

## การทดลองที่ 9 การศึกษาและปรับแก้อินพุตและเอาต์พุตต่างๆ

การทดลองในภาคผนวกนี้ จะช่วยอธิบายเนื้อหาในบทที่ 2 ซึ่งเกี่ยวข้องกับอุปกรณ์อินพุต/เอาต์พุตที่หลากหลายบนเครื่องคอมพิวเตอร์ตั้งโต๊ะ โดยมีวัตถุประสงค์เหล่านี้

- เพื่อให้เข้าใจการปรับแก้อุปกรณ์อินพุตและเอาต์พุตชนิดต่างๆ บนระบบปฏิบัติการ Raspbian
- เพื่อให้เข้าใจความแตกต่างระหว่างอุปกรณ์อินพุตและเอาต์พุตชนิดต่างๆ บนบอร์ด Pi3
- เพื่อให้สามารถอ่านข้อความแสดงรายละเอียดของอุปกรณ์อินพุตและเอาต์พุตชนิดต่างๆ

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

### I.1 จอแสดงผลผ่านพอร์ท HDMI

หน่วยความจำสำหรับจอแสดงผลหรือ GPU (Graphic Processing Unit) ถูกแบ่งพื้นที่ออกจาก หน่วยความจำ DRAM บนบอร์ด เพื่อใช้งานร่วมกันทำให้ประหยัดต้นทุน แต่มีข้อเสียในด้านประสิทธิภาพจะลดลงบ้าง เมื่อผู้ใช้งานต้องการภาพที่มี อัตราการเปลี่ยนแปลง (Refresh Rate) หรืออัตราเฟรมเรท (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

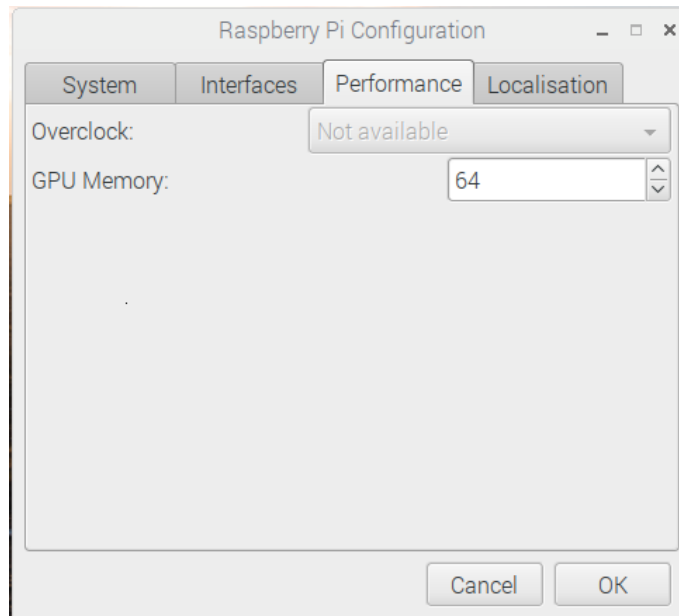


Figure I.1: หน้าต่างกำหนดขนาดหน่วยความจำสำหรับ GPU ที่ 64 เมกะไบต์

### I.1.2 การปรับแก้ความละเอียดของจอแสดงผล

เมื่อขนาดหน่วยความจำของ GPU มีเพียงพอ ผู้ใช้สามารถปรับเพิ่มหรือลดความละเอียดของจอแสดงผลได้ โดยกดปุ่มบนเมนูดังนี้

