

## ภาคผนวก I

# การทดลองที่ 9 การศึกษาและปรับแก้อินพุตและเอาต์พุตต่างๆ

การทดลองในภาคผนวกนี้ จะช่วยอธิบายเนื้อหาในบทที่ 6 ซึ่งเกี่ยวข้องกับอุปกรณ์อินพุต/เอาต์พุตที่หลากหลายบนเครื่องคอมพิวเตอร์ตั้งโต๊ะ โดยมีวัตถุประสงค์เหล่านี้

- เพื่อให้เข้าใจการปรับแก้อุปกรณ์อินพุตและเอาต์พุตชนิดต่างๆ บนระบบปฏิบัติการ Raspbian
- เพื่อให้เข้าใจความแตกต่างระหว่างอุปกรณ์อินพุตและเอาต์พุตชนิดต่างๆ บนบอร์ด Pi3
- เพื่อให้สามารถอ่านข้อความแสดงรายละเอียดของอุปกรณ์อินพุตและเอาต์พุตชนิดต่างๆ

หลักการและพื้นฐานความเข้าใจจะช่วยแนะแนวทางให้อ่านสามารถศึกษาค้นคว้า อินพุต/เอาต์พุตอื่นๆ ในชิพและบนบอร์ดได้เพิ่มเติม รวมไปถึงบนโทรศัพท์เคลื่อนที่ แท็บเล็ตคอมพิวเตอร์ และอุปกรณ์อินเทอร์เน็ตสรรพสิ่ง (Internet of Things)

## I.1 จอแสดงผลผ่านพอร์ต HDMI

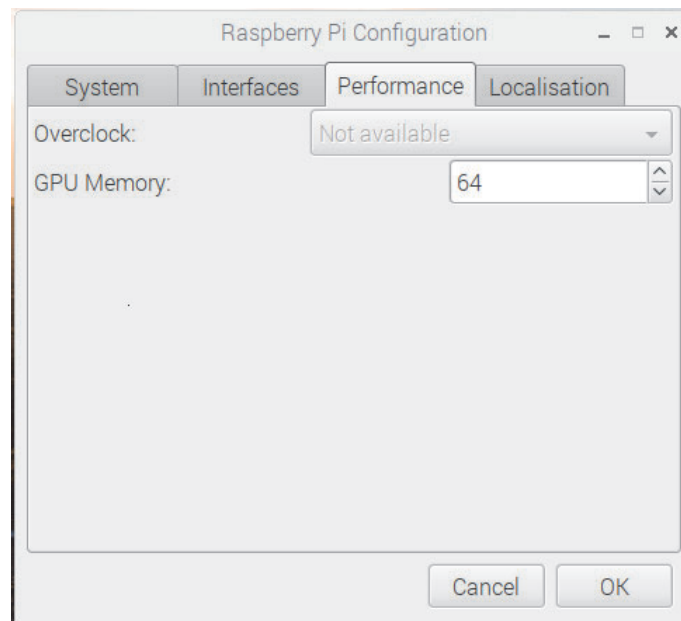
หน่วยความจำสำหรับจอแสดงผลหรือ GPU (Graphic Processing Unit) ถูกแบ่งพื้นที่ออกจาก หน่วยความจำ DRAM บนบอร์ด เพื่อใช้งานร่วมกันทำให้ประหยัดต้นทุน แต่มีข้อเสียในด้านประสิทธิภาพจะลดลง เมื่อผู้ใช้งานต้องการภาพที่มี อัตราการเปลี่ยนแปลง (Refresh Rate) หรืออัตราเฟรมเรต (Frame Rate) สูง เช่น ภาพเคลื่อนไหว เกมส์ 3 มิติ

### I.1.1 การปรับแก้ขนาดหน่วยความจำของ GPU

ความละเอียดของจอแสดงผลขึ้นตรงกับขนาดของหน่วยความจำของ GPU ผู้อ่านสามารถปรับแก้ขนาดหน่วยความจำของ GPU ดังนี้

menu->Preferences->Raspberry Pi Configuration->Set Resolution->Performance

โดยหน้าต่างที่ปรากฏขึ้นมีลักษณะดังนี้ ผู้ใช้สามารถกำหนดขนาดที่ต้องการโดยขั้นต่ำคือ 64 MB เพื่อให้ระบบสามารถแสดงผลได้ หากผู้ใช้กำหนดสูงเกินไป จะทำให้บอร์ดมีหน่วยความจำไม่เพียงพอ

