

นาย ก้ารพ์กิจ ษะมงคล ๖๒๐๑๐๖๘๔
นาย ภูมินทร์ เที่ยงสุข ๖๒๐๑๐๗๑๐

ภาคผนวก I

การทดลองที่ 9 การศึกษาและปรับแก้อินพุตและเอาท์พุตต่างๆ

การทดลองในภาคผนวกนี้ จะช่วยอธิบายเนื้อหาในบทที่ ๖ ซึ่งเกี่ยวข้องกับอุปกรณ์อินพุต/เอาท์พุตที่หลากหลายบนเครื่องคอมพิวเตอร์ตั้งต่อ โดยมีวัตถุประสงค์เหล่านี้

- เพื่อให้เข้าใจการปรับแก้อุปกรณ์อินพุตและเอาท์พุตนิดต่างๆ บนระบบปฏิบัติการ Raspbian
- เพื่อให้เข้าใจความแตกต่างระหว่างอุปกรณ์อินพุตและเอาท์พุตนิดต่างๆ บนบอร์ด Pi3
- เพื่อให้สามารถอ่านข้อมูลความแสดงรายการและอุปกรณ์อินพุตและเอาท์พุตนิดต่างๆ

หลักการและพื้นฐานความเข้าใจจะช่วยแนะนำทางให้ผู้อ่านสามารถศึกษาค้นคว้า อินพุต/เอาท์พุตอื่นๆ ในชิปและบอร์ดได้เพิ่มเติม รวมไปถึงบันโทรศัพท์เคลื่อนที่ แท็บเล็ตคอมพิวเตอร์ และอุปกรณ์อินเทอร์เน็ต สรรพสิ่ง (Internet of Things)

I.1 จอแสดงผลผ่านพอร์ท HDMI

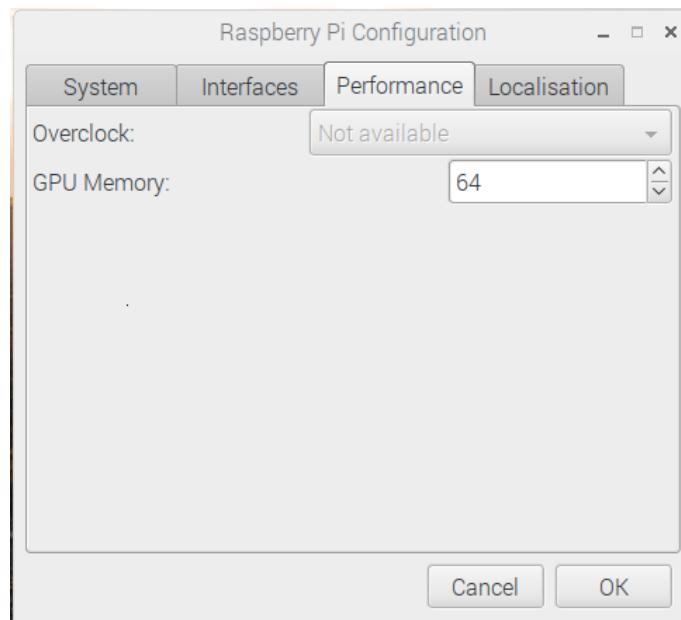
หน่วยความจำสำหรับจอแสดงผลหรือ GPU (Graphic Processing Unit) ถูกแบ่งเป็นที่ออกจากหน่วยความจำ DRAM บนบอร์ด เพื่อใช้งานร่วมกันทำให้ประหยัดต้นทุน แต่มีข้อเสียในด้านประสิทธิภาพจะลดลง เมื่อผู้ใช้งานต้องการภาพที่มี อัตราการเปลี่ยนแปลง (Refresh Rate) หรืออัตราเฟรมเรท (Frame Rate) สูง เช่น ภาพเคลื่อนไหว เกมส์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

I.1.1 การปรับแก้ขนาดหน่วยความจำของ GPU

ความละเอียดของจอแสดงผลขึ้นตระ簇กับขนาดของหน่วยความจำของ GPU ผู้อ่านสามารถปรับแก้ขนาดหน่วยความจำของ GPU ดังนี้

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

โดยหน้าต่างที่ปรากฏขึ้นมีลักษณะดังนี้ ผู้ใช้สามารถกำหนดขนาดที่ต้องการโดยขึ้นต่ำคือ 64 MB เพื่อให้ระบบสามารถแสดงผลได้ หากผู้ใช้กำหนดสูงเกินไป จะทำให้บอร์ดมีหน่วยความจำไม่เพียงพอ

