在Raspberry Pi 3上建構簡易Linux系統

開發學生: 賴郁文

開發教師: 陳鵬升

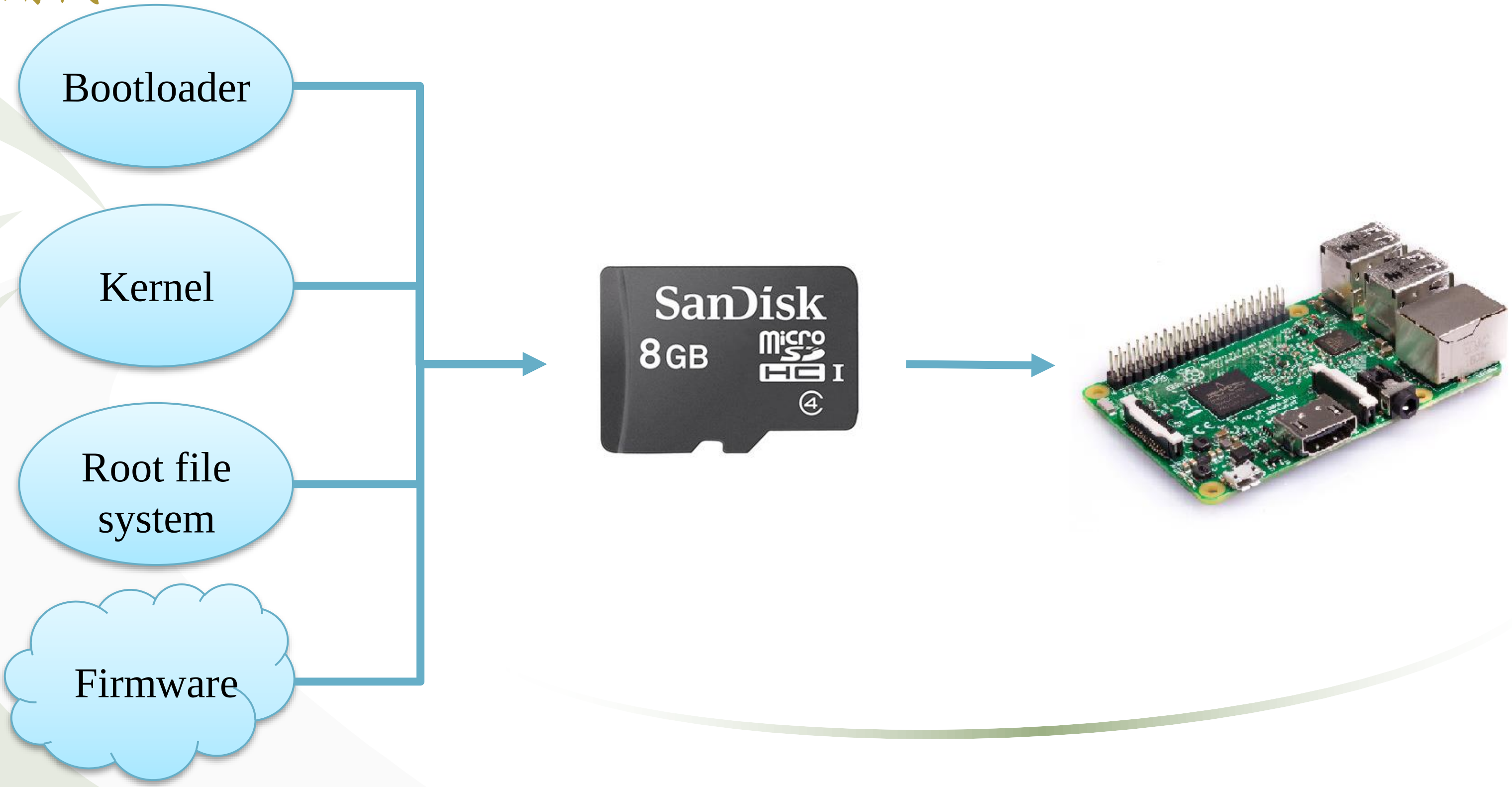
國立中正大學 資訊工程學系

Outline

- 實驗目的
- 軟硬體與環境需求
- Raspberry Pi 3介紹
- 簡易Linux系統架構
- Buildroot介紹
- 實驗流程
- Bootloader & Kernel
- Root file system
- Firmware
- 啟動與測試
- References

實驗目的

- 實驗目的：
  - 於Raspberry Pi3上安裝Linux作業系統。編譯Bootloader、Kernel Image與Root file system，將其存入儲存裝置中，將Raspberry Pi 3開機。