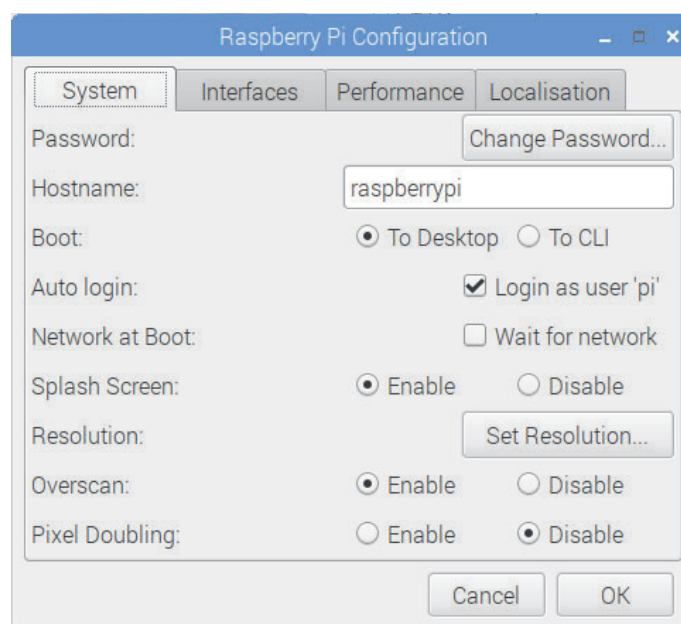


รูปที่ I.1: หน้าต่างกำหนดขนาดหน่วยความจำสำหรับ GPU ที่ 64 เมกะไบต์

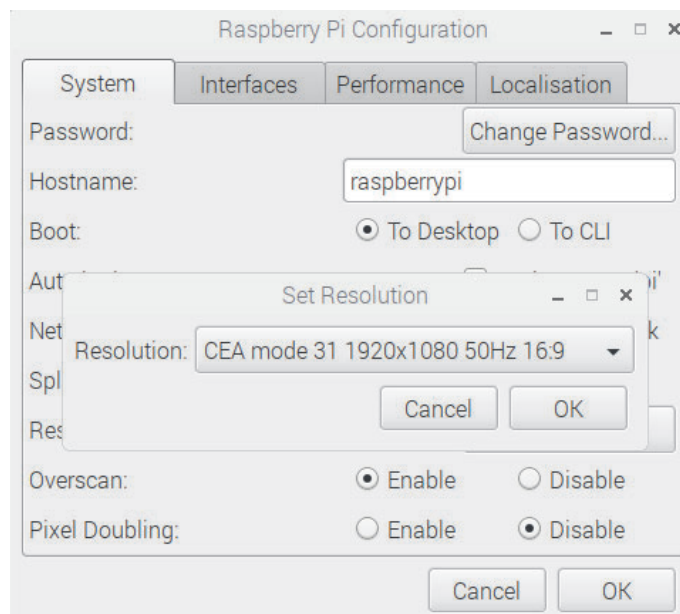
### I.1.2 การปรับแก้ความละเอียดของจอแสดงผล

เมื่อขนาดหน่วยความจำของ GPU มีเพียงพอ ผู้ใช้สามารถปรับเพิ่มหรือลดความละเอียดของจอแสดงผลได้โดยกดปุ่มบนเมนูดังนี้

menu->Preferences->Raspberry Pi Configuration->Set Resolution

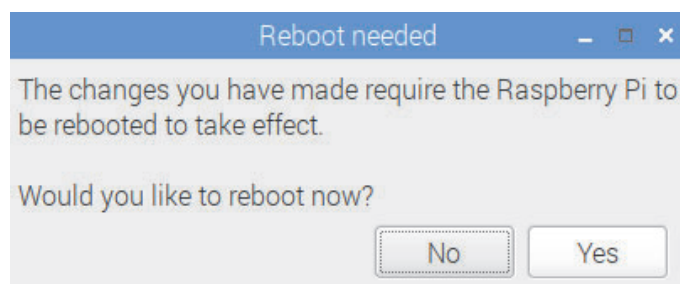


รูปที่ I.2: หน้าต่าง Raspberry Pi Configuration แท็บ System สำหรับกำหนดความละเอียดหน้าจอแสดงผล (Resolution)



รูปที่ I.3: หน้าต่าง Set Resolution สำหรับกำหนดความละเอียดหน้าจอที่ต้องการ

กดปุ่ม Set Resolution สำหรับกำหนดความละเอียดหน้าจอที่ต้องการ ในรูปที่ ผู้เขียนต้องการแสดงผลที่ความละเอียด CEA Mode 31 1920x1080 50Hz 16:9 หลังจากนั้นกดปุ่ม OK หน้าต่าง Reboot needed จะปรากฏขึ้น



รูปที่ I.4: หน้าต่าง Reboot needed กดปุ่ม Yes เมื่อต้องการรีบูท ณ เวลานั้น

## I.2 ระบบเสียงดิจิทัล

อุปกรณ์ระบบเสียงดิจิทัลที่ติดตั้งมาบนบอร์ด Pi3 จากโรงงาน ผู้ใช้สามารถเพิ่มเติมได้ผ่านพอร์ต USB และปรับแต่งระดับเสียงได้เช่นกัน

### I.2.1 รายชื่ออุปกรณ์ในระบบเสียง

ระบบเสียงในระบบปฏิบัติการ Linux ควบคุมการทำงานของเสียงผ่านระบบ ALSA (Advanced Linux Sound Architecture) ซึ่งจัดเตรียมไดรเวอร์ (Device Driver) สำหรับเสียงให้กับเคอร์เนล และอุปกรณ์ที่เกี่ยวข้องกับเสียงผ่านพอร์ต USB เช่น ไมโครโฟน, หูฟังพร้อมไมโครโฟน, เว็บบแคม เป็นต้น ผู้อ่านสามารถแสดงรายชื่อไฟล์หรือไดเรกทอรีที่เกี่ยวข้องกับระบบเสียงดังนี้

```
$ ls -l /proc/asound
```

```
lrwxrwxrwx 1 root root 5 Mar 26 20:59 ALSA -> card0
dr-xr-xr-x 4 root root 0 Mar 26 20:59 card0
-r--r--r-- 1 root root 0 Mar 26 20:59 cards
-r--r--r-- 1 root root 0 Mar 26 20:59 devices
-r--r--r-- 1 root root 0 Mar 26 20:59 modules
-r--r--r-- 1 root root 0 Mar 26 20:59 pcm
dr-xr-xr-x 2 root root 0 Mar 26 20:59 seq
-r--r--r-- 1 root root 0 Mar 26 20:59 timers
-r--r--r-- 1 root root 0 Mar 26 20:59 version
```

```
pi@raspberrypi:~ $ ls -l /proc/asound
total 0
lrwxrwxrwx 1 root root 5 Mar 29 22:54 b1 -> card0
dr-xr-xr-x 4 root root 0 Mar 29 15:23 card0
dr-xr-xr-x 4 root root 0 Mar 29 15:23 card1
-r--r--r-- 1 root root 0 Mar 29 22:54 cards
-r--r--r-- 1 root root 0 Mar 29 22:54 devices
lrwxrwxrwx 1 root root 5 Mar 29 22:54 Headphones -> card1
-r--r--r-- 1 root root 0 Mar 29 22:54 modules
dr-xr-xr-x 4 root root 0 Mar 29 22:54 oss
-r--r--r-- 1 root root 0 Mar 29 22:54 pcm
dr-xr-xr-x 2 root root 0 Mar 29 22:54 seq
-r--r--r-- 1 root root 0 Mar 29 22:54 timers
-r--r--r-- 1 root root 0 Mar 29 22:54 version
```