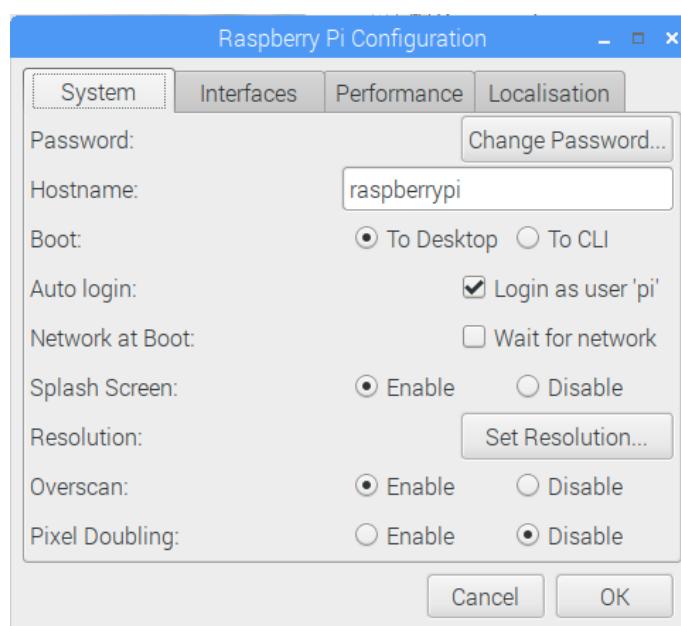


รูปที่ I.1: หน้าต่างกำหนดขนาดหน่วยความจำสำหรับ GPU ที่ 64 เมกะไบต์

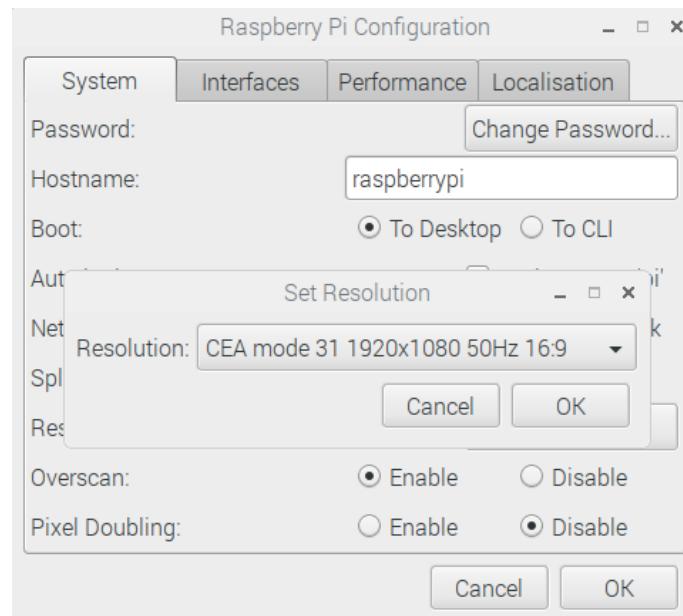
I.1.2 การปรับแก้ความละเอียดของจอแสดงผล

เมื่อขนาดหน่วยความจำของ GPU มีเพียงพอ ผู้ใช้สามารถปรับเพิ่มหรือลดความละเอียดของจอแสดงผลได้โดยกดปุ่มบนเมนูดังนี้

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

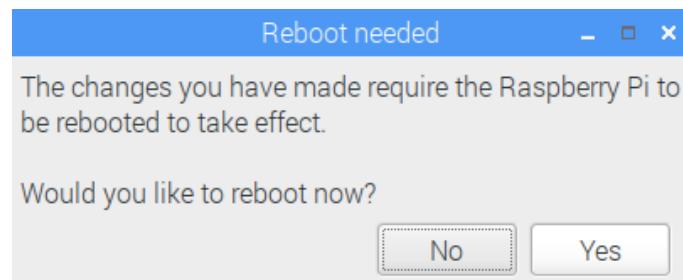


รูปที่ I.2: หน้าต่าง Raspberry Pi Configuration แท็บ System สำหรับกำหนดความละเอียดหน้าจอแสดงผล (Resolution)



รูปที่ 1.3: หน้าต่าง Set Resolution สำหรับกำหนดความละเอียดหน้าจอที่ต้องการ

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



รูปที่ 1.4: หน้าต่าง Reboot needed กดปุ่ม Yes เมื่อต้องการรีบูต ณ เวลานั้น

I.2 ระบบเสียงดิจิทัล

อุปกรณ์ระบบเสียงดิจิทัลที่ติดตั้งมาบนบอร์ด Pi3 จากโรงงาน ผู้ใช้สามารถเพิ่มเติมได้ผ่านพอร์ท USB และปรับแต่งระดับเสียงได้ เช่นกัน

I.2.1 รายชื่ออุปกรณ์ในระบบเสียง

ระบบเสียงในระบบปฏิบัติการ Linux ควบคุมการทำงานของเสียงผ่านระบบ ALSA (Advanced Linux Sound Architecture) ซึ่งจัดเตรียมไดรเวอร์ (Device Driver) สำหรับเสียงให้กับเครื่องเนล และอุปกรณ์ที่เกี่ยวข้องกับเสียงผ่านพอร์ท USB เช่น ไมโครโฟน, หูฟังพร้อมไมโครโฟน, เว็บแคม เป็นต้น ผู้อ่านสามารถแสดงรายชื่อไฟล์ หรือไดเรกทอรีที่เกี่ยวข้องกับระบบเสียงดังนี้