Bootloader

Kernel

Root file system

Firmware

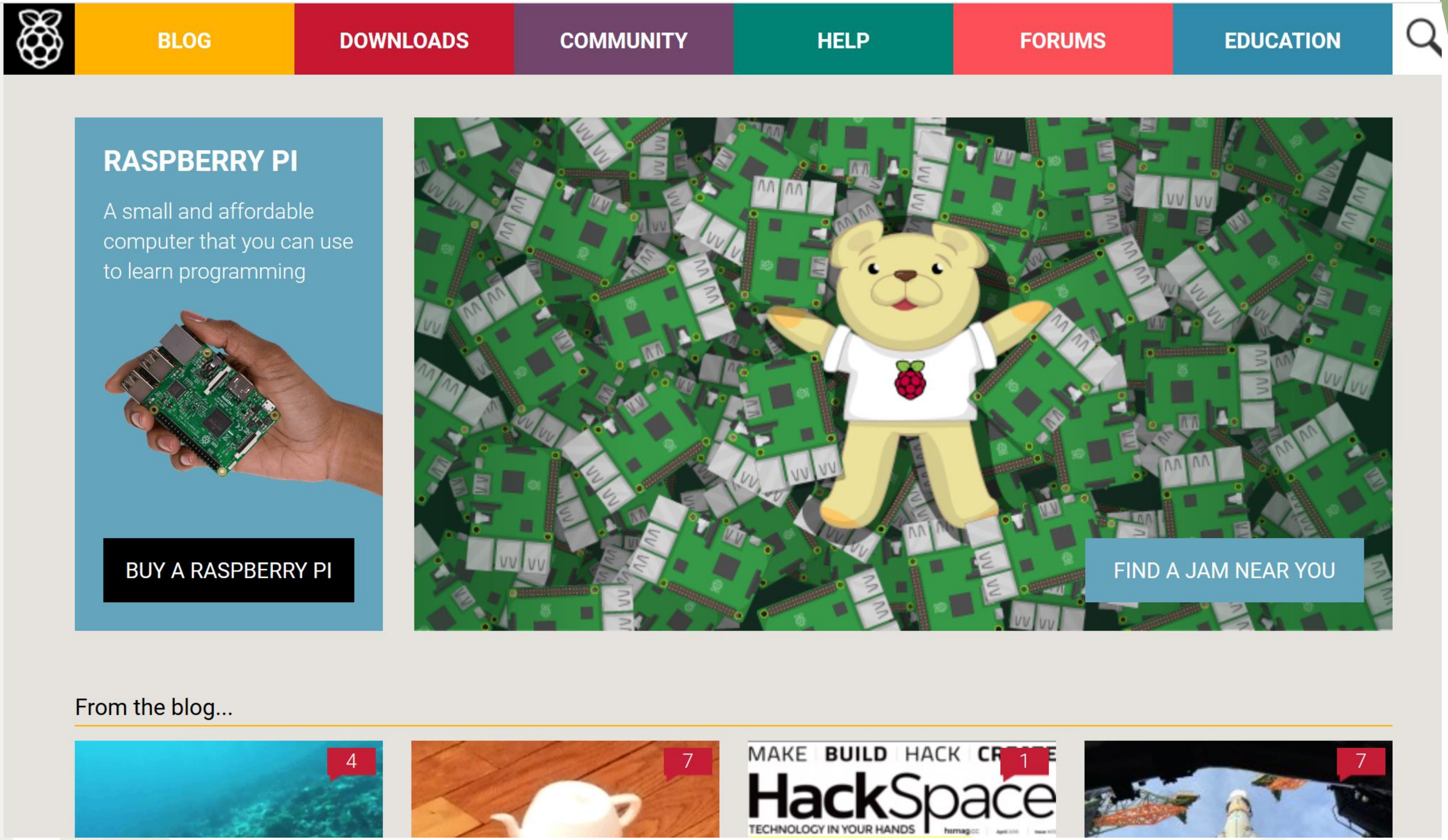
軟硬體與環境需求

- 環境
  - Ubuntu 12.04 LTS
- Software
  - buildroot
- 硬體
  - Raspberry Pi 3 model B
  - 8G SDcard

Raspberry Pi 3介紹

- 一款基於Linux的單板電腦，由英國的Raspberry Pi Foundation所開發。
- 規格：
  - SoC：Broadcom BCM2837（CPU，GPU DSP和SDRAM、USB）
  - CPU：ARM Cortex-A53 64位元 1.2GHz
  - 記憶體：1024 MB (LPDDR2)
  - 網路介面：10/100Mbps 乙太網介面（RJ45介面），支援802.11n無線網路及藍牙4.1
  - 電源輸入：5V (通過MicroUSB或經GPIO輸入)

www.raspberrypi.org



RASPBERRY PI

A small and affordable computer that you can use to learn programming

From the blog...