ผลลัพธ์คือ รายชื่ออุปกรณ์ที่เกี่ยวข้องกับเสียง โดยเฉพาะ ALSA ซึ่งได้แสดงไปก่อนหน้านี้ ผู้อ่านจะสังเกตได้ว่าไดเรกทอรี `/proc/asound/pcm` จะเชื่อมโยงกับเนื้อหาในหัวข้อที่ 6.4 จะสังเกตเห็นว่ามีไดเรกทอรีชื่อ `card0` อยู่สองตำแหน่งคือ ในแถวแรก และแถวที่มีชื่อ `ALSA -> card0` สัญลักษณ์ `->` เรียกว่าซิมบออลลิงค์ (Symbolic Link) หมายความว่า ไดเรกทอรีชื่อ `ALSA` คือไดเรกทอรี `card0`

1. ผู้อ่านสามารถทดสอบโดยพิมพ์คำสั่งต่อไปนี้

```
$ cat /proc/asound/ALSA
```

บันทึกผลลัพธ์ในพื้นที่ว่างต่อไปนี้

```
pi@raspberrypi:~ $ cat /proc/asound/b1
cat: /proc/asound/b1: Is a directory
```

2. ผู้อ่านสามารถทดสอบโดยพิมพ์คำสั่งต่อไปนี้

```
$ cat /proc/asound/card0
```

บันทึกผลลัพธ์ในพื้นที่ว่างต่อไปนี้ และเปรียบเทียบกับผลลัพธ์ก่อนหน้าว่าแตกต่างกันหรือไม่

```
pi@raspberrypi:~ $ cat /proc/asound/card0
cat: /proc/asound/card0: Is a directory
```

ได้ผลลัพธ์เหมือนกันคือ Is a directory

3. ค้นคว้าเพิ่มเติมเพื่อหาความหมายของ Symbolic Link และจดบันทึก

Symbolic Link เป็นการสร้างตัวอ้างอิงจากไฟล์ที่มีอยู่แล้ว ทำให้เมื่อไฟล์ต้นฉบับถูกลบ ข้อมูลในส่วนนั้นก็จะไม่สามารถเข้าถึงได้จาก Link ที่สร้างไว้ได้  
การสร้าง Symbolic Link สามารถสร้างบนระบบไฟล์ที่แตกต่างกันได้

4. พิมพ์คำสั่งนี้ในโปรแกรม Terminal

```
$ cat /proc/asound/cards
```

โดยคำสั่ง cat ซึ่งได้อธิบายแล้วในการทดลองที่ 4 ภาคผนวก D สามารถอ่านไฟล์และแสดงข้อมูลภายในไฟล์ผ่านทางหน้าจอแสดงผล บันทึกในพื้นที่ว่างต่อไปนี้

```
pi@raspberrypi:~ $ cat /proc/asound/cards
0 [b1 ]: bcm2835_hdmi - bcm2835 HDMI 1
bcm2835 HDMI 1
1 [Headphones ]: bcm2835_headphonbcm2835 Headphones - bcm2835 Headphones
bcm2835 Headphones
```

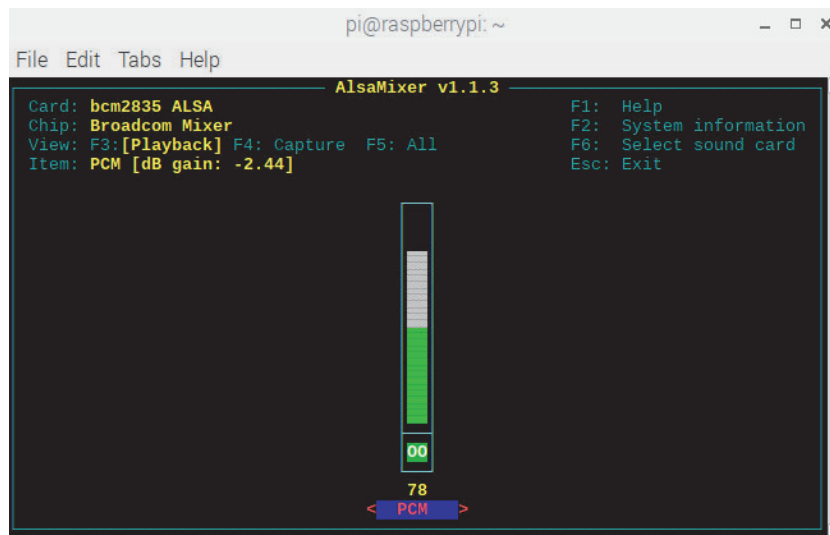
อภิปรายผลที่ได้ ดังนี้ ผลลัพธ์ได้จากบอร์ด Pi3 ใช้ชิพ BCM2835 แต่ยังใช้ไดรเวอร์เสียงเดียวกันกับ BCM2835 โดย หมายเลข 0 คือ หมายเลขของระบบเสียงที่ติดตั้งใช้งานเพียงระบบเดียว และตรงกับอุปกรณ์ ชื่อ \_\_\_\_ 0 [b1 ] : bcm2835 hdmi 1