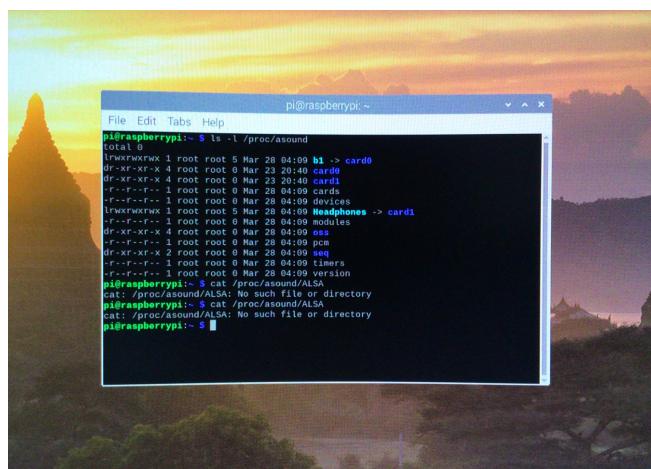
```
$ ls -l /proc/asound
lrwxrwxrwx 1 root root 5 Mar 26 20:59 ALSA -> card0
dr-xr-xr-x 4 root root 0 Mar 26 20:59 card0
-r--r--r-- 1 root root 0 Mar 26 20:59 cards
-r--r--r-- 1 root root 0 Mar 26 20:59 devices
-r--r--r-- 1 root root 0 Mar 26 20:59 modules
-r--r--r-- 1 root root 0 Mar 26 20:59 pcm
dr-xr-xr-x 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
```

ผลลัพธ์คือ รายชื่ออุปกรณ์ที่เกี่ยวข้องกับเสียง โดยเฉพาะ ALSA ซึ่งได้แสดงไปก่อนหน้านี้ ผู้อ่านจะสังเกตได้ว่าไดเรกทอรี /proc/asound/pcm จะเชื่อมโยงกับเนื้อหาในหัวข้อที่ 6.4 จะสังเกตเห็นว่ามีไดเรกทอรีชื่อ card0 อยู่สองตำแหน่งคือ ในແລະແຮກ และແລະที่มีชื่อ ALSA -> card0 สัญลักษณ์ -> เรียกว่าซิมบลิคลิงค์ (Symbolic Link) หมายความว่า ไดเรกทอรีชื่อ ALSA คือไดเรกทอรี card0

- ผู้อ่านสามารถทดสอบโดยพิมพ์คำสั่งต่อไปนี้

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

บันทึกผลลัพธ์ในพื้นที่ว่างต่อไปนี้



- ผู้อ่านสามารถทดสอบโดยพิมพ์คำสั่งต่อไปนี้

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

บันทึกผลลัพธ์ในพื้นที่ว่างต่อไปนี้ และเปรียบเทียบกับผลลัพธ์ก่อนหน้าว่าแตกต่างกันหรือไม่

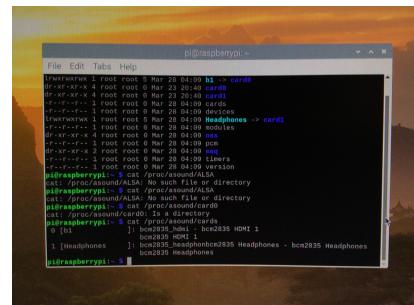
```
pi@raspberrypi:~ $ ls -l /proc/asound
total 0
lrwxrwxrwx 1 root root 5 Mar 28 04:09 b1 > card0
dr-xr-xr-x 4 root root 0 Mar 23 20:40 card0
dr-xr-xr-x 4 root root 0 Mar 23 20:40 card1
dr-xr-xr-x 4 root root 0 Mar 28 04:09 card2
dr-xr-xr-x 4 root root 0 Mar 28 04:09 cards
lrwxrwxrwx 1 root root 5 Mar 28 04:09 Headphones > card1
dr-xr-xr-x 1 root root 0 Mar 28 04:09 modules
dr-xr-xr-x 4 root root 0 Mar 28 04:09 oss
dr-xr-xr-x 2 root root 0 Mar 28 04:09 pcm
dr-xr-xr-x 2 root root 0 Mar 28 04:09 timer
dr-xr-xr-x 1 root root 0 Mar 28 04:09 timers
dr-xr-xr-x 1 root root 0 Mar 28 04:09 version
pi@raspberrypi:~ $ cat /proc/asound/ALSA
cat: /proc/asound/ALSA: No such file or directory
pi@raspberrypi:~ $ cat /proc/asound/ALSA
cat: /proc/asound/ALSA: No such file or directory
pi@raspberrypi:~ $ cat /proc/asound/card0
cat: /proc/asound/card0: is a directory
pi@raspberrypi:~ $
```

3. ค้นคว้าเพิ่มเติมเพื่อทำความหมายของ Symbolic Link และจดบันทึก
เมืองกรุงศรีวิชัยก้าวสำคัญ ก้าวที่ไม่ต้องเดิน ก้าวให้เมืองไฟล์ตั้งแต่บัดก่อน.
ที่อยู่ในลักษณะนี้ก้าวไม่ใช่วิธีการเข้าถึงไฟล์ link ก้าวจะได้ ภาระรับ
Symbolic link ภาระก็จะเป็นไปตามที่เราต้องการก้าวไปก้าวต่อไป