Boot Sequence

When the Pi is powered up,

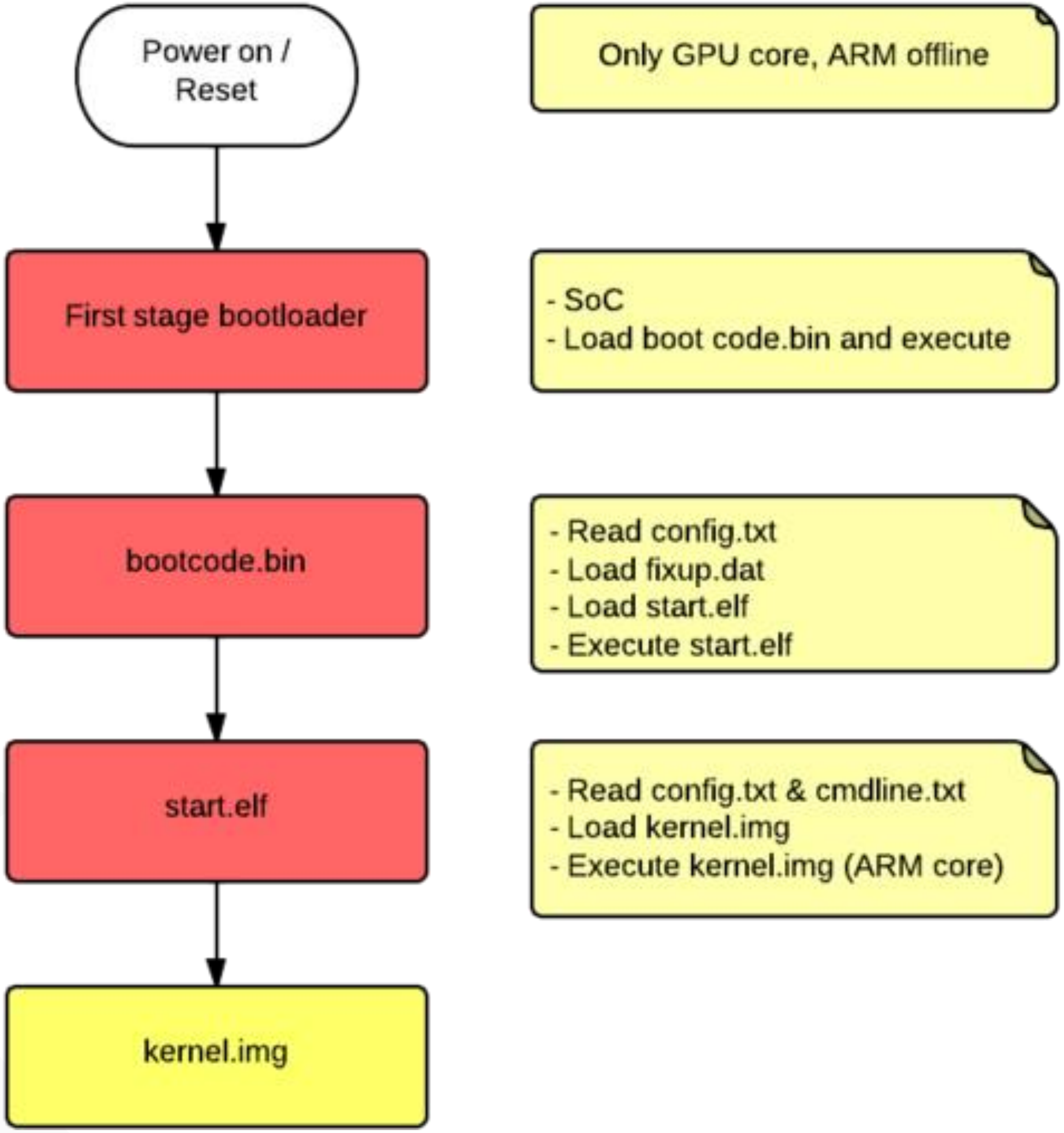
- The ARM processor remains off, SDRAM is disabled, and the GPU core is the one that starts the booting procedure.
- The GPU starts executing the first stage bootloader which is stored in the ROM on the SoC (not on the SD Card).
  - This firmware mainly enables the GPU to access the SD Card, and read a FAT32 Partition on it.