## 1.2.2 การควบคุมระดับเสียง

ผู้อ่านสามารถควบคุมระดับความดังของเสียงทั้งด้านอินพุตและเอาต์พุต โดยพิมพ์คำสั่งนี้

```
$ alsamixer
```

หน้าต่างต่อไปนี้จะปรากฏขึ้น ผู้อ่านสามารถกดปุ่มลูกศรขึ้น/ ลง เพื่อเพิ่ม/ลด ระดับความดังได้



รูปที่ I.5: โปรแกรม ALSA Mixer สำหรับควบคุมระดับเสียงบนบอร์ด Pi3

หมายเหตุ ผู้อ่านสามารถติดตั้งอุปกรณ์เสียงผ่านพอร์ต USB และใช้คำสั่งเหล่านี้เพื่อตรวจสอบและควบคุมการทำงาน

## I.3 พอร์ตเชื่อมต่ออุปกรณ์ USB

### I.3.1 รายชื่ออุปกรณ์กับพอร์ต USB

1. ในการทดลองนี้ ขอผู้อ่านให้ดึงหัวเชื่อมต่อ USB ของเมาส์ที่ใช้อยู่ออก แล้วพิมพ์คำสั่งนี้ในโปรแกรม Terminal

```
$ lsusb
```

เพื่อแสดงรายชื่ออุปกรณ์ USB ที่เชื่อมต่ออยู่ทั้งหมดในบอร์ด ดังตัวอย่างต่อไปนี้

```
Bus 001 Device 005: ID 413c:2003 Dell Computer Corp. Keyboard
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
                    SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

ผู้อ่านจะเห็นรายชื่ออุปกรณ์ที่เชื่อมต่อกับพอร์ต USB เรียงลำดับย้อนกลับ จาก Device 005 - Device 001 โดย

- Device 005 คือ คีย์บอร์ดมีหมายเลข ID = 413c:2003 ผลิตโดย บริษัท Dell Computer Corp. ซึ่งคีย์บอร์ดของผู้อ่านอาจจะแตกต่าง
- Device 003 คือ วงจร Ethernet สำหรับเชื่อมต่อเครือข่ายชนิดสาย มีหมายเลข ID = 0424:ec00 ผลิตโดย บริษัท Standard Microsystems Corp. รุ่น SMSC9512/9514

- Device 002 คือ วงจร USB Hub สำหรับเชื่อมต่อพอร์ต USB เพิ่มเติม มีหมายเลข ID = 0424:9514 ผลิตโดย บริษัท Standard Microsystems Corp. รุ่น SMSC9514
- Device 001 คือ วงจร Root Hub เป็นวงจรภายในชิพ BCM2837 สำหรับเชื่อมต่อพอร์ต USB เพิ่มเติม มีหมายเลข ID = 1d6b:0002

## 2. บันทึกผลลัพธ์ของผู้อ่าน

```
pi@raspberrypi:~$ lsusb
Bus 001 Device 005: ID 04d9:a0cd Holtek Semiconductor, Inc.
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Bus 001 Device 005 : ID = 0 4 d 9: a 0 c d

Bus 001 Device 003 : ID = 0 4 2 4: e c 0 0

Bus 001 Device 002 : ID = 0 4 2 4: 9 5 1 4

Bus 001 Device 001 : ID = 1 d 6 b: 0 0 0 2

## 3. ผู้อ่านเสียบเมาส์กลับเข้าไปที่พอร์ต USB ใหม่อีกครั้ง แล้วแสดงรายชื่ออุปกรณ์ USB ด้วยคำสั่ง

```
$ lsusb
Bus 001 Device 005: ID 04d9:a0cd Holtek Semiconductor, Inc.
Bus 001 Device 006: ID 046d:c077 Logitech, Inc. M105 Optical Mouse
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

เช่นเดิม บันทึกผลลงในพื้นที่จัดเตรียมไว้ให้ โปรดสังเกตการเปลี่ยนแปลง

Bus 001 Device 005 : ID = 0 4 d 9: a 0 c d

Bus 001 Device 006 : ID = 0 4 6 d: c 0 7 7

Bus 001 Device 003 : ID = 0 4 2 4: e c 0 0

Bus 001 Device 002 : ID = 0 4 2 4: 9 5 1 4

Bus 001 Device 001 : ID = 1 d 6 b 0 0 0 2

## 4. รายการที่เพิ่มขึ้น คือ

Device 006 : ID = 0 4 6 d: c 0 7 7

คือ เมาส์หมายเลขลำดับที่ Device 006 ที่เพิ่งถูกเสียบกลับเข้าไปยังบอร์ด



### I.3.2 รายละเอียดการเชื่อมต่ออุปกรณ์กับพอร์ท USB

คำสั่งต่อไป คือ **dmesg** สามารถแสดงรายการทำงาน หรือ Log ของระบบปฏิบัติการว่าตั้งแต่เริ่มเปิดเครื่อง โดยคำว่า **dmesg** ย่อมาจากคำสั่ง “display message or display driver” ซึ่งเคอร์เนลได้บันทึกไว้ใน บัฟเฟอร์ชนิดวงแหวน (Ring Buffer) ซึ่งข้อความตอนต้นจะถูกเขียนทับเมื่อบัฟเฟอร์เต็ม