- #### 4. พิมพ์คำสั่งนี้ในโปรแกรม Terminal

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

โดยคำสั่ง cat ซึ่งได้อธิบายแล้วในการทดลองที่ 4 ภาคผนวก D สามารถอ่านไฟล์และแสดงข้อมูลภายในไฟล์ผ่านทางหน้าจอแสดงผล บันทึกในที่ว่างต่อไปนี้ 



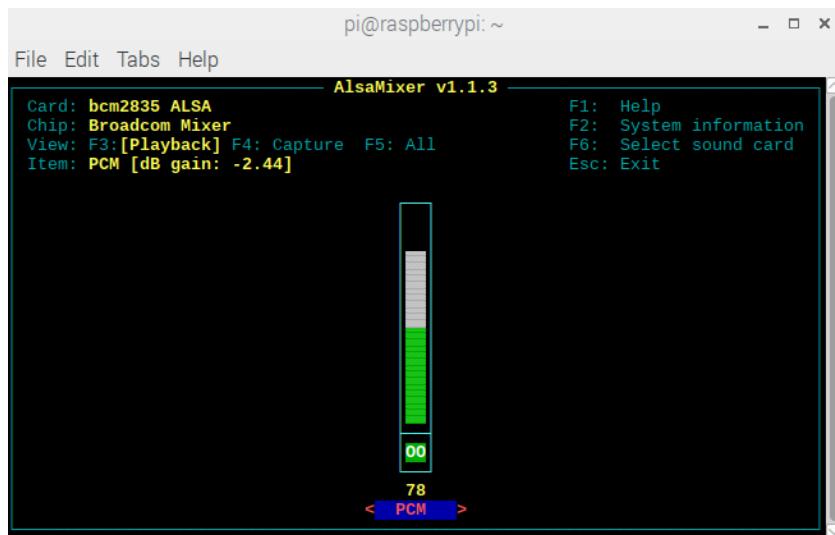
อภิปรายผลที่ได้ ดังนี้ ผลลัพธ์ได้จากบอร์ด Pi3 ใช้ชิป BCM283_ แต่ยังใช้ไดรเวอร์เสียงเดียวกันกับ BCM283_ โดย หมายเลข 0 คือ หมายเลขของระบบเสียงที่ติดตั้งใช้งานเพียงระบบเดียว และตรงกับ อุปกรณ์ชื่อ **b1** 0

I.2.2 การควบคุมระดับเสียง

ผู้อ่านสามารถควบคุมระดับความดังของเสียงทั้งด้านอินพุตและเอาท์พุต โดยพิมพ์คำสั่งนี้

```
$ alsamixer
```

หน้าต่างต่อไปนี้จะปรากฏขึ้น ผู้อ่านสามารถกดปุ่มลูกศรขึ้น/ลง เพื่อเพิ่ม/ลด ระดับความดังได้



รูปที่ I.5: โปรแกรม ALSA Mixer สำหรับควบคุมระดับเสียงบนบอร์ด Pi3

หมายเหตุ ผู้อ่านสามารถติดตั้งอุปกรณ์เสียงผ่านพอร์ท USB และใช้คำสั่งเหล่านี้เพื่อตรวจสอบและควบคุมการทำงาน

I.3 พор์ทเชื่อมต่ออุปกรณ์ USB

I.3.1 รายชื่ออุปกรณ์กับพอร์ท USB

1. ในการทดลองนี้ ขอผู้อ่านให้ดึงหัวเชื่อมต่อ USB ของมาส์ที่ใช้อยู่ออก และพิมพ์คำสั่งนี้ในโปรแกรม 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
```

ผู้อ่านจะเห็นรายชื่ออุปกรณ์ที่เชื่อมต่อกับพอร์ท USB เรียงลำดับย้อนกลับ จาก Device 005 - Device 001 โดย

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

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

2. บันทึกผลลัพธ์ของผู้อ่าน

Bus 001 Device 001 : ID = 1d6b:0002 Linux Foundation 3.0 root hub

Bus 001 Device 002 : ID = 0951:1642 Kingston Technology

Bus 001 Device 004 : ID = 2109:3432 VIA Labs, Inc. Hub

Bus 001 Device 001 : ID = 1d6b:0002 Linux Foundation 2.0 root hub

3. ผู้อ่านเสียบมาสักกลับเข้าไปที่พอร์ท USB ใหม่อีกครั้ง แล้วแสดงรายชื่ออุปกรณ์ USB ด้วยคำสั่ง

\$ lsusb

เข่นเดิม บันทึกผลงานในพื้นที่จัดเตรียมไว้ให้ โปรดสังเกตการเปลี่ยนแปลง

Bus 001 Device 001 : ID = 1d6b:0002 Linux Foundation 3.0 root hub

Bus 001 Device 002 : ID = 0951:1642 Kingston Technology

Bus 001 Device 004 : ID = 2109:3432 VIA Labs, Inc. Hub

Bus 001 Device 001 : ID = 1d6b:0002 Linux Foundation 2.0 root hub

4. รายการที่เพิ่มขึ้น คือ

Device 005 : ID = 1532:0037 Razer USA, Ltd

คือ เม้าส์หมายเลขลำดับที่ Device 005 ที่เพิ่งถูกเสียบกลับเข้าไปยังบอร์ด

I.3.2 รายละเอียดการเชื่อมต่ออุปกรณ์กับพอร์ต USB

คำสั่งต่อไป คือ **dmesg** สามารถแสดงรายการการทำงาน หรือ Log ของระบบปฏิบัติการว่าตั้งแต่เริ่มเปิดเครื่อง โดยคำว่า **dmesg** ย่อมาจากคำสั่ง “display message or display driver” ซึ่งเครื่องเนลได้บันทึกไว้ในบัฟเฟอร์ชนิดวงแหวน (Ring Buffer) ซึ่งข้อมูลความตอนต้นจะถูกเขียนทับเมื่อบัฟเฟอร์เต็ม