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

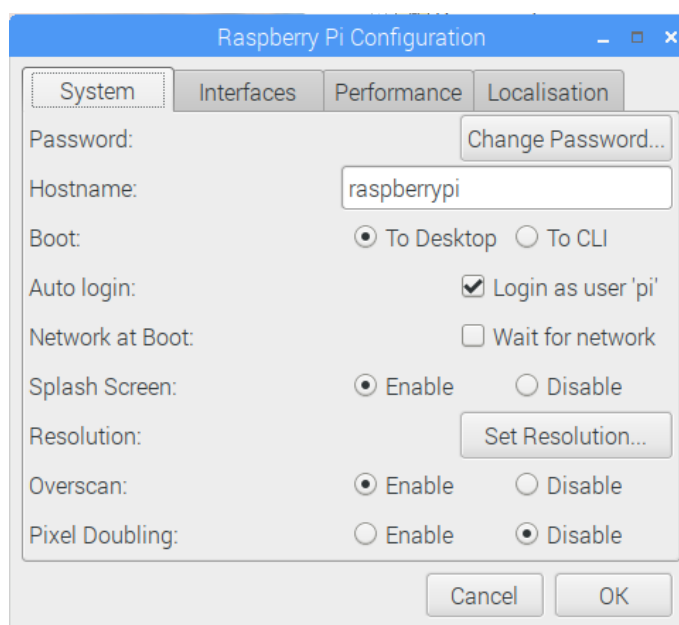


Figure I.2: หน้าต่าง Raspberry Pi Configuration แท็บ System สำหรับกำหนดความละเอียดหน้าจอแสดงผล (Resolution)

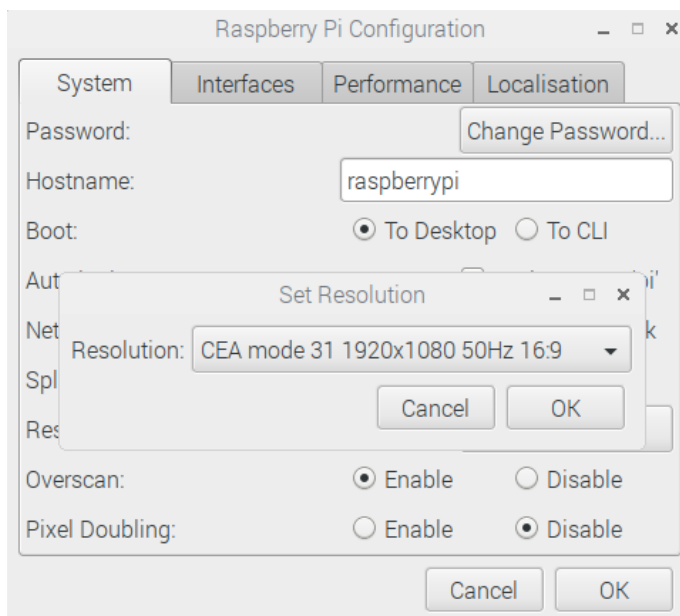


Figure I.3: หน้าต่าง Set Resolution สำหรับกำหนดความละเอียดหน้าจอที่ต้องการ

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

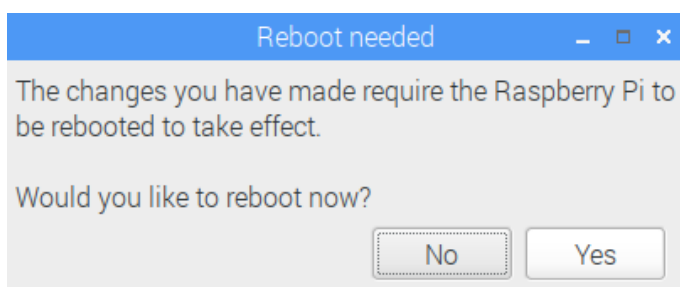


Figure I.4: หน้าต่าง Reboot needed กดปุ่ม Yes เมื่อต้องการรีบูต ณ เวลานั้น

## I.2 ระบบเสียง

อุปกรณ์ด้านระบบเสียงที่ติดตั้งมาบนบอร์ด Pi3 จากโรงงาน ผู้ใช้สามารถเพิ่มเติมได้ผ่านพอร์ท USB และปรับแต่งระดับเสียงได้เช่นกัน

### I.2.1 รายชื่ออุปกรณ์ด้านระบบเสียง

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

```
$ 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 ผู้อ่านจะสังเกตได้ว่าโพลเดอร์ `/proc/asound/pcm` จะเชื่อมโยงกับเนื้อหาในหัวข้อที่ 2.4 จะสังเกตเห็นว่ามีโพลเดอร์ชื่อ `card0` อยู่สองตำแหน่งคือ ในแถวแรก และแถวที่มีชื่อ `ALSA -> card0` สัญลักษณ์ `->` เรียกว่า Symbolic Link หมายความว่าชื่อ ALSA คือ `card0` ผู้อ่านสามารถทดสอบโดยพิมพ์คำสั่งต่อไปนี้

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

จะได้ผลลัพธ์เดียวกัน

ผู้ใช้พิมพ์คำสั่งนี้ในโปรแกรม Terminal

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

โดยคำสั่ง `cat` ซึ่งได้อธิบายแล้วในการทดลองที่ 4 ภาคผนวก D สามารถอ่านไฟล์และแสดงข้อมูลภายในไฟล์ผ่านทางหน้าจอแสดงผล ผลลัพธ์ตัวอย่างอาจแตกต่างกันไปตามฮาร์ดแวร์ที่ใช้งาน ดังนี้