```
$ dmesg
```

```
[0.000000] Booting Linux on physical CPU 0x0
[0.000000] Linux version 4.14.71-v7+ (dc4@dc4-XPS13-9333)
          (gcc version 4.9.3 (crosstool-NG crosstool-ng-1.22.0-88-g8460611))
          #1145 SMP Fri Sep 21 15:38:35 BST 2018
[0.000000] CPU: ARMv7 Processor [410fd034] revision 4 (ARMv7), cr=10c5383d
[0.000000] CPU: div instructions available: patching division code
[0.000000] CPU: PIPT / VIPT nonaliasing data cache,
          VIPT aliasing instruction cache
[0.000000] OF: fdt: Machine model: Raspberry Pi 3 Model B Rev 1.2
[0.000000] Memory policy: Data cache writealloc
[0.000000] cma: Reserved 8 MiB at 0x3ac00000
[0.000000] On node 0 totalpages: 242688
...
[0.000000] Memory: 940232K/970752K available (7168K kernel code, 576K rwddata,
          2076K rodata, 1024K init, 698K bss, 22328K reserved,
          8192K cma-reserved)
[0.000000] Virtual kernel memory layout:
          vector   : 0xfffff000 - 0xfffff1000   (  4 kB)
          fixmap   : 0xfffc0000 - 0xffff00000   (3072 kB)
          vmalloc   : 0xbb800000 - 0xff800000    (1088 MB)
          lowmem    : 0x80000000 - 0xbb400000    ( 948 MB)
          modules   : 0x7f000000 - 0x80000000    (  16 MB)
                   .bss : 0x80c97f10 - 0x80d468b0   ( 699 kB)
                   .data : 0x80c00000 - 0x80c9017c   ( 577 kB)
                   .init : 0x80b00000 - 0x80c00000   (1024 kB)
                   .text : 0x80008000 - 0x80800000   (8160 kB)
...
```

ผู้เขียนสามารถอธิบายผลลัพธ์ได้ดังต่อไปนี้ โดยเรียงลำดับตามเหตุการณ์ ซึ่งมีสัญลักษณ์ [xxxx.yyyyyy] แสดงลำดับที่เกิดขึ้นตามเวลา โดย xxxx คือเลขวินาทีตั้งแต่เคอร์เนลเริ่มทำงาน และ yyyyyy คือเศษวินาที ข้อความที่แสดงเป็น 0.000000 เนื่องจากเคอร์เนลอยู่ระหว่างการเริ่มต้น

- เริ่มต้นการบูทระบบปฏิบัติการด้วยซีพียูคอร์หมายเลข 0
- แสดงรายละเอียดหมายเลขเวอร์ชันของลินุกซ์



- แสดงรายละเอียดของ CPU ซึ่งใช้คำสั่งภาษาแอสเซมบลีเวอร์ชัน 7 (ARMv7)
- แสดงผลการตรวจจับว่าซีพียูรองรับคำสั่ง DIV
- รายงานว่า แคชข้อมูล ทำงานแบบ nonaliasing PIPT (Physically Indexed, Physically Tagged) หรือ VIPT (Virtually Indexed, Physically Tagged) อย่างใดอย่างหนึ่ง และแคชคำสั่ง ทำงานแบบ aliasing VIPT แคชข้อมูลไม่สามารถแชร์ข้ามโปรเซสได้เนื่องจากทำงานแบบ nonaliasing ในขณะที่แคชคำสั่งสามารถแชร์ข้ามโปรเซสได้ เนื่องจากทำงานแบบ aliasing [ข้อมูลเพิ่มเติม](#)
- แสดงผลการตรวจจับว่าเป็นบอร์ด Raspberry Pi 3 Model B Rev 1.2
- การทำงานของแคชข้อมูลเป็นชนิด writealloc ย่อมาจาก Write Allocation ซึ่งซีพียูจะเขียนข้อมูลทั้งในแคชก่อน เมื่อแคชจะต้องถูกย้ายออกจึงค่อยเขียนในหน่วยความจำหลักภายหลัง [ข้อมูลเพิ่มเติม](#)
- cma (Contiguous Memory Allocator) สำหรับขบวนการ DMA เริ่มต้นที่แอดเดรส 0x3AC00000 ขนาด 8 เมกะไบต์
- ...
- พื้นที่การจัดวางหน่วยความจำเสมือนของเคอร์เนล (Virtual kernel memory layout) ผู้เขียนได้ทำการจัดเรียงใหม่ตามหมายเลขแอดเดรสที่ตำแหน่งมาก ไกลลงมาจนถึงหมายเลขน้อย เพื่อให้ผู้อ่านมองเห็นภาพและเข้าใจง่ายขึ้น โดยแบ่งเป็นพื้นที่ที่สำคัญๆ ตามลำดับดังนี้
  - จัดเก็บเวกเตอร์สำหรับการขัดจังหวะ (Interrupt Vector) ขนาด 4 กิโลไบต์ จากหมายเลข 0xFFFF\_0000 ถึง 0xFFFF\_1000
  - พื้นที่สำหรับจองหน่วยความจำเสมือน (vmalloc) ขนาด 1088 เมกะไบต์ จากหมายเลข 0xBB80\_0000 ถึง 0xFF80\_0000
  - bss เช็กเมนต์ (.bss) ขนาด 699 กิโลไบต์ จากหมายเลข 0x80C9\_7F10 ถึง 0x80D4\_68B0
  - ดาตาเช็กเมนต์ (.data) ขนาด 577 กิโลไบต์ จากหมายเลข 0x80C0\_0000 ถึง 0x80C9\_017C
  - init เช็กเมนต์ (.init) ขนาด 1024 กิโลไบต์ จากหมายเลข 0x80B0\_8000 ถึง 0x80C0\_0000
  - เท็กซ์เช็กเมนต์ (.text) ขนาด 8160 กิโลไบต์ จากหมายเลข 0x8000\_8000 ถึง 0x8080\_0000

ในการทดลองนี้ ระบบสามารถตรวจจับอุปกรณ์ USB และติดตั้งไดรเวอร์ได้อย่างถูกต้องปราศจากข้อผิดพลาด

1. ผู้อ่านสามารถล้างบัฟเฟอร์โดยใช้คำสั่ง ต่อไปนี้