Reference from Ahmed El-Arabawy’s slide

Boot Sequence

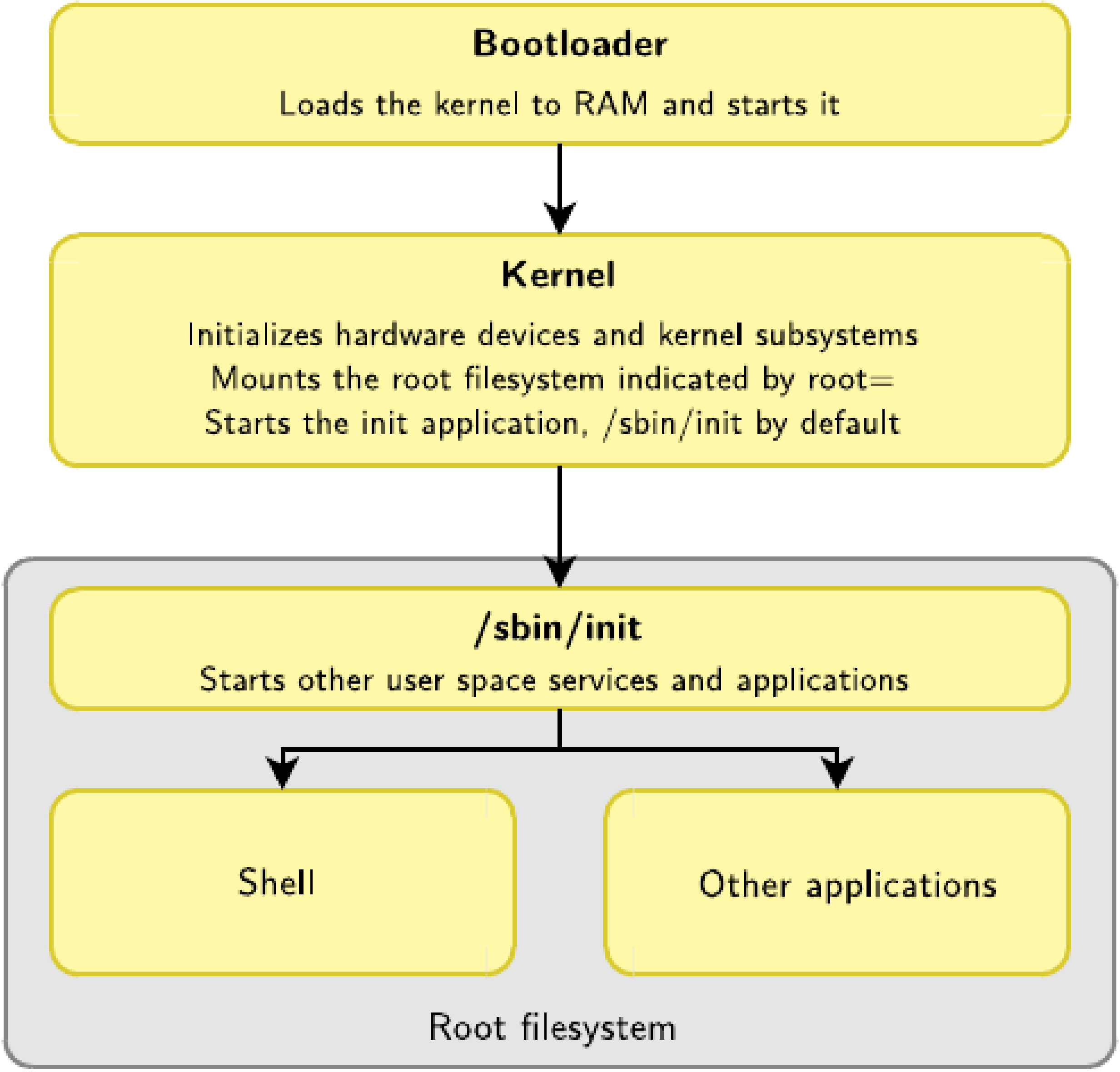
- The first file read from the SD Card is the second stage bootloader (bootcode.bin). The GPU reads this file and executes it.
- The job of this stage of the bootloader is to enable the SDRAM and it loads and runs start.elf.
- The last bootloader stage start.elf reads the kernel image (kernel.img), configuration file(config.txt), and kernel command line parameters (cmdline.txt), then it loads them in memory, and wakes up the ARM core

Reference from Ahmed El-Arabawy’s slide



Reference from https://github.com/gogojesse/gos/wiki/Raspberry-Pi

簡易Linux系統架構



Bootloader

Loads the kernel to RAM and starts it

Kernel

Initializes hardware devices and kernel subsystems
Mounts the root filesystem indicated by root=
Starts the init application, /sbin/init by default

/sbin/init

Starts other user space services and applications

Shell

Other applications

Root filesystem

Buildroot介紹

- Target：
  - Making Embedded Linux Easy
- 基本概念：
  - 建立一個自用的cross-compilation toolchain， 再經由此cross-compiler編譯Kernel與Root file system
- 主要設計理念：
  - Simple to use
  - Simple to customize
  - Reproducible builds
  - Small root file system
  - Relatively fast boot
  - Easy to understand
- 官網：https://buildroot.org/

實驗流程

- 使用Buildroot內的source code編譯出：
  - Bootloader
  - Kernel
  - Root file system
- 切割並格式化SD card，將Bootloader、Kernel Image與Root file system 放入儲存裝置中
- 啟動與測試

