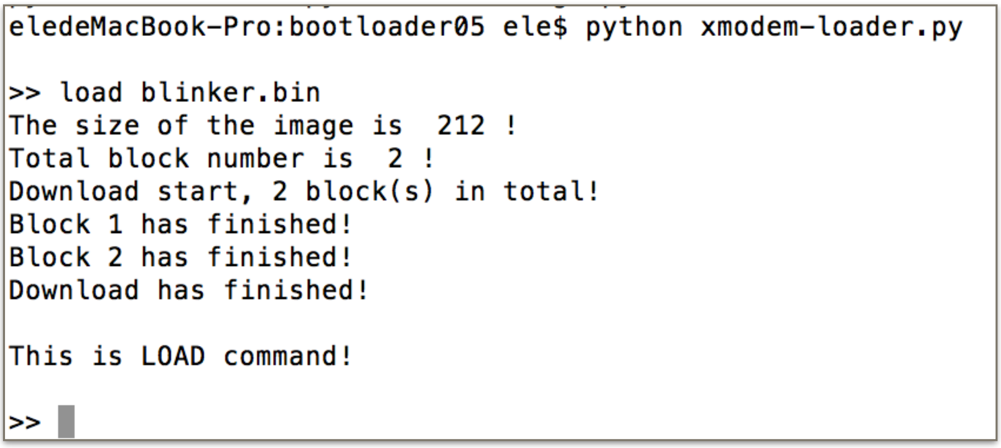
```
eledeMacBook-Pro:bootloader05 ele$ scp kernel.img pi@7.24.12.106:/home/pi/upload
The authenticity of host '7.24.12.106 (7.24.12.106)' can't be established.
RSA key fingerprint is 92:ab:36:c3:da:32:a9:a0:0d:20:21:90:b0:49:d6:4c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '7.24.12.106' (RSA) to the list of known hosts.
pi@7.24.12.106's password:
kernel.img                                    100% 2018KB   2.0MB/s   00:00
eledeMacBook-Pro:bootloader05 ele$
```

14 通过ssh连接raspberry，将原kernel.img进行备份，并将生产kernel.img复制到/boot/目录下，重启raspberrypi；



```
eledeMacBook-Pro:bootloader05 ele$ ssh pi@7.24.12.106
pi@7.24.12.106's password:
Linux raspberrypi 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May  9 01:43:17 2015 from 192.168.1.100

NOTICE: the software on this Raspberry Pi has not been fully configured. Please
run 'sudo raspi-config'

pi@raspberrypi ~ $ cp /boot/kernel.img /boot/kernel.img.cp
cp: cannot create regular file `/boot/kernel.img.cp': Permission denied
pi@raspberrypi ~ $ sudo cp /boot/kernel.img /boot/kernel.img.cp
pi@raspberrypi ~ $ sudo cp /home/pi/upload/kernel.img /boot/kernel.img
pi@raspberrypi ~ $
```