```
$ sudo dmesg -C
```

โดย -C คือ Clear เป็นคำสั่งเพิ่มเติมให้ dmesg ล้างข้อความในบัฟเฟอร์ออก โปรดสังเกต ตัว C ใหญ่ หลังจากนั้น ผู้อ่านทดสอบโดยการถอดเมาส์ออก แล้วเสียบกลับเข้าไปใหม่

2. ผู้อ่านจะต้องถอดและเสียบเมาส์กลับเข้าไปใหม่อีกรอบ
3. ผู้อ่านสามารถแสดงข้อความที่เพิ่มเข้ามาในบัฟเฟอร์ได้อีก โดยเรียกคำสั่ง

```
$ sudo dmesg
```

#### 4. จดบันทึก

```
pi@raspberrypi:~$ sudo dmesg
[ 1002.368984] usb 1-1.2: USB disconnect, device number 6
[ 1003.420265] usb 1-1.2: new low-speed USB device number 7 using dwc_otg
[ 1003.563238] usb 1-1.2: New USB device found, idVendor=046d, idProduct=c077, bcdDevice=72.00
[ 1003.563263] usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 1003.563270] usb 1-1.2: Product: USB Optical Mouse
[ 1003.563293] usb 1-1.2: Manufacturer: Logitech
[ 1003.569212] input: Logitech USB Optical Mouse as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.2/1-1.2:1.0/0003:046D:C077.0006/input/input7
[ 1003.569880] hid-generic 0003:046D:C077.0006: input,hidraw0: USB HID v1.11 Mouse [Logitech USB Optical Mouse] on usb-3f980000.usb-1.2/input0
```

#### 5. อภิปรายผลลัพธ์ที่บันทึกได้ในพื้นที่ว่างต่อไปนี้

- ตัดการเชื่อมต่อ USB อุปกรณ์หมายเลข 6
- low-speed USB ใหม่ อุปกรณ์หมายเลข 7 ใช้ dwc\_otg
- พบ usb ใหม่ idVendor = 046d, idProduct = c077, bcdDevice = 72.00
- อุปกรณ์ใหม่ : Mfr=1, Product=2, SerialNumber=0
- ผลิตภัณฑ์ : USB optical Logitech
- ผู้ผลิต : Logitech

ในการเชื่อมต่อพอร์ต USB หากระบบแจ้งชื่ออุปกรณ์โดยไม่มีข้อความผิดพลาด แต่อุปกรณ์นั้นยังไม่สามารถทำงานได้ แสดงว่าอุปกรณ์ขาดซอฟต์แวร์ซึ่งทำหน้าที่เป็นไดรเวอร์ ขอให้ผู้อ่านค้นหาจากหมายเลขประจำตัวของผู้ผลิต (idVendor) หากผู้ผลิตมิได้เปิดเผยซอฟต์แวร์ ผู้อ่านจำเป็นต้องดาวน์โหลดหรือคอมไพล์เองจากนักพัฒนารายอื่นแทน

## I.4 พอร์ตเชื่อมต่อเครือข่าย WiFi และ Ethernet

### I.4.1 รายชื่ออุปกรณ์เครือข่าย

1. ผู้อ่านสามารถตรวจสอบรายชื่ออุปกรณ์สำหรับเชื่อมต่อเครือข่ายได้จากคำสั่ง **ifconfig** ทางโปรแกรม Terminal ตัวอย่างผลลัพธ์เป็นดังนี้

```
$ ifconfig
```

2. เติมข้อมูลหรือตัวเลขในช่องว่าง \_ ที่เตรียมไว้ให้จากผลลัพธ์ที่ได้ต่อไปนี้ ซึ่งลำดับรายการอาจแตกต่างกัน

```
eth0: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500 _
    inet _____._____._____._____
    netmask _____._____._____._____
    broadcast _____._____._____._____
    ...
```

ไม่มี เนื่องจากเชื่อมต่อผ่าน WiFi

```

ether b8:27:eb:be:78:e0 txqueuelen 1000 (Ethernet)

lo: flags=73<UP, LOOPBACK, RUNNING> mtu 65536
    inet 127.0.0.1
    netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)

wlan0: flags=4099<UP, BROADCAST, MULTICAST> mtu 1500
    inet 192.168.0.10
    netmask 255.255.255.0
    broadcast 192.168.0.255
% ether b8:27:eb:eb:2d:b5

```

3. โปรดสังเกตคำเริ่มต้นในแต่ละรายการ คั่นคว่ำ และกรอกรายละเอียดเพิ่มเติม ดังนี้

- eth0 หมายถึง ชื่อที่ใช้ในการเชื่อมต่อผ่านสาย LAN
- lo0 หมายถึง **loopback**
- wlan0 หมายถึง ชื่อที่ใช้ในการเชื่อมต่อผ่าน WiFi

## I.4.2 การเปิด/ปิดอุปกรณ์เครือข่าย

1. ผู้อ่านสามารถเปิดอุปกรณ์ eth0 ได้ตามต้องการแล้วทำการตรวจสอบ ดังนี้

```
$ sudo ifconfig eth0 down
$ ifconfig
```

จกว่าข้อความใดที่บ่งบอกว่า eth0 ไม่ทำงานแล้ว **ไม่แสดง eth0**