Bootloader & Kernel

- make raspberrypi3_defconfig
- output/images
  - bcm2710-rpi-3-b.dtb
  - zImage

Root file system

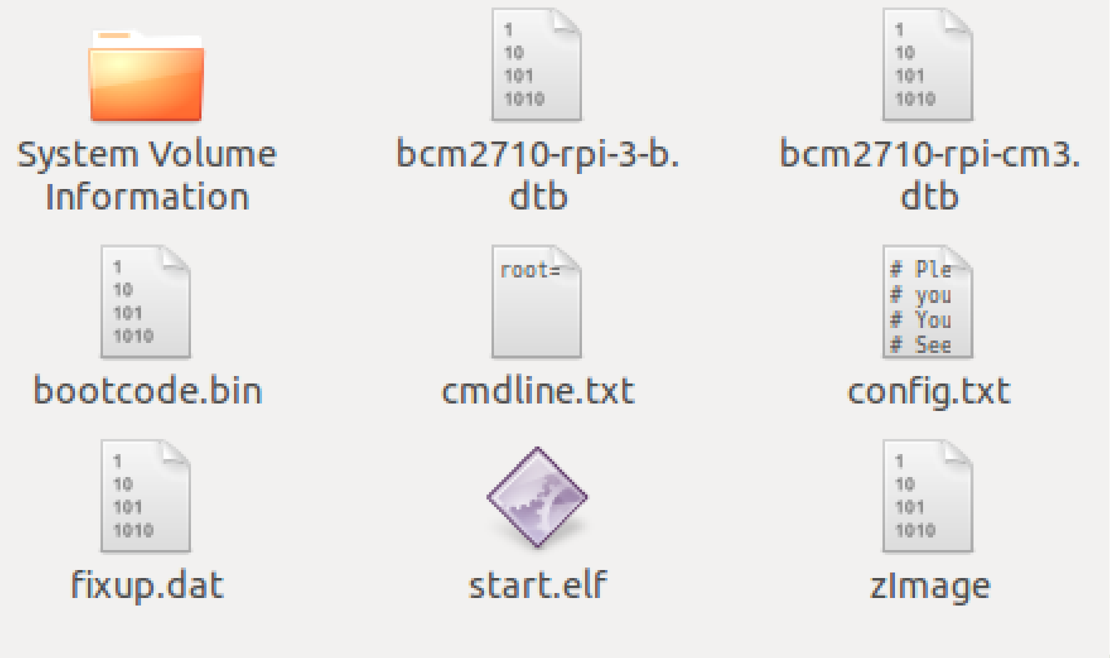
- make menuconfig
  - Filesystem images → tar the root file system → Compression method
  - Bootloaders → U-boot (Ubuntu 14.x ↑**UP**)
- output/images
  - rootfs.tar.gz or .bz2

Firmware

- output/rpi-firmware/
  - bootcode.bin
  - cmdline.txt
  - cofing.txt
  - fixup.dat
  - start.elf

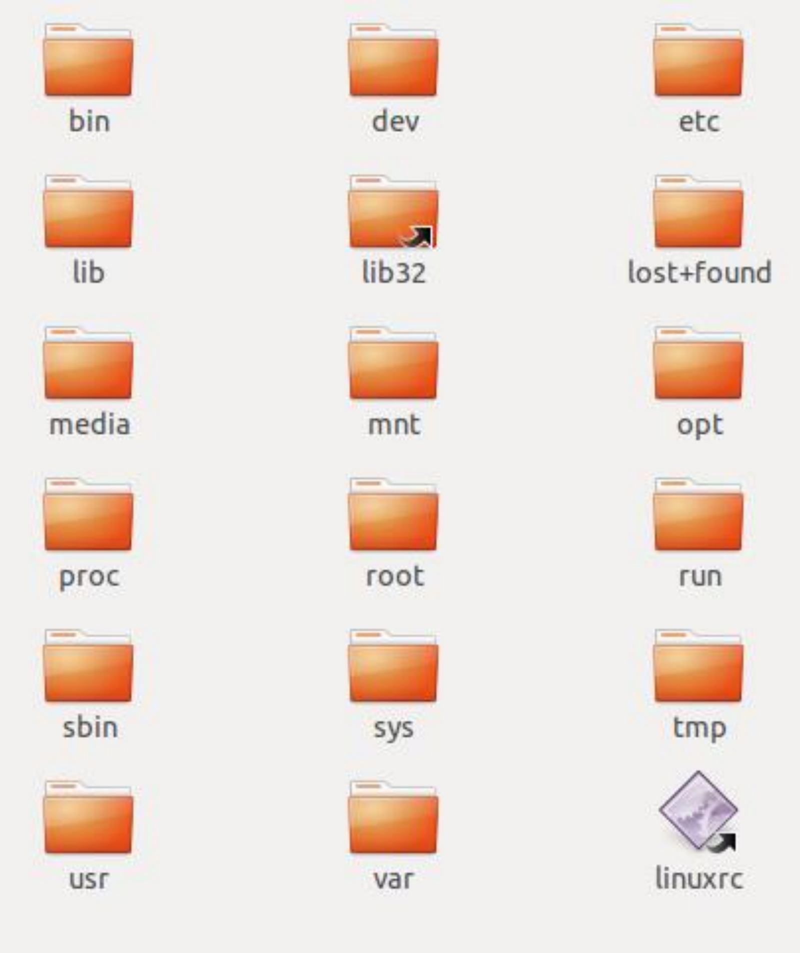
SD card(1/2)

