

## การสร้าง Udev Rules สำหรับ ESP32 และ Lidar

ในการสร้าง Udev Rules ทั่วไป สิ่งที่ต้องจำเป็นอย่างยิ่งมี 2 อย่างหลักๆ คือ

**1. ID Vendor** (รหัสที่ใช้ระบุผู้ผลิต ของอุปกรณ์นั้นๆ)

**2. ID Product** (รหัสที่ใช้ระบุรุ่นหรือประเภทของอุปกรณ์ที่ผลิต)

เพราะ การ Set Udev Rules จะอ้างอิงจาก ID 2 ตัวนี้ตอนที่เสียบ Port USB

การทำ Udev Rules เพื่อแก้ปัญหาไม่ต้องปวดหัว สลับ Port /dev/tty\_USB0 >> /dev/tty\_USB1 ในกรณีที่ มีการย้ายเสียบ หรือ อุปกรณ์ไหนเสียบก่อนและหลัง การทำ Udev Rules นี้เราสามารถสร้างชื่อ Port USB เป็นชื่อที่ต้องการได้เลย หรือ ชื่อตามอุปกรณ์ได้เลย

**\*\*แต่ในกรณีนี้ ทั้ง ID Vendor และ ID Product เหมือนกันทั้ง 2 ตัวเลย ระหว่าง ESP32 และ Lidar จึงต้องมีการเขียน Doc นี้\*\*\*\***

### การสร้าง UDEV RULES แบบปกติ ( ID ไม่ซ้ำกัน )

- ขั้นตอนที่1 : `cd` // เพื่อทำการ Back มายัง Directory เริ่มต้นคือ Home
- ขั้นตอนที่2 : `cd /etc/udev/rules.d` // เข้าไปยัง Dir ที่เก็บ Udev Rules ทั้งหมดไว้ ก่อนทำการสร้าง ให้ `ls` ดูก่อนว่าซ้ำกับตัวไหนไหม

- ขั้นตอนที่3 : `sudo touch 75-esp32.rules` // เพื่อทำการสร้าง Udev Rules ของอุปกรณ์ที่ต้องการ (ตัวเลข ที่อยู่หน้าชื่ออุปกรณ์ คือ การเรียงลำดับการเรียกใช้ จากเลขน้อยไปมาก เลขน้อยก็จะถูกเรียกก่อน)
- ขั้นตอนที่4 : เปิดหน้าต่าง Terminal อันใหม่ขึ้นมา 1 อัน แล้ว พิมพ์ว่า `lsusb` แล้วจะแสดงหน้าต่าง ที่มี ID Vendor และ ID Product

```
maker@maker-vm:/etc/udev/rules.d$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 009: ID 10c4:ea60 Silicon Labs CP210x UART Bridge
Bus 001 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 001 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

คัดลอก หรือ จำไว้ ID แรกจะเป็น ID Vendor และตามด้วย ID Product

- ขั้นตอนที่5 : กลับมาที่ Terminal หลักที่ Path : `/etc/udev/rules.d` เปิด ไฟล์ที่สร้างใหม่ขึ้นมา `sudo gedit namefile.rules` แล้ว วาง โค้ดนี้ลงไป : `KERNEL=="ttyUSB*", ATTRS{idVendor}=="Id Vendor", ATTRS{idProduct}=="Id Product", MODE:="0666", SYMLINK+="Name"` \*\*\*แก้ไข ID Vendor และ ID Product เป็นของอุปกรณ์ที่ คัดลอกมา ตอนแรก\*\*\* จากนั้น Save ให้เรียบร้อย
- ขั้นตอนที่6 : `cd` // กลับมาที่ DIR Home

\*\*\* ถ้า ID Vendor และ ID Product ไม่เหมือนกันไม่ต้องทำวิธีด้านล่าง \*\*\*

**\*\* กรณีนี้ เพื่อเช็ค ว่า Attributes Port เหมือนหรือต่างกัน เพื่อใช้ในการทำ Udev Rules ที่ต่างประเภทกัน \*\***

**\*\* ตอนเช็ค Attributes Port เลียบที่ละอุปกรณ์ \*\*\***

- ขั้นตอนที่ 1 : `lsusb` // เพื่อหา ID Vendor และ ID Product

คัดลอก ชื่อของ Port ไว้เลย :

```
maker@maker-vm:/etc/udev/rules.d$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 009: ID 10c4:ea60 Silicon Labs CP210x UART Bridge
Bus 001 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 001 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

- ขั้นตอนที่ 2 : `ls /dev/ttyUSB*` // เพื่อเช็คดูว่าต่ออยู่ที่ Port ไหน

เช่น `/dev/ttyUSB0`

- ขั้นตอนที่ 3 : `udevadm info -a -n /dev/ttyUSB0 | grep serial`

จะได้หน้าตาแบบนี้ :

```
maker@maker-vm:~$ udevadm info -a -n /dev/ttyUSB0 | grep serial
SUBSYSTEMS=="usb-serial"
ATTRS{serial}=="0001"
ATTRS{serial}=="0000:02:00.0"
maker@maker-vm:~$
```