```

pi@raspberrypi:~$ ifconfig
lo: flags=73<UP, LOOPBACK, RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 5 bytes 284 (284.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5 bytes 284 (284.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
    inet 192.168.0.10 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::b27:ebff:feb2:b5 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:eb:2d:b5 txqueuelen 1000 (Ethernet)
    RX packets 13905 bytes 1365999 (1.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14097 bytes 8065419 (7.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

2. ผู้อ่านสามารถเปิดอุปกรณ์ eth0 ได้ตามต้องการแล้วทำการตรวจสอบ ดังนี้

```
$ sudo ifconfig eth0 up
$ ifconfig
```

จกว่าข้อความใดที่บ่งบอกว่า eth0 ทำงานแล้ว **แสดง eth0**

```

pi@raspberrypi:~$ ifconfig
eth0: flags=4099<UP, BROADCAST, MULTICAST> mtu 1500
    ether b8:27:eb:be:78:e0 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP, LOOPBACK, RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 5 bytes 284 (284.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5 bytes 284 (284.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
    inet 192.168.0.10 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::b27:ebff:feb2:b5 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:eb:2d:b5 txqueuelen 1000 (Ethernet)
    RX packets 15587 bytes 1518593 (1.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16388 bytes 8355124 (7.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

3. ผู้อ่านสามารถใช้คำสั่ง ifconfig สำหรับปิด อุปกรณ์ wlan0 ดังนี้

```
$ sudo ifconfig wlan0 down
$ ifconfig
```

```

pi@raspberrypi:~$ ifconfig
eth0: flags=4099<UP, BROADCAST, MULTICAST> mtu 1500
    ether b8:27:eb:be:78:e0 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP, LOOPBACK, RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 5 bytes 284 (284.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5 bytes 284 (284.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

#### 4. ผู้อ่านสามารถใช้คำสั่ง ifconfig สำหรับเปิด อุปกรณ์ wlan0 ดังนี้

```
$ sudo ifconfig wlan0 up
$ ifconfig
```

จดว่าข้อความใดที่บ่งบอกว่า eth0 ทำงานแล้ว **แสดง wlan0**

```

$ ifconfig
eth0: flags=4096<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:27:eb:be:78:e0 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 5 bytes 284 (284.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5 bytes 284 (284.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.10 netmask 255.255.255.0 broadcast 192.168.0.255
    ether b8:27:eb:eb:2d:b5 txqueuelen 1000 (Ethernet)
    RX packets 1607 bytes 1818747 (1.8 MiB)
    RX errors 0 dropped 1 overruns 0 frame 0
    TX packets 1719 bytes 855300 (8.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
  
```

#### 5. นอกเหนือจากการเปิดปิดอุปกรณ์เครือข่าย ผู้อ่านสามารถตรวจสอบรายชื่อเครือข่าย WiFi ที่บอร์ดเคยเชื่อมต่อสำเร็จได้จากไฟล์ wpa\_supplicant.conf ซึ่งจะบันทึกรายละเอียดต่างๆ ของการเชื่อมต่อนั้นๆ รวมถึงพาสเวิร์ด (password) โดยพิมพ์คำสั่งต่อไปนี้ในโปรแกรม Terminal

```
$ cat /etc/wpa_supplicant/wpa_supplicant.conf
```

บันทึกผลที่ได้โดยกรอกในช่อง \_ เท่านั้น

```

network={
    ssid="_ pjpure _"
    psk="*****"
    key_mgmt=WPA-PSK
}
  
```

- ssid หมายถึง **ชื่อ network ที่ตั้งขึ้น**
- ssid ย่อมาจาก **Service Set Identifier**
- psk ย่อมาจาก **Pre-Shared Key**
- key\_mgmt คือ **ชนิดของการเข้ารหัส**

### I.4.3 การตรวจสอบการเชื่อมต่อกับเครือข่ายเบื้องต้น

เมื่อผู้อ่านเปิดและทำการเชื่อมต่อสำเร็จ แล้วจึงสามารถตรวจสอบการเชื่อมต่อในระดับชั้นเครือข่าย โดยใช้คำสั่ง ping ใน Terminal ดังนี้

```
$ ping <ip address or host name>
```

การตรวจสอบการเชื่อมต่อเบื้องต้น คือ การ ping ไปหาเราเตอร์ฝั่งต้นทางที่บอร์ดเชื่อมต่อ ผู้อ่านสามารถสืบค้นหมายเลขไอพีของเราเตอร์ที่ต้นทาง โดยสังเกตที่ inet ของ eth0 หรือ wlan0 ว่าเริ่มต้นด้วยหมายเลข 192.168.x.y ซึ่งเราเตอร์ต้นทางมักจะมีหมายเลข 192.168.x.1 หรือ 192.168.x.254

นี่เป็นตัวอย่างผลลัพธ์ที่ได้ของคำสั่ง ping 192.168.1.1 ที่ผู้อ่านจะต้องเติมหมายเลขลงใน \_ ที่เตรียมไว้ให้