```
$ 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 rwdta,
              2076K rodata, 1024K init, 698K bss, 22328K reserved,
              8192K cma-reserved)
[0.000000] Virtual kernel memory layout:
              vector   : 0xfffff0000 - 0xfffff1000      (    4 kB)
              fixmap   : 0xfffc00000 - 0xfffff00000      (3072 kB)
              vmalloc  : 0xbb800000 - 0xff8000000      (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 คือเลขวินาทีตั้งแต่เครื่องเนลเริ่มทำงาน และ yyyy คือเศษวินาที ข้อมูลที่แสดงเป็น 0.000000 เนื่องจากเครื่องเนลอยู่ระหว่างการเริ่มต้น

- เริ่มต้นการรุบระบบปฏิบัติการด้วยซีพียูคอร์หมายเลข 0
- แสดงรายละเอียดหมายเลขเวอร์ชันของลีนุกซ์

- 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 แคชข้อมูลไม่สามารถแซร์ข้ามprocessorได้เนื่องจากทำงานแบบ nonaliasing ในขณะที่ แคชคำสั่งสามารถแซร์ข้ามprocessorได้ เนื่องจากทำงานแบบ aliasing **ข้อมูลเพิ่มเติม**
- แสดงผลการตรวจจับว่าเป็นบอร์ด Raspberry Pi 3 Model B Rev 1.2
- การทำงานของแคชข้อมูลเป็นชนิด writealloc ย่อมมาจาก Write Allocation ซึ่งซีพียูจะเขียนข้อมูลทั้ง ในแคชก่อน เมื่อแคชจะต้องถูกบ้ายอกจึงค่อยเขียนในหน่วยความจำหลักภายหลัง **ข้อมูลเพิ่มเติม**
- cma (Contiguous Memory Allocator) สำหรับขบวนการ DMA เริ่มต้นที่แอดเดรส 0x3AC00000 ขนาด 8 เมกะไบต์
- ...
- พื้นที่การจัดวางหน่วยความจำเสมือนของเครื่องใน (Virtual kernel memory layout) ผู้เขียนได้ทำการจัดเรียงใหม่ตามหมายเลขแอดเดรสที่ตำแหน่งมาก ไล่ลงมาจนถึงหมายเลขน้อย เพื่อให้ผู้อ่านมองเห็นภาพและเข้าใจง่ายขึ้น โดยแบ่งเป็นพื้นที่สำคัญๆ ตามลำดับดังนี้
 - จัดเก็บเวคเตอร์สำหรับการจัดจังหวะ (Interrupt Vector) ขนาด 4 กิโลไบต์ จากหมายเลข 0xFFFF_0000 ถึง 0xFFFF_1000
 - พื้นที่สำหรับจองหน่วยความจำเสมือน (vmalloc) ขนาด 1088 เมกะไบต์ จากหมายเลข 0xBB80_0000 ถึง 0xFF80_0000
 - bss เช็คเม้นท์ (.bss) ขนาด 699 กิโลไบต์ จากหมายเลข 0x80C9_7F10 ถึง 0x80D4_68B0
 - ดาต้าเช็คเม้นท์ (.data) ขนาด 577 กิโลไบต์ จากหมายเลข 0x80C0_0000 ถึง 0x80C9_017C
 - init เช็คเม้นท์ (.init) ขนาด 1024 กิโลไบต์ จากหมายเลข 0x80B0_8000 ถึง 0x80C0_0000
 - เท็กซ์เช็คเม้นท์ (.text) ขนาด 8160 กิโลไบต์ จากหมายเลข 0x8000_8000 ถึง 0x8080_0000

ในการทดลองนี้ ระบบสามารถตรวจจับอุปกรณ์ USB และติดตั้งไดรเวอร์ได้อย่างถูกต้องปราศจากข้อผิดพลาด

- ผู้อ่านสามารถล้างบัฟเฟอร์โดยใช้คำสั่ง ต่อไปนี้

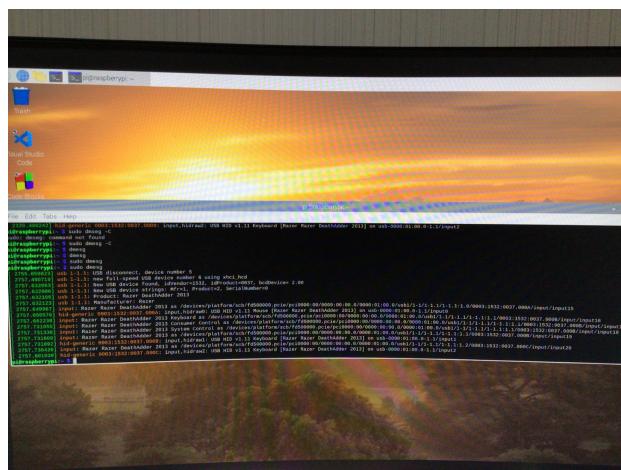
```
$ sudo dmesg -C
```

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

- ผู้อ่านจะต้องทดสอบและเสียบมาส์กกลับเข้าไปใหม่อีกรอบ
- ผู้อ่านสามารถแสดงข้อความที่เพิ่มเข้ามาในบัฟเฟอร์ได้อีก โดยเรียกคำสั่ง

```
$ sudo dmesg
```