- 掛載buildroot產生出的sdcard.img來切割SD card
  - sudo dd=sdcard.img of=/dev/sdx
- 將bootloader、kernel image與firmware放入BOOT

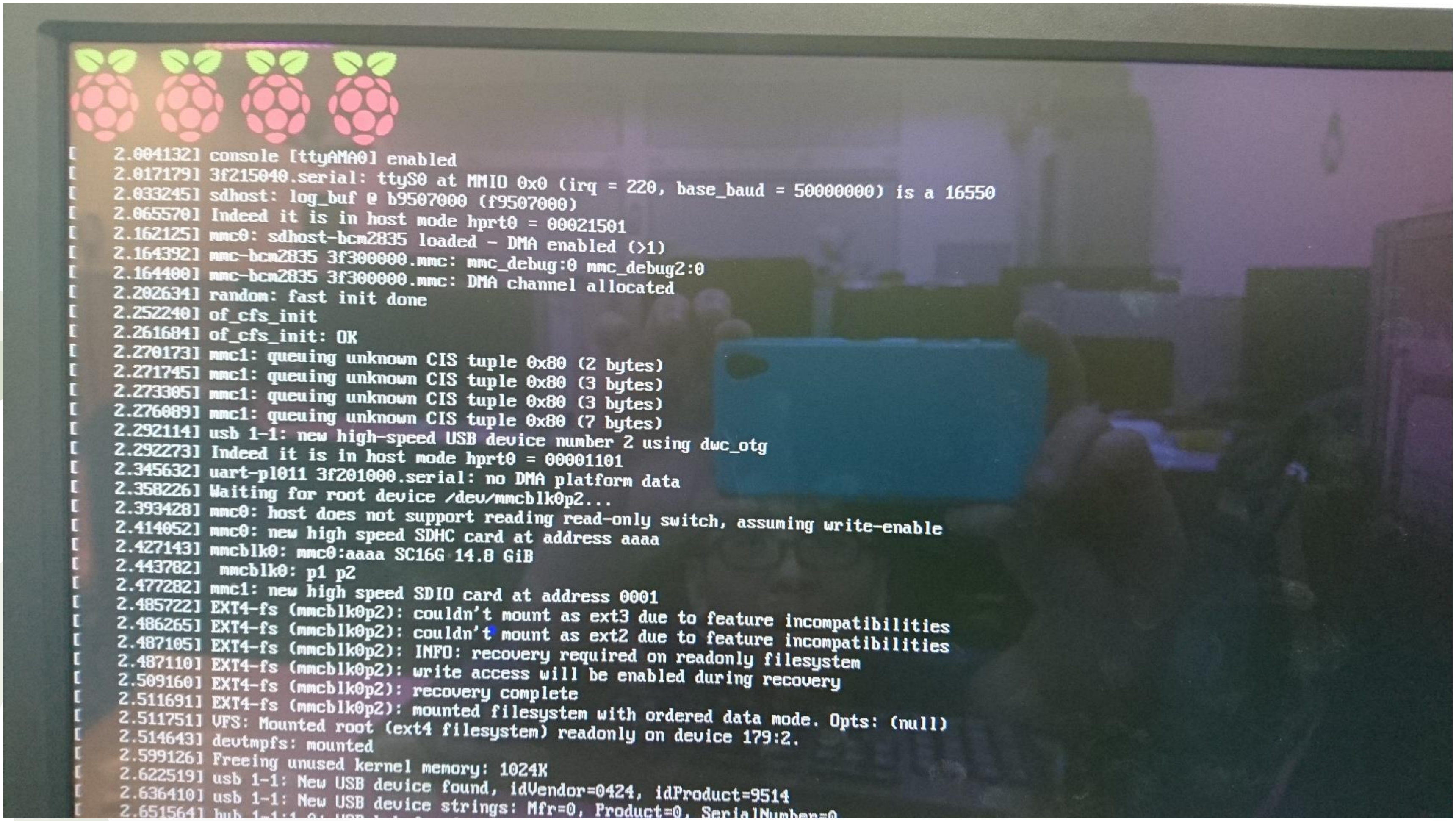


SD card(2/2)