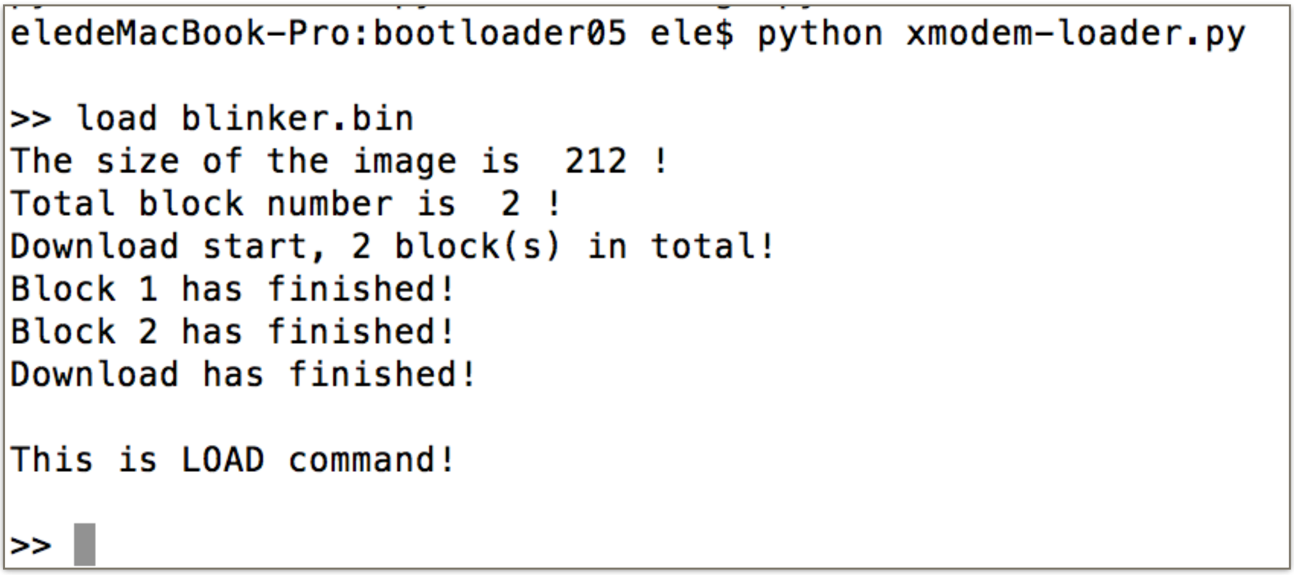
15 连接raspberry与pc之间的串口，运行py文件



```
eledeMacBook-Pro:bootloader05 ele$ python xmodem-loader.py

>> load blinker.bin
The size of the image is  212 !
Total block number is  2 !
Download start, 2 block(s) in total!
Block 1 has finished!
Block 2 has finished!
Download has finished!

This is LOAD command!

>>
```

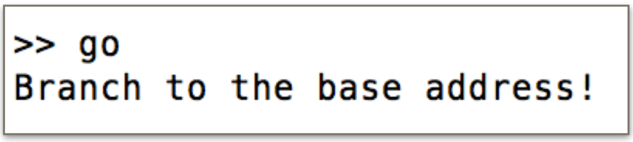
四、 实验结果

1 执行load指令，加载blinker.bin文件；

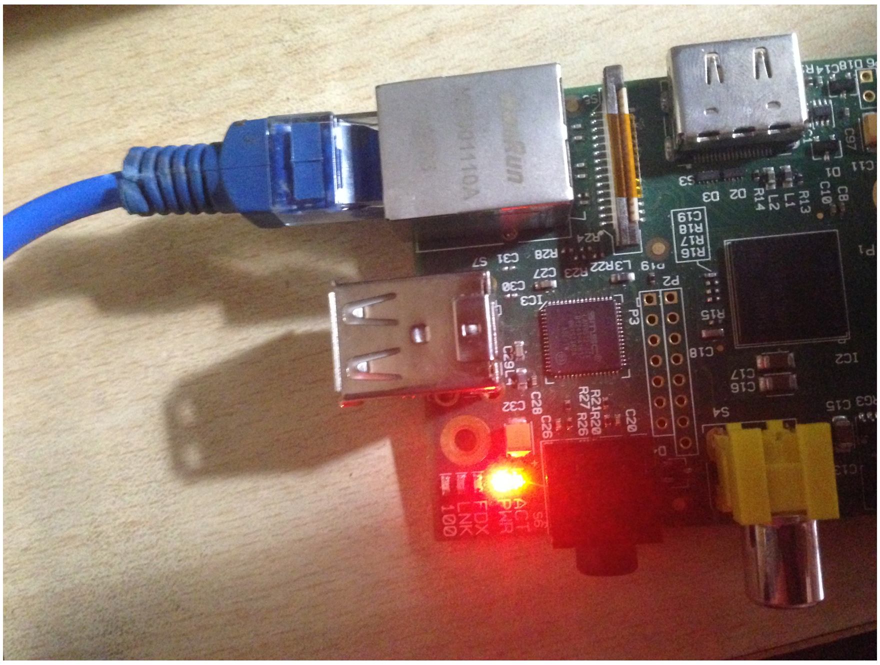


```
eledeMacBook-Pro:bootloader05 ele$ python xmodem-loader.py

>> load blinker.bin
The size of the image is  212 !
Total block number is  2 !
Download start, 2 block(s) in total!
Block 1 has finished!
Block 2 has finished!
Download has finished!

This is LOAD command!

>> 
```