4. จดบันทึก



5. อภิปรายผลลัพธ์ที่บันทึกได้ในพื้นที่ว่างต่อไปนี้

usb: disconnect, device number 5 → non usb

usb : new device found → few USB.

usb : product : Razer death adder → windows

usb : manufacturer : Razer → višen

၃ ရာဇ်ဝေါ်ကမျှ

usb: input: Razer → mouse is an input.

ในการเชื่อมต่อพอร์ท USB หากระบบแจ้งชื่ออุปกรณ์โดยไม่มีข้อความผิดพลาด แต่อุปกรณ์นั้นยังไม่สามารถทำงานได้ แสดงว่าอุปกรณ์ขาดซอฟท์แวร์ซึ่งทำหน้าที่เป็นดีไวซ์ไดเรกอร์ ขอให้ผู้อ่านค้นหาจากหมายเลขอประจำตัวของผู้ผลิต (idVendor) หากผู้ผลิตมิได้เปิดเผยซอฟท์แวร์ ผู้อ่านจำเป็นต้องดาวน์โหลดหรือคอมไพล์เองจากนักพัฒนารายอื่นแทน

I.4 พอร์ทเชื่อมต่อเครือข่าย WiFi และ Ethernet

I.4.1 รายชื่อผู้บรรยาย

- ผู้อ่านสามารถตรวจสอบรายชื่ออุปกรณ์สำหรับเชื่อมต่อเครือข่ายได้จากคำสั่ง `ifconfig` ทางโปรแกรม Terminal ตัวอย่างผลลัพธ์เป็นดังนี้

```
$ ifconfig
```

2. เติมข้อมูลหรือตัวเลขในช่องว่าง _____ ที่เตรียมไว้ให้จากผลลัพธ์ที่ได้ต่อไปนี้ ซึ่งลำดับรายการอาจแตกต่างกัน

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet _____._____._____._____.

netmask . . .

broadcast ____ : ____ : ____

} ყანე სივრცი = ყალბის სისტემა LAN

Tes WiFi #

```

ether dc:ab:32:f3:2a:52 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 192.168.1.7
        netmask 255.255.255.0
        broadcast 192.168.1.255
    ether dc:ab:32:f3:2a:52

```

3. โปรดสังเกตคำเริ่มต้นในแต่ละรายการ ค้นคว้า และกรอกรายละเอียดเพิ่มเติม ดังนี้

- eth0 หมายถึง **ethernet** → การเชื่อมต่อแบบ LAN
- lo0 หมายถึง **loop back device**
- wlan0 หมายถึง **wireless** → การเชื่อมต่อ internet แบบไร้สาย WiFi

I.4.2 การเปิด/ปิดอุปกรณ์เครือข่าย

1. ผู้อ่านสามารถเปิดอุปกรณ์ eth0 ได้ตามต้องการแล้วทำการตรวจสอบ ดังนี้

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

จดว่าข้อความใดที่บ่งบอกว่า eth0 ไม่ทำงานแล้ว

2. ผู้อ่านสามารถเปิดอุปกรณ์ eth0 ได้ตามต้องการแล้วทำการตรวจสอบ ดังนี้

```
$ sudo ifconfig eth0 up
```

```
$ ifconfig
```

จดว่าข้อความใดที่บ่งบอกว่า eth0 ทำงานแล้ว

3. ผู้อ่านสามารถใช้คำสั่ง ifconfig สำหรับปิด อุปกรณ์ wlan0 ดังนี้

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

4. ผู้อ่านสามารถใช้คำสั่ง ifconfig สำหรับเปิด อุปกรณ์ wlan0 ดังนี้

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

จดว่าข้อความใดที่บ่งบอกว่า eth0 ทำงานแล้ว

5. นอกเหนือจากการเปิดปิดอุปกรณ์เครือข่าย ผู้อ่านสามารถตรวจสอบรายชื่อเครือข่าย WiFi ที่บอร์ดโดย เชื่อมต่อสำเร็จได้จากไฟล์ wpa_supplicant.conf ซึ่งจะบันทึกรายละเอียดต่างๆ ของการเชื่อมต่อนั้นๆ รวมถึงพาสเวิร์ด (password) โดยพิมพ์คำสั่งต่อไปนี้ในโปรแกรม Terminal

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

บันทึกผลที่ได้โดยกรอกในช่อง _____ เท่านั้น

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

- ssid หมายถึง ชื่อ unique vor network ก่อให้เกิดการรับสัญญาณ ใน host เท่านั้น (ชื่อ WiFi นี้)
- ssid ย่อมาจาก Service Set Identifier
- psk ย่อมาจาก pre-shared key
- key_mgmt คือ มาตรฐาน protocol ก่อให้เกิดการรับสัญญาณ.