```
0 [ALSA]: bcm2835_alsa - bcm2835 ALSA
      bcm2835 ALSA
```

ผลลัพธ์ได้จากบอร์ด Pi3 ที่ใช้ชิพเซ็ตของ BCM2835 และ ไดรเวอร์เดียวกันกับ BCM2837 โดย หมายเลข 0 คือ หมายเลขของระบบเสียงที่ติดตั้งใช้งานเพียงระบบเดียว และตรงกับชื่อ card0

## I.2.2 การควบคุมระดับเสียง

ผู้อ่านสามารถควบคุมระดับความดังของเสียงทั้งด้านอินพุตและเอาต์พุต โดยพิมพ์คำสั่งนี้

```
$ alsamixer
```

หน้าต่างต่อไปนี้จะปรากฏขึ้น ผู้อ่านสามารถกดปุ่มลูกศรขึ้น/ ลง เพื่อเพิ่ม/ลด ระดับความดังได้

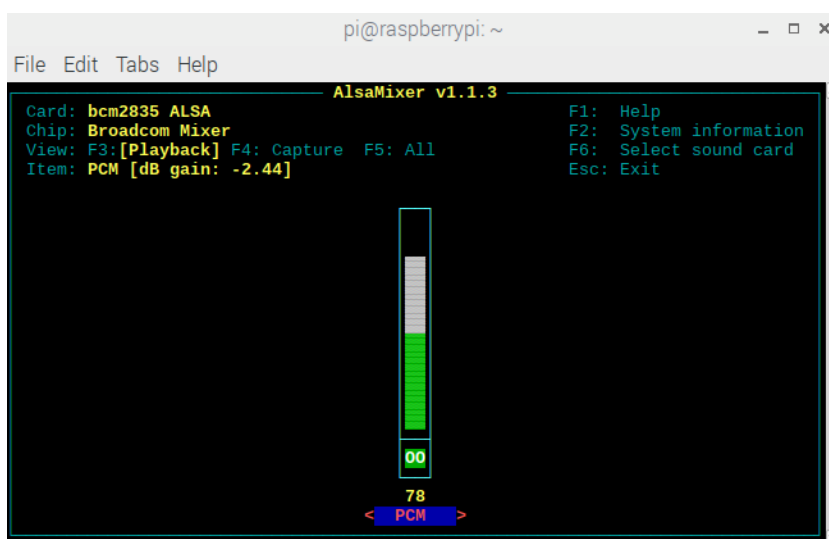


Figure I.5: โปรแกรม ALSA Mixer สำหรับควบคุมระดับเสียงบนบอร์ด Pi3

หมายเหตุ ผู้อ่านสามารถติดตั้งอุปกรณ์เสียงผ่านพอร์ต USB และใช้คำสั่งเหล่านี้เพื่อตรวจสอบและควบคุมการทำงาน

## I.3 พอร์ตเชื่อมต่ออุปกรณ์ USB

### I.3.1 รายชื่ออุปกรณ์กับพอร์ต USB

ในการทดลองนี้ ขอผู้อ่านให้ดึงหัวเชื่อมต่อ USB ของเมาส์ที่ใช้อยู่ออก แล้วพิมพ์คำสั่งนี้ในหน้าต่าง 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. ซึ่งคีย์บอร์ดของผู้อ่านจะแตกต่างจากที่ผู้เขียนอ้างอิงไม่ใช่ประเด็นสำคัญ
- 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

ผู้อ่านต้องเสียบเมาส์กลับเข้าไปที่พอร์ต USB ใหม่อีกครั้ง แล้วแสดงรายชื่ออุปกรณ์ USB ด้วยคำสั่ง lsusb เช่นเดิม โปรดสังเกตการเปลี่ยนแปลง