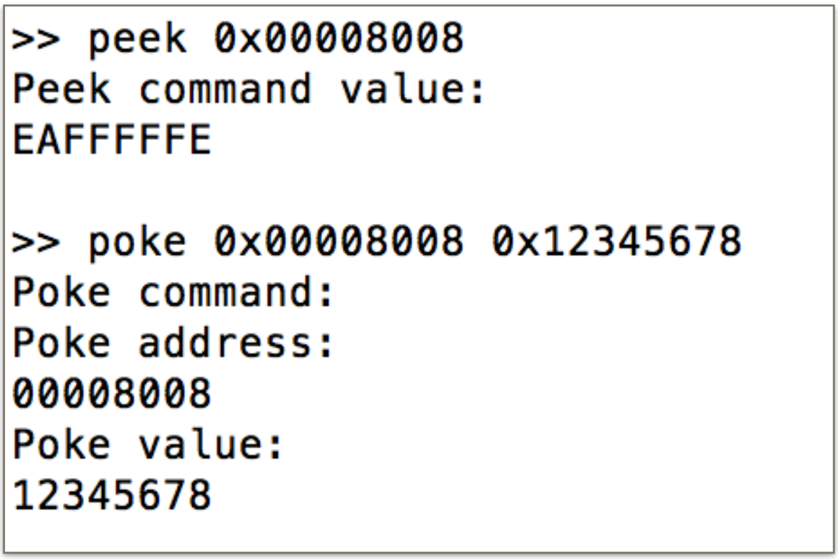
2 执行go指令，执行结果是板子上ACT灯缓慢闪烁；



```
>> go
Branch to the base address!
```