I.4.3 การตรวจสอบการเชื่อมต่อกับเครือข่ายเบื้องต้น

เมื่อผู้อ่านเปิดและทำการเชื่อมต่อสำเร็จ และจึงสามารถตรวจสอบการเชื่อมต่อในระดับขั้นเครือข่าย โดยใช้คำสั่ง ping ใน Terminal ดังนี้

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

การตรวจสอบการเชื่อมต่อเบื้องต้น คือ การ ping ไปหาเราเตอร์ผ่านทางที่บอร์ดเชื่อมต่อ 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 ที่ผู้อ่านจะต้องเติมหมายเลขใน _____ 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=_ ttl=_ time=25.3 ms
64 bytes from 192.168.1.1: icmp_seq=_ ttl=_ time=38.2 ms
64 bytes from 192.168.1.1: icmp_seq=_ ttl=_ time=53.3 ms
64 bytes from 192.168.1.1: icmp_seq=_ ttl=_ time=37.6 ms
64 bytes from 192.168.1.1: icmp_seq=_ ttl=_ time=18.9 ms
64 bytes from 192.168.1.1: icmp_seq=_ ttl=_ time=17.4 ms
64 bytes from 192.168.1.1: icmp_seq=_ ttl=_ time=6.99 ms
```

โดย 192.168.1.1 คือหมายเลขไอพีแอดเดรสของอุปกรณ์ที่คำสั่งจะส่งแพ็คเก็ต ICMP (Internet Control Message Protocol) ความยาว 64 ไบท์ไป แล้วรออุปกรณ์หมายเลขนี้ตอบกลับมายังบอร์ด Pi3 โดยจับเวลาตั้งแต่ส่งไปและรอตอบกลับมา ของแพ็คเก็ตลำดับที่ __ (icmp_seq=_) เป็นระยะเวลา 2.03 มิลลิวินาที ส่วน ttl=__ ย่อมาจากคำว่า time to live หมายถึง เลขจำนวนเต็มที่ผู้ส่งกำหนดค่าอายุของแพ็คเก็ตที่สามารถเดินทางผ่านเครือข่าย หากตั้งไว้น้อยจะทำให้แพ็คเก็ตข้อมูลนี้อยู่สั้นและอาจเดินทางไปไม่ถึงปลายทางเนื่องจากหมดอายุก่อน โดย ttl=64 เป็นค่าปกติ

ผู้อ่านจะสังเกตเห็นว่า ระยะเวลาเมื่อค่าตั้งแต่ 1.98-53.3 มิลลิวินาที ขึ้นอยู่กับคุณภาพ ของสาย Ethernet หรือความแรงของสัญญาณ WiFi คุณภาพดีจะทำให้ระยะเวลาสั้นกว่า หลังจากตรวจสอบว่าบอร์ดสามารถเชื่อมต่อกับเราเตอร์ต้นทางได้ตามตัวอย่างก่อนหน้า ผู้อ่านสามารถใช้ตรวจสอบการเชื่อมต่อได้ว่า เราเตอร์ต้นทางสามารถเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้สำเร็จหรือไม่ โดย Host name คือ ชื่อเซิร์ฟเวอร์ปลายทางที่จดทะเบียนโดเมนเนม (Domain Name) เรียบร้อยแล้ว เช่น ping www.google.com

I.5 กิจกรรมท้ายการทดลอง

1. จงค้นหาว่าความลະเอียดของการแสดงผลผ่านพอร์ท HDMI ในหัวข้อที่ [I.1.2](#) เก็บบันทึกลงในไฟล์ชื่ออะไร
2. จงอธิบายความสัมพันธ์ระหว่าง แอดเดรสบัส 0xC000_0000 และ แอดเดรสภายนอกที่มีชื่อว่า VC SDRAM ในรูปที่ [6.16](#) และเหตุใดจึงอยู่ในพื้นที่ Direct Uncached
3. ใช้คำสั่ง ifconfig ปิดอุปกรณ์ I00 แล้วใช้คำสั่ง ping 127.0.0.1 ว่ามีการตอบสนองกลับมาหรือไม่ เปิดอุปกรณ์ I00 แล้ว ping อีกรอบ จงอธิบายว่า 127.0.0.1 คืออะไร
4. ใช้คำสั่ง ping เพื่อทดสอบเราเตอร์ที่อยู่ต้นทางของผู้อ่าน เช่น ping 192.168.x.1 หรือ 192.168.x.254 โดย x มีค่าเท่ากับ 0, 1, 2, ... จนกว่าจะมีการตอบสนองกลับมา
5. ใช้คำสั่ง ping เพื่อตรวจสอบการเชื่อมต่อไปยัง www.google.com

4

```

File Edit Tabs Help
xpi@raspberrypi: ~
ping 192.168.0.1 ping statistics ...
18 packets transmitted, 18 packets received, 0% packet loss, time 737ms
rtt min/avg/max/mdev = 0.084/0.099/0.147/0.017 ms
ping: S: send ifconfig txq down No such device
10: ERROR while getting interface flags
pi@raspberrypi: ~

```