```
Bus 001 Device 005: ID 413c:2003 Dell Computer Corp. Keyboard
```

```
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
```

รายการที่เพิ่มขึ้น คือ

- Device 006 คือ เมาส์มีหมายเลข ID = 046d:c077 ผลิตโดย บริษัท Logitech, Inc. รุ่น M105

ซึ่งลำดับที่ตั้งแต่ Device 003 เป็นต้นไปอาจเปลี่ยนแปลงตามรายการและลำดับที่อุปกรณ์ของผู้อ่านทำการเชื่อมต่อกับบอร์ด Pi3 และผู้ผลิตเมาส์ของผู้อ่านจะแตกต่างจากที่ผู้เขียนบ้างก็อาจไม่ใช่ประเด็นสำคัญ

### I.3.2 รายละเอียดการเชื่อมต่ออุปกรณ์กับพอร์ท USB

คำสั่งต่อไป คือ **dmesg** สามารถแสดงรายการทำงาน หรือ Log ของระบบปฏิบัติการว่าตั้งแต่เริ่มเปิดเครื่อง โดยคำว่า **dmesg** ย่อมาจากคำสั่ง “display message or display driver” ซึ่งเคอร์เนลได้บันทึกไว้ในบัฟเฟอร์ชนิดวงแหวน (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 rwdata,
          2076K rodata, 1024K init, 698K bss, 22328K reserved,
          8192K cma-reserved)
[0.000000] Virtual kernel memory layout:
          vector   : 0xffff0000 - 0xffff1000   (   4 kB)
          fixmap   : 0xffc00000 - 0xfff00000   (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 คือเลขวินาทีตั้งแต่คอร์เนลเริ่มทำงาน และ yyyyyy คือเศษวินาที ข้อความที่แสดงเป็น 0.000000 เนื่องจากคอร์เนลอยู่ระหว่างการเริ่มต้น

- 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 แคชข้อมูลไม่สามารถแชร์ข้ามโพรเซสได้เนื่องจากทำงานแบบ nonaliasing ในขณะที่แคชคำสั่งสามารถแชร์ข้ามโพรเซสได้ 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
- ...
- พื้นที่การจัดวางหน่วยความจำเสมือนของคอร์เนล (Virtual kernel memory layout) ผู้เขียนได้ทำการจัดเรียงใหม่ตามหมายเลขแอดเดรสที่ตำแหน่งมาก ไหลลงมาจนถึงหมายเลขน้อย เพื่อให้ผู้อ่านมองเห็นภาพและเข้าใจง่ายขึ้น โดยแบ่งเป็นพื้นที่สำคัญๆ ตามลำดับดังนี้
  - จัดเก็บเว็คเตอร์สำหรับการขัดจังหวะ (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 เซ็กเมนต์ (.init) ขนาด 1024 กิโลไบต์ จากหมายเลข 0x80b0\_8000 ถึง 0x80c0\_0000
- เท็กซ์เซ็กเมนต์ (.text) ขนาด 8160 กิโลไบต์ จากหมายเลข 0x8000\_8000 ถึง 0x8080\_0000

ในตัวอย่างนี้ ระบบสามารถตรวจจับอุปกรณ์ USB และติดตั้งไดรเวอร์ได้อย่างถูกต้องปราศจาก ข้อผิดพลาด คำสั่งนี้จะแสดงรายละเอียดคร่าวๆ ของอุปกรณ์แต่ละตัวประมาณ 6-8 บรรทัด ผู้อ่านสามารถล้างบัฟเฟอร์โดยใช้คำสั่ง ต่อไปนี้

```
$ sudo dmesg -C
```

โดย -C คือ Clear เป็นคำสั่งเพิ่มเติมให้ dmesg ล้างข้อความในบัฟเฟอร์ออก โปรดสังเกต ตัว C ใหญ่ หลังจากนั้น ผู้อ่านทดสอบโดยการถอดเมาส์ออก แล้วเสียบกลับเข้าไปใหม่ ผู้อ่านสามารถแสดงข้อความที่เพิ่มเข้ามาในบัฟเฟอร์ได้อีก โดยเรียกคำสั่ง **dmesg** อีกรอบ โดยข้อความเหล่านี้ เกิดจากผู้เขียนถอดและเสียบเมาส์กลับเข้าไปใหม่อีกรอบ