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
```

```
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=2.03 ms
```

```
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.98 ms
```

64 bytes from 192.168.1.1: icmp\_seq=3 ttl=64 time=25.3 ms  
 64 bytes from 192.168.1.1: icmp\_seq=4 ttl=64 time=38.2 ms  
 64 bytes from 192.168.1.1: icmp\_seq=5 ttl=64 time=53.3 ms  
 64 bytes from 192.168.1.1: icmp\_seq=6 ttl=64 time=37.6 ms  
 64 bytes from 192.168.1.1: icmp\_seq=7 ttl=64 time=18.9 ms  
 64 bytes from 192.168.1.1: icmp\_seq=8 ttl=64 time=17.4 ms  
 64 bytes from 192.168.1.1: icmp\_seq=9 ttl=64 time=6.99 ms

โดย 192.168.1.1 คือหมายเลขไอพีแอดเดรสของอุปกรณ์ที่คำสั่งจะส่งแพ็กเก็ต ICMP (Internet Control Message Protocol) ความยาว 64 ไบต์ไป แล้วรออุปกรณ์หมายเลขนี้ตอบกลับมายังบอร์ด Pi3 โดยจับเวลาตั้งแต่ส่งไปและรอตอบกลับมา ของแพ็กเก็ตลำดับที่ \_\_ (icmp\_seq=\_\_ ) เป็นระยะเวลา 2.03 มิลลิวินาที ส่วน ttl=\_\_ ย่อมาจากคำว่า time to live หมายถึง เลขจำนวนเต็มที่ผู้ส่งกำหนดค่าอายุของแพ็กเก็ตที่สามารถเดินทางผ่านเครือข่าย หากตั้งไว้น้อยจะทำให้แพ็กเก็ตข้อมูลนี้อายุสั้นและอาจเดินทางไปไม่ถึงปลายทางเนื่องจากหมดอายุก่อน โดย ttl=64 เป็นค่าปกติ

ผู้อ่านจะสังเกตเห็นว่า ระยะเวลาที่มีค่าตั้งแต่ 1.98-53.3 มิลลิวินาที ขึ้นอยู่กับคุณภาพ ของสาย Ethernet หรือความแรงของสัญญาณ WiFi คุณภาพดีจะทำให้ระยะเวลาสั้นกว่า หลังจากตรวจสอบว่าบอร์ดสามารถเชื่อมต่อ กับเราเตอร์ต้นทางได้ตามตัวอย่างก่อนหน้านี้ ผู้อ่านสามารถใช้ตรวจสอบการเชื่อมต่อได้ว่า เราเตอร์ต้นทาง สามารถเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้สำเร็จหรือไม่ โดย Host name คือ ชื่อเซิร์ฟเวอร์ปลายทางที่จดทะเบียนโดเมนเนม (Domain Name) เรียบร้อยแล้ว เช่น ping [www.google.com](http://www.google.com)

## 1.5 กิจกรรมท้ายการทดลอง

1. จงค้นหาว่าความละเอียดของการแสดงผลผ่านพอร์ต HDMI ในหัวข้อที่ 1.1.2 เก็บบันทึกลงในไฟล์ชื่ออะไร ชื่อ [config.txt](#)
2. จงอธิบายความสัมพันธ์ระหว่าง แอดเดรสบัส 0xC000\_0000 และ แอดเดรสกายภาพที่มีชื่อว่า VC SDRAM ในรูปที่ 6.16 และเหตุใดจึงอยู่ในพื้นที่ Direct Uncached
3. ใช้คำสั่ง ifconfig ปิดอุปกรณ์ lo0 แล้วใช้คำสั่ง ping 127.0.0.1 ว่ามีการตอบสนองกลับมาหรือไม่ เปิดอุปกรณ์ lo0 แล้ว ping อีกรอบ จงอธิบายว่า 127.0.0.1 คืออะไร [เมื่อปิดไม่มีการตอบสนอง เมื่อเปิดมีการตอบสนอง 127.0.0.1 คือ localhost](#)
4. ใช้คำสั่ง ping เพื่อทดสอบเราเตอร์ที่อยู่ต้นทางของผู้อ่าน เช่น ping 192.168.x.1 หรือ 192.168.x.254 โดย x มีค่าเท่ากับ 0, 1, 2, ... จนกว่าจะมีการตอบสนองกลับมา [คำตอบอยู่ด้านล่าง](#)
5. ใช้คำสั่ง ping เพื่อตรวจสอบการเชื่อมต่อไปยัง [www.google.com](http://www.google.com)

```
pi@raspberrypi:~$ ping www.google.com
PING www.google.com (172.217.24.164) 56(84) bytes of data:
64 bytes from kul08s01-in-f4.1e100.net (172.217.24.164): icmp_seq=1 ttl=52 time=90.6 ms
64 bytes from kul08s01-in-f4.1e100.net (172.217.24.164): icmp_seq=2 ttl=52 time=74.2 ms
64 bytes from kul08s01-in-f4.1e100.net (172.217.24.164): icmp_seq=3 ttl=52 time=71.2 ms
64 bytes from kul08s01-in-f4.1e100.net (172.217.24.164): icmp_seq=4 ttl=52 time=72.7 ms
64 bytes from kul08s01-in-f4.1e100.net (172.217.24.164): icmp_seq=5 ttl=52 time=72.2 ms
```

[มีการตอบสนองจาก \[www.google.com\]\(http://www.google.com\)](#)

## 1. ข้อมูลในไฟล์ config.txt

```
GNU nano 3.2 config.txt
#overscan_right=16
#overscan_top=16
#overscan_bottom=16

# uncomment to force a console size. By default it will be display's size minus
# overscan.
#framebuffer_width=1280
#framebuffer_height=720

# uncomment if hdmi display is not detected and composite is being output
#hdmi_force_hotplug=1

# uncomment to force a specific HDMI mode (this will force VGA)
#hdmi_group=1
#hdmi_mode=1
hdmi_ignore_edid=0xa5000080
hdmi_group=2
hdmi_mode=83

# uncomment to force a HDMI mode rather than DVI. This can make audio work in
# DMT (computer monitor) modes
#hdmi_drive=2

# uncomment to increase signal to HDMI, if you have interference, blanking, or
# no display
#config_hdmi_boost=4
```

## 4. ใช้คำสั่ง ping 192.168.0.1 แล้วมีการตอบสนองกลับมา

```
pi@raspberrypi:~ $ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=3.74 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=16.7 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=6.16 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=6.81 ms
```