- 將root file system於FILESYSTEM下解壓縮

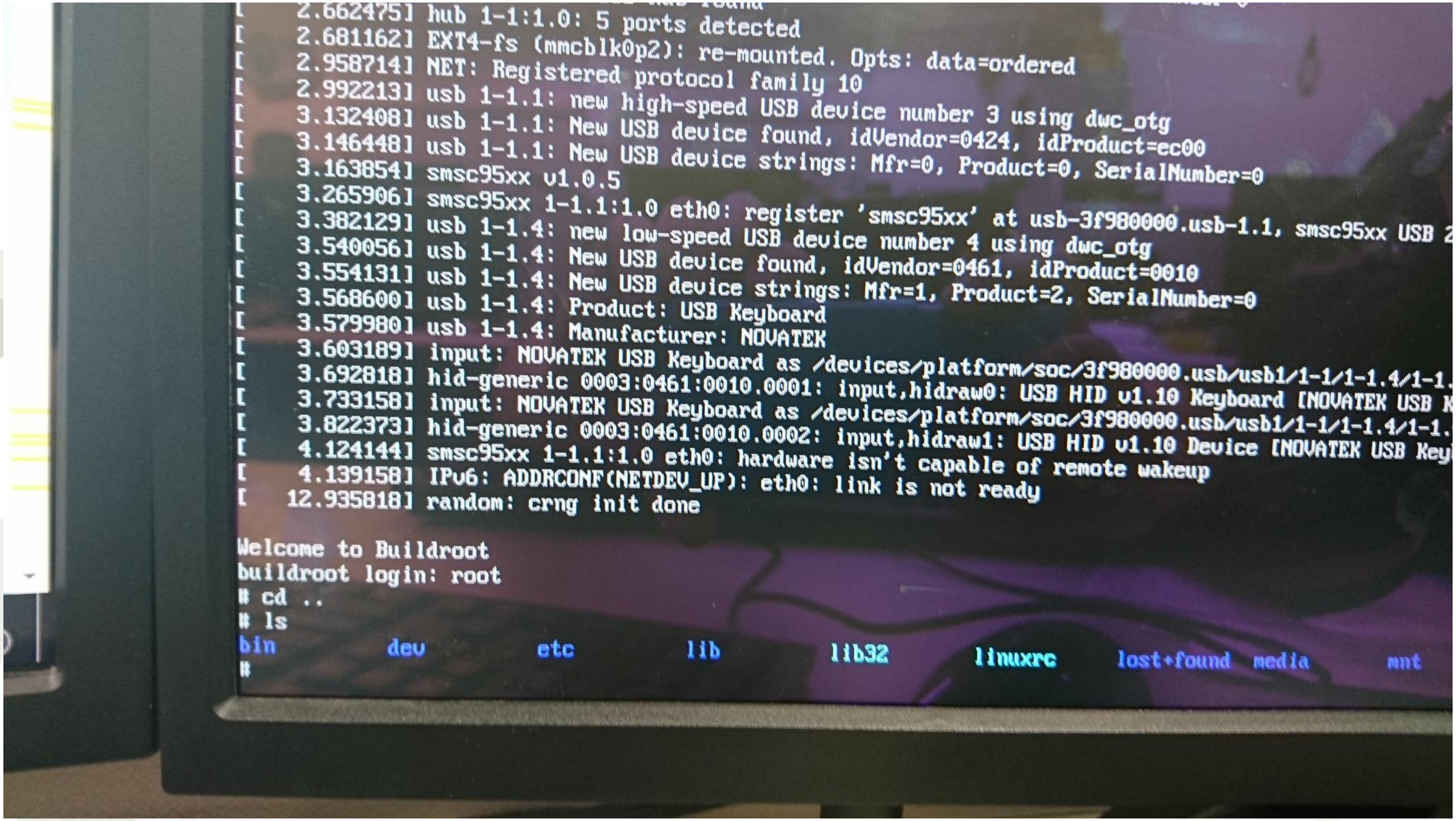


開機與測試(1/2)



```
[    2.004132] console [ttyAMA0] enabled
[    2.017179] 3f215040.serial: ttyS0 at MMIO 0x0 (irq = 220, base_baud = 50000000) is a 16550
[    2.032245] sdhost: log_buf @ b9507000 (f9507000)
[    2.065570] Indeed it is in host mode hprt0 = 00021501
[    2.162125] mmc0: sdhost-bcm2835 loaded - DMA enabled (>1)
[    2.164392] mmc-bcm2835 3f300000.mmc: mmc_debug:0 mmc_debug2:0
[    2.164400] mmc-bcm2835 3f300000.mmc: DMA channel allocated
[    2.202634] random: fast init done
[    2.252240] of_cfs_init
[    2.261684] of_cfs_init: OK
[    2.270173] mmc1: queuing unknown CIS tuple 0x80 (2 bytes)
[    2.271745] mmc1: queuing unknown CIS tuple 0x80 (3 bytes)
[    2.273305] mmc1: queuing unknown CIS tuple 0x80 (3 bytes)
[    2.276089] mmc1: queuing unknown CIS tuple 0x80 (7 bytes)
[    2.292114] usb 1-1: new high-speed USB device number 2 using dwc_otg
[    2.292273] Indeed it is in host mode hprt0 = 00001101
[    2.345632] uart-pl011 3f201000.serial: no DMA platform data
[    2.358226] Waiting for root device /dev/mmcblk0p2...
[    2.393428] mmc0: host does not support reading read-only switch, assuming write-enable
[    2.414052] mmc0: new high speed SDHC card at address aaaa
[    2.427143] mmcblk0: mmc0:aaaa SC16G 14.8 GiB
[    2.443782]  mmcblk0: p1 p2
[    2.477282] mmc1: new high speed SDIO card at address 0001
[    2.485722] EXT4-fs (mmcblk0p2): couldn't mount as ext3 due to feature incompatibilities
[    2.486265] EXT4-fs (mmcblk0p2): couldn't mount as ext2 due to feature incompatibilities
[    2.487105] EXT4-fs (mmcblk0p2): INFO: recovery required on readonly filesystem
[    2.487110] EXT4-fs (mmcblk0p2): write access will be enabled during recovery
[    2.509160] EXT4-fs (mmcblk0p2): recovery complete
[    2.511691] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
[    2.511751] VFS: Mounted root (ext4 filesystem) readonly on device 179:2.
[    2.514643] devtmpfs: mounted
[    2.599126] Freeing unused kernel memory: 1024K
[    2.622519] usb 1-1: New USB device found, idVendor=0424, idProduct=9514
[    2.636410] usb 1-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[    2.651564] hub 1-1:1.0: USB