```
[526.313715] usb 1-1.2: USB disconnect, device number 6
[527.653054] usb 1-1.2: new low-speed USB device number 7 using dwc_otg
[527.788253] usb 1-1.2: New USB device found, idVendor=046d, idProduct=c077
[527.788268] usb 1-1.2: New USB device strings: Mfr=1, Product=2,
          SerialNumber=0
[527.788277] usb 1-1.2: Product: USB Optical Mouse
[527.788285] usb 1-1.2: Manufacturer: Logitech
[527.793119] input: Logitech USB Optical Mouse as /devices/platform/soc/
          3f980000.usb/usb1/1-1/1-1.2/1-1.2:1.0/0003:046D:C077.0004/
          input/input3
[527.793804] hid-generic 0003:046D:C077.0004: input,hidraw0: USB HID v1.11
          Mouse [Logitech USB Optical Mouse] on usb-3f980000.usb-1.2/
          input0
```

ผู้อ่านจะเห็นว่า อุปกรณ์ USB หมายเลข 6 ขาดการเชื่อมต่อ หลังจากนั้นเวลาผ่านไปประมาณ 1.3 วินาที และเชื่อมต่อใหม่เป็นอุปกรณ์หมายเลข 7 โดยระบบเก็บรายละเอียดทั้งหมดและพบว่า เมาส์ USB นี้ มีหมายเลข idVendor=046d หมายถึง บริษัท Logitech, Inc. และ idProduct=c077 ซึ่งตรงกับเมาส์ที่ได้จากคำสั่ง **lsusb**

ในการเชื่อมต่อพอร์ท USB หากระบบแจ้งชื่ออุปกรณ์โดยไม่มีข้อความผิดพลาด แต่อุปกรณ์นั้นยังไม่สามารถทำงานได้ แสดงว่าอุปกรณ์ขาดซอฟต์แวร์ทำหน้าที่เป็นไดรเวอร์ ขอให้อ่านค้นหาจากหมายเลขประจำตัวของผู้ผลิต (idVendor) หากผู้ผลิตมิได้เปิดเผยซอฟต์แวร์ ผู้อ่านจำเป็นต้องดาวน์โหลดหรือคอมไพล์เองจากนักพัฒนารายอื่นแทน

## I.4 พอร์ตเชื่อมต่อเครือข่าย WiFi และ Ethernet

### I.4.1 รายชื่ออุปกรณ์เครือข่าย

ผู้อ่านสามารถตรวจสอบรายชื่ออุปกรณ์สำหรับเชื่อมต่อเครือข่ายได้จากคำสั่ง `ifconfig` ทางโปรแกรม Terminal ตัวอย่างผลลัพธ์เป็นดังนี้

```
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.33 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::440b:2da7:638f:9061 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:18:77:2d txqueuelen 1000 (Ethernet)
    RX packets 283 bytes 58857 (57.4 KiB)
    RX errors 0 dropped 2 overruns 0 frame 0
    TX packets 45 bytes 6515 (6.3 KiB)
    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 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

wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:27:eb:4d:22:78 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
```

โปรดสังเกตคำเริ่มต้นในแต่ละรายการ ดังนี้

- eth0 ซึ่งหมายถึงอุปกรณ์เชื่อมต่อเครือข่ายสาย
- lo0 ซึ่งหมายถึงอุปกรณ์ Loopback สำหรับทดสอบการเชื่อมต่อหาสัญญาณ Tx ย้อนกลับมาที่ขา Rx และ

- wlan0 ซึ่งหมายถึงอุปกรณ์เชื่อมต่อเครือข่าย WiFi

## I.4.2 การเปิดปิดอุปกรณ์เครือข่าย

ผู้อ่านสามารถใช้คำสั่ง `ifconfig` สำหรับเปิด หรือ ปิด อุปกรณ์ wlan0 ดังนี้

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

คำสั่ง `$ sudo ifconfig wlan0 down` สำหรับสั่งปิดอุปกรณ์ ส่วนคำสั่งต่อมาใช้ทดสอบว่าอุปกรณ์ wlan0 ยังมีอยู่ในรายการหรือไม่ คำสั่ง `$ sudo ifconfig wlan0 up` สำหรับสั่งเปิดอุปกรณ์ ส่วนคำสั่งสุดท้ายใช้ทดสอบว่าอุปกรณ์ wlan0 เปิดทำงานหรือยัง

ผู้อ่านสามารถเปิดปิดอุปกรณ์อื่นๆ โดยการพิมพ์ชื่อแทนที่ชื่ออุปกรณ์ wlan0 ได้ตามต้องการ เช่น

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

นอกเหนือจากการเปิดปิดอุปกรณ์เครือข่าย ผู้อ่านสามารถตรวจสอบรายชื่อเครือข่าย WiFi ที่บอร์ดเคยเชื่อมต่อสำเร็จได้จากไฟล์ `wpa_supplicant.conf` ซึ่งจะบันทึกรายละเอียดต่างๆ ของการเชื่อมต่อ นั้นๆ รวมถึงพาสเวิร์ด (password) โดยพิมพ์คำสั่งต่อไปนี้ใน Terminal

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

นี่เป็นตัวอย่างผลลัพธ์ที่ได้ โดย