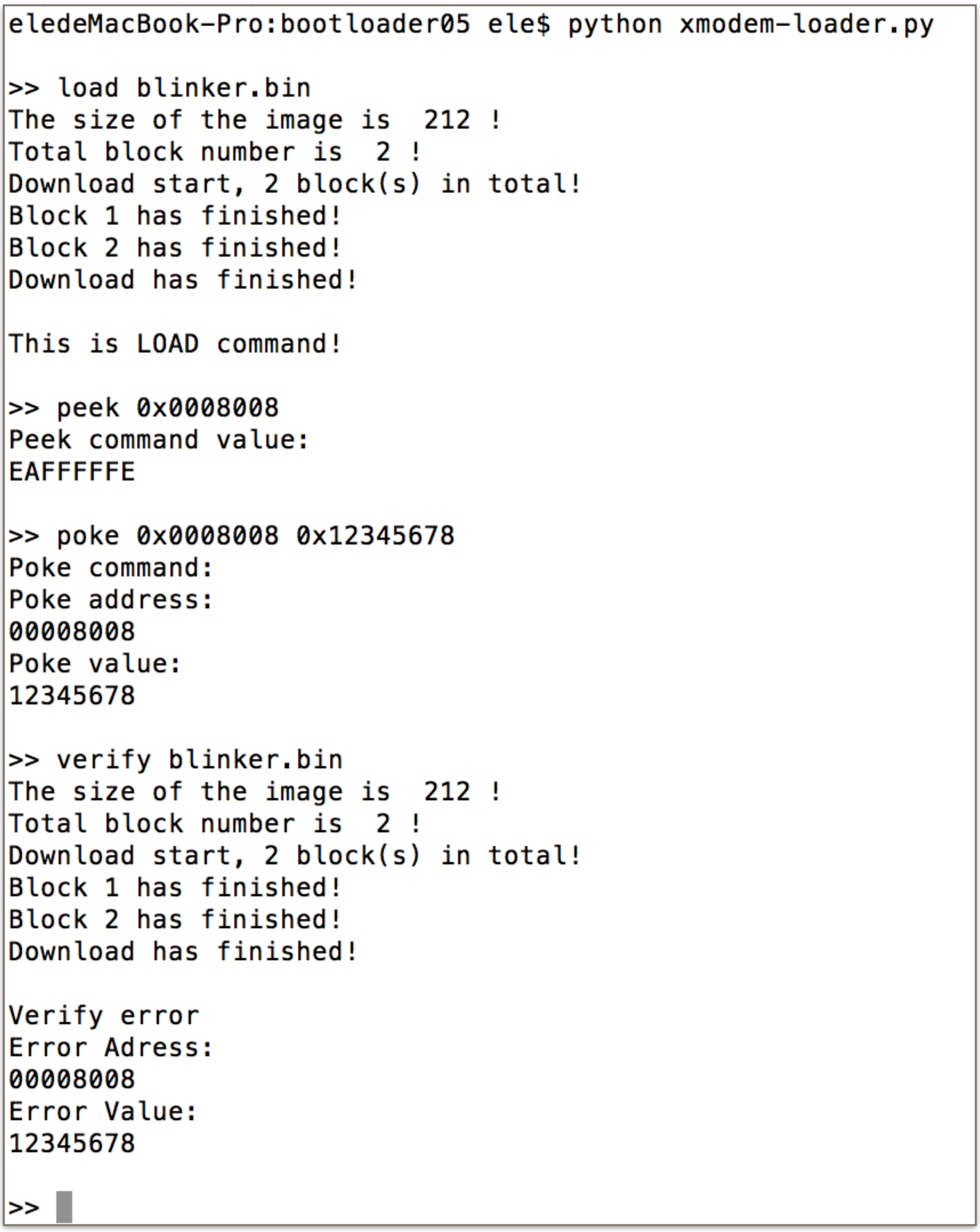


3 执行peek指令，获得对应地址数值，并通过poke修改对应数值；



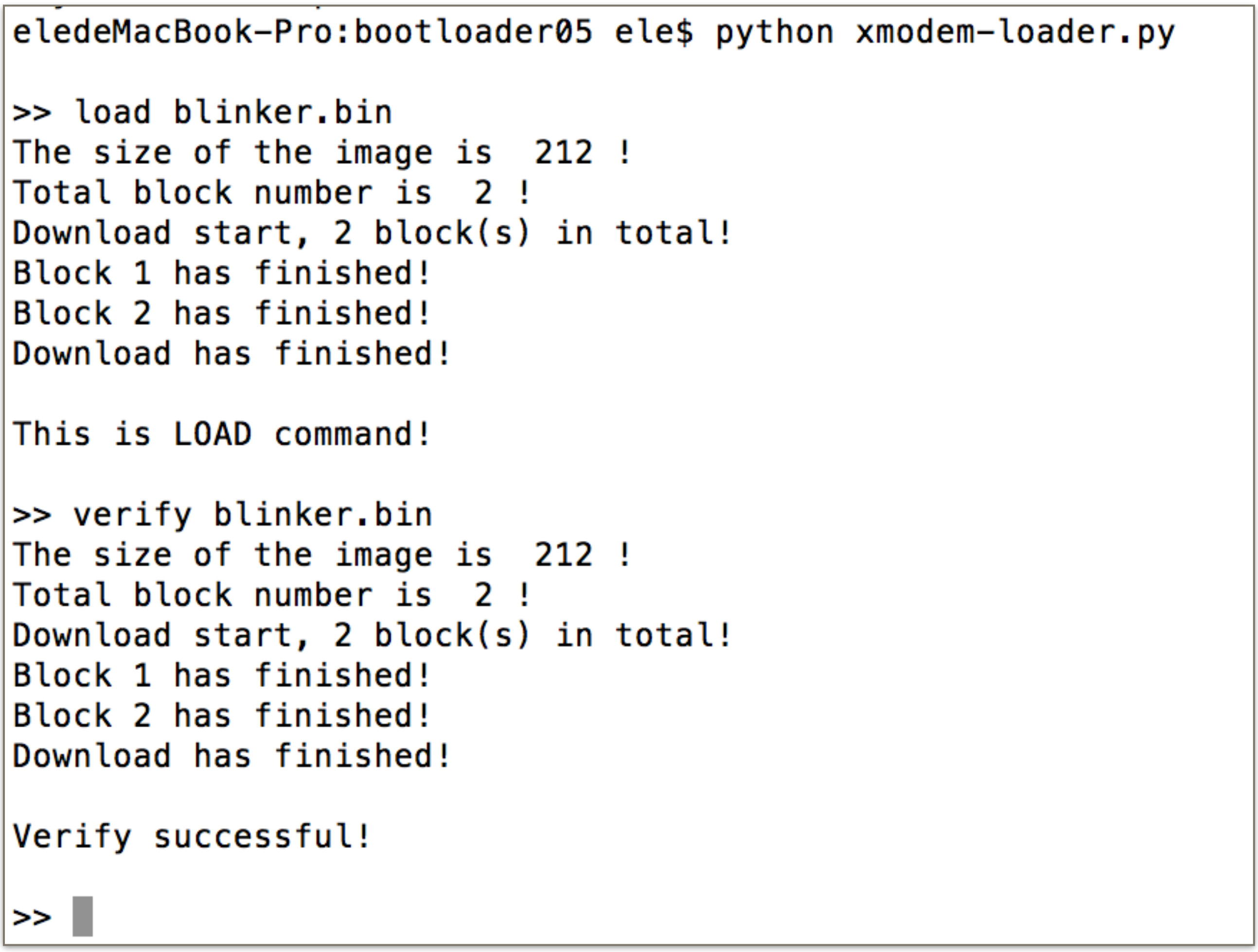
```
>> peek 0x00008008
Peek command value:
EAFFFFFE

>> poke 0x00008008 0x12345678
Poke command:
Poke address:
00008008
Poke value:
12345678
```

4 执行verify指令，输出error结果，x00008008地址数据不符合；



```
eledeMacBook-Pro:bootloader05 ele$ python xmodem-loader.py

>> load blinker.bin
The size of the image is  212 !
Total block number is  2 !
Download start, 2 block(s) in total!
Block 1 has finished!
Block 2 has finished!
Download has finished!

This is LOAD command!

>> peek 0x0008008
Peek command value:
EAFFFFFE

>> poke 0x0008008 0x12345678
Poke command:
Poke address:
00008008
Poke value:
12345678

>> verify blinker.bin
The size of the image is  212 !
Total block number is  2 !
Download start, 2 block(s) in total!
Block 1 has finished!
Block 2 has finished!
Download has finished!

Verify error
Error Adress:
00008008
Error Value:
12345678

>>
```

5 在没有修改的情况下指令verify，输出结果正确；



```
eledeMacBook-Pro:bootloader05 ele$ python xmodem-loader.py

>> load blinker.bin
The size of the image is  212 !
Total block number is  2 !
Download start, 2 block(s) in total!
Block 1 has finished!
Block 2 has finished!
Download has finished!

This is LOAD command!

>> verify blinker.bin
The size of the image is  212 !
Total block number is  2 !
Download start, 2 block(s) in total!
Block 1 has finished!
Block 2 has finished!
Download has finished!

Verify successful!

>> 
```