```

File Edit Tabs Help
xpi@raspberrypi: ~
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=65 time=20.1 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=65 time=1.54 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=65 time=1.49 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=65 time=1.61 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=65 time=124 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=65 time=1.45 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=65 time=1.45 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=65 time=3.12 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=65 time=1.45 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=65 time=1.44 ms
64 bytes from 192.168.1.1: icmp_seq=11 ttl=65 time=1.50 ms
64 bytes from 192.168.1.1: icmp_seq=12 ttl=65 time=1.49 ms
64 bytes from 192.168.1.1: icmp_seq=13 ttl=65 time=1.49 ms
64 bytes from 192.168.1.1: icmp_seq=14 ttl=65 time=1.49 ms
64 bytes from 192.168.1.1: icmp_seq=15 ttl=65 time=1.54 ms
64 bytes from 192.168.1.1: icmp_seq=16 ttl=65 time=1.49 ms
64 bytes from 192.168.1.1: icmp_seq=17 ttl=65 time=1.52 ms
64 bytes from 192.168.1.1: icmp_seq=18 ttl=65 time=1.52 ms
64 bytes from 192.168.1.1: icmp_seq=19 ttl=65 time=1.52 ms
64 bytes from 192.168.1.1: icmp_seq=20 ttl=65 time=1.47 ms
64 bytes from 192.168.1.1: icmp_seq=21 ttl=65 time=1.45 ms
64 bytes from 192.168.1.1: icmp_seq=22 ttl=65 time=1.47 ms
64 bytes from 192.168.1.1: icmp_seq=23 ttl=65 time=1.45 ms
64 bytes from 192.168.1.1: icmp_seq=24 ttl=65 time=2.71 ms
64 bytes from 192.168.1.1: icmp_seq=25 ttl=65 time=1.39 ms
64 bytes from 192.168.1.1: icmp_seq=26 ttl=65 time=1.39 ms
64 bytes from 192.168.1.1: icmp_seq=27 ttl=65 time=1.54 ms
64 bytes from 192.168.1.1: icmp_seq=28 ttl=65 time=1.57 ms
64 bytes from 192.168.1.1: icmp_seq=29 ttl=65 time=1.46 ms
64 bytes from 192.168.1.1: icmp_seq=30 ttl=65 time=1.45 ms
64 bytes from 192.168.1.1: icmp_seq=31 ttl=65 time=3.07 ms
64 bytes from 192.168.1.1: icmp_seq=32 ttl=65 time=1.77 ms
64 bytes from 192.168.1.1: icmp_seq=33 ttl=65 time=2.02 ms
64 bytes from 192.168.1.1: icmp_seq=34 ttl=65 time=1.99 ms
64 bytes from 192.168.1.1: icmp_seq=35 ttl=65 time=1.50 ms
64 bytes from 192.168.1.1: icmp_seq=36 ttl=65 time=1.59 ms
64 bytes from 192.168.1.1: icmp_seq=37 ttl=65 time=1.59 ms
64 bytes from 192.168.1.1: icmp_seq=38 ttl=65 time=1.45 ms
64 bytes from 192.168.1.1: icmp_seq=39 ttl=65 time=3.01 ms
64 bytes from 192.168.1.1: icmp_seq=40 ttl=65 time=1.70 ms
64 bytes from 192.168.1.1: icmp_seq=41 ttl=65 time=1.45 ms
64 bytes from 192.168.1.1: icmp_seq=42 ttl=65 time=1.47 ms
64 bytes from 192.168.1.1: icmp_seq=43 ttl=65 time=1.63 ms
64 bytes from 192.168.1.1: icmp_seq=44 ttl=65 time=1.53 ms
64 bytes from 192.168.1.1: icmp_seq=45 ttl=65 time=1.51 ms
64 bytes from 192.168.1.1: icmp_seq=46 ttl=65 time=1.49 ms
64 bytes from 192.168.1.1: icmp_seq=47 ttl=65 time=3.09 ms
64 bytes from 192.168.1.1: icmp_seq=48 ttl=65 time=1.50 ms
64 bytes from 192.168.1.1: icmp_seq=49 ttl=65 time=1.50 ms
64 bytes from 192.168.1.1: icmp_seq=50 ttl=65 time=1.49 ms
64 bytes from 192.168.1.1: icmp_seq=51 ttl=65 time=1.52 ms
64 bytes from 192.168.1.1: icmp_seq=52 ttl=65 time=1.49 ms
64 bytes from 192.168.1.1: icmp_seq=53 ttl=65 time=1.51 ms
64 bytes from 192.168.1.1: icmp_seq=54 ttl=65 time=1.51 ms
... more ...

```

IP Address : 192.168.1.1

↳ 7.192.168.2.1 ... 192.168.9.1 Үүрүү

more default gateway нийтийн тоо 192.168.1.1

5

```

File Edit Tabs Help
xpi@raspberrypi: ~
ping 8.8.8.8 ping statistics ...
41 packets transmitted, 41 packets received, 0% packet loss, time 100ms
rtt min/avg/max/mdev = 22.517/31.068/130.732/24.153 ms
pi@raspberrypi: ~

```

IP Address google : 8.8.8.8