```
network={
ssid="CE_ParaLab24"
psk="*****"
key_mgmt=WPA-PSK
}
```

- ssid หมายถึงชื่อเครือข่าย WiFi ซึ่งในตัวอย่าง คือ CE\_ParaLab24
- psk ย่อมาจาก Public Shared Key โดยผู้เขียนได้ใส่ตัวอักษรอื่นแทนเพื่อความปลอดภัย
- key\_mgmt คือ วิธีการเข้ารหัสและจัดการคีย์ หรือพาสเวิร์ด ซึ่งในตัวอย่าง คือ WPA-PSK ขึ้นกับผู้ดูแล (Administrator) ได้ติดตั้ง (Configure) อุปกรณ์ WiFi นั้น

### 1.4.3 การตรวจสอบการเชื่อมต่อกับเครือข่ายเบื้องต้น

เมื่อผู้อ่านเปิดและทำการเชื่อมต่อสำเร็จ แล้วจึงสามารถตรวจสอบการเชื่อมต่อในระดับชั้นเครือข่าย โดยใช้คำสั่ง ping ใน Terminal ดังนี้

```
$ ping <ip add 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

```
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 1 (icmp\_seq=1) เป็นระยะเวลา 2.03 มิลลิวินาที ส่วน ttl=64 ย่อมาจากคำว่า time to live หมายถึง เลขจำนวนเต็มที่ผู้ส่งกำหนดค่าอายุของแพ็กเก็ต ที่สามารถเดินทางผ่านเครือข่าย หากตั้งไว้น้อยจะทำให้แพ็กเก็ตข้อมูลนี้อายุสั้นและอาจเดินทางไปไม่ถึงปลายทางเนื่องจากหมดอายุก่อน โดย ttl=64 เป็นค่าปกติ

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

## I.5 กิจกรรมท้ายการทดลอง

1. จงค้นหาว่าความละเอียดของการแสดงผลผ่านพอร์ท HDMI ในหัวข้อที่ [1.1.2](#) เก็บบันทึกลงในไฟล์ชื่ออะไร **config.txt**
2. ใช้คำสั่ง ifconfig ปิดอุปกรณ์ lo0 แล้วใช้คำสั่ง ping 127.0.0.1 ว่ามีการตอบสนองกลับมาหรือไม่ เปิดอุปกรณ์ lo0 แล้ว ping อีกรอบ จงอธิบายว่า 127.0.0.1 คือ อะไร **หากปิดจะไม่มีการตอบสนอง คือ IP ของ localhost**
3. ใช้คำสั่ง ping เพื่อทดสอบเราเตอร์ที่อยู่ต้นทางของผู้อ่าน เช่น ping 192.168.x.1 หรือ 192.168.x.254 โดย x มีค่าเท่ากับ 0, 1, 2, ... จงกว่าจะมีการตอบสนองกลับมา **1, 2 มีการตอบสนอง**
4. ใช้คำสั่ง ping เพื่อตรวจสอบการเชื่อมต่อไปยัง www.google.com

```
pi@raspberrypi:~ $ ping www.google.com
PING www.google.com (172.217.166.132) 56(84) bytes of data.
64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp_seq=1 ttl=47 time
=30.8 ms
64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp_seq=2 ttl=47 time
=34.4 ms
64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp_seq=3 ttl=47 time
=51.4 ms
```

[เชื่อมต่อได้ปกติ](#)