```

開機與測試(2/2)



```
[    2.662475] hub 1-1:1.0: 5 ports detected
[    2.681162] EXT4-fs (mmcblk0p2): re-mounted. Opts: data=ordered
[    2.958714] NET: Registered protocol family 10
[    2.992213] usb 1-1.1: new high-speed USB device number 3 using dwc_otg
[    3.132408] usb 1-1.1: New USB device found, idVendor=0424, idProduct=ec00
[    3.146448] usb 1-1.1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[    3.163854] smsc95xx v1.0.5
[    3.265906] smsc95xx 1-1.1:1.0 eth0: register 'smsc95xx' at usb-3f980000.usb-1.1, smsc95xx USB 2
[    3.382129] usb 1-1.4: new low-speed USB device number 4 using dwc_otg
[    3.540056] usb 1-1.4: New USB device found, idVendor=0461, idProduct=0010
[    3.554131] usb 1-1.4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[    3.568600] usb 1-1.4: Product: USB Keyboard
[    3.579980] usb 1-1.4: Manufacturer: NOVATEK
[    3.603189] input: NOVATEK USB Keyboard as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.4/1-1.
[    3.692818] hid-generic 0003:0461:0010.0001: input,hidraw0: USB HID v1.10 Keyboard [NOVATEK USB K
[    3.733158] input: NOVATEK USB Keyboard as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.4/1-1.
[    3.822373] hid-generic 0003:0461:0010.0002: input,hidraw1: USB HID v1.10 Device [NOVATEK USB Key
[    4.124144] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
[    4.139158] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[   12.935818] random: crng init done

Welcome to Buildroot
buildroot login: root
# cd ..
# ls
bin        dev        etc        lib        lib32      linuxrc    lost+found media      mnt
#
```

References

- http://fichugh.blogspot.tw/2016/02/buildroot-study.html
- http://www.embeddedforu.com/embedded-linux/raspberry-pi/embedded-linux-development-on-raspberry-pi-using-buildroot-part1/
- http://www.embeddedforu.com/embedded-linux/raspberry-pi/embedded-linux-development-on-raspberry-pi-using-buildroot-part2-2/

DEMO Target

- 確實使用Buildroot建構出最簡易Linux系統，並成功開機且使用root權限進入根目錄

（註：請確實確定能完全成功地開機，而不是進入initramfs）

Q & A (請於實驗報告裡回覆)

- 請問下面firmware目錄下的各個檔案的用途分別是什麼?
  - output/rpi-firmware/
    - bootcode.bin
    - cmdline.txt
    - cofing.txt
    - fixup.dat
    - start.elf
- "cmdline.txt"的內容是什麼，相關參數的意義是什麼?