Copy Attributes Serial ตรงกลางไว้

ถอดอุปกรณ์ออก แล้ว เสียบอุปกรณ์ใหม่เข้าไปได้เลย

## จากนั้นทำตามเหมือนเดิมกับอุปกรณ์แรก

\*\*\* หลังจากเช็ค Attributes Port แล้ว ถ้าไม่เหมือนกัน ทำ Udev Rules วิธีต่อไปนี้ \*\*\*

- ขั้นตอนที่1 : เสียบทีละ อุปกรณ์เหมือนเดิม แล้ว lsusb
- ขั้นตอนที่2 : `ls /dev/ttyUSB` // เพื่อหาว่าเสียบอยู่ที่ Port ใด
- ขั้นตอนที่3 : `udevadm info -a -n /dev/ttyUSB0 | grep serial`  
คัดลอก Attributes Port อันที่สองมา

```
maker@maker-vm:~$ udevadm info -a -n /dev/ttyUSB0 | grep serial
SUBSYSTEMS=="usb-serial"
ATTRS{serial}=="0001"
ATTRS{serial}=="0000:02:00.0"
maker@maker-vm:~$
```

- ขั้นตอนที่4 : `cd /etc/udev/rules.d`
- ขั้นตอนที่5 : `sudo touch 75-ชื่ออุปกรณ์.rules`
- ขั้นตอนที่6 : `sudo gedit` ชื่อไฟล์ที่สร้าง.rules
- ขั้นตอนที่7 : วางโค้ดนี้ลงไป : `KERNEL=="ttyUSB[0-9]*",`  
`ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",`  
`ATTRS{serial} == "e2af823b8a98ef11807ccc63a8793231",`  
`SYMLINK="tty_esp32", GROUP="dialout", MODE="0666"`

ทำการแก้ไข ข้อมูล ID Vendor , ID Product และ ATTRS{serial} ให้ตรงกับบอร์ดของเรา **\*\* แก้ที่ตัวหนังสือสีแดงในโค้ด \*\***

**\*\* ตัวหนังสือ สีฟ้า ในโค้ด คือค่าที่บังคับให้เป็นชื่อนี้ \*\***

อีก 1 อุปกรณ์ก็ทำแบบนี้เช่นกัน..

### ถ้ากรณีที่ Attributes Port เหมือนกัน

จะใช้วิธี Fix อุปกรณ์ใดอุปกรณ์หนึ่ง ให้ใช้ Port ช่องนั้นเลย

**\*\*\* แต่ถ้ามีกรณีที่สมมุติว่า Fix ไว้ที่ Port 1-1.1 แต่ไปเสียบบที่ 1-1.2 Udev ที่สร้างขึ้นมานี้ก็จะไม่ทำงาน \*\*\***

**\*\*\* ทำตามวิธีด้านล่างนี้ \*\*\***

- ขั้นตอนที่1 : `ls /dev/ttyUSB*` // เพื่อเช็ค ว่า อุปกรณ์เสียบมาที่ Port ไหน

- ขั้นตอนที่2 : `udevadm info -a -n /dev/ttyUSB | grep`

KERNELS จะแสดงหน้าจอดังนี้ :

```
maker@maker-robot:~$ udevadm info -a -n /dev/ttyUSB0 | grep KERNELS
KERNELS=="ttyUSB0"
KERNELS=="1-1.3:1.0"
KERNELS=="1-1.3"
KERNELS=="1-1"
KERNELS=="usb1"
KERNELS=="0000:01:00.0"
KERNELS=="0000:00:00.0"
KERNELS=="pci0000:00"
KERNELS=="fd500000.pcie"
KERNELS=="scb"
KERNELS=="platform"
```

สังเกตที่ กรอบสีเหลือง จะมีตัวเลขของ Port ว่าตอนนี้เสียบอยู่ที่ไหน(คือ ภายนอกภาพภายในพอร์ต) 1-1.1 = สีส้ม, 1-1.2 = สีแดง, 1-1.3 = สีเขียว 1-1.4 = สีฟ้า



- ขั้นตอนที่3 : `cd /etc/udev/rules.d` // เพื่อเข้าไปสร้าง Udev
- ขั้นตอนที่4 : `ls` // เพื่อดูว่า Udev มีลำดับเท่าไรบ้างเพื่อป้องกันสร้างทับ Udev อื่น
- ขั้นตอนที่5 : `sudo touch 80-esp32.rules` // สร้างไฟล์ Udev
- ขั้นตอนที่6 : `sudo gedit 80-esp32.rules` // เพื่อเข้าไปวางโค้ด

- ขั้นตอนที่ 7 : วางโค้ด KERNEL=="ttyUSB[0-9]\*",  
 KERNELS=="1-1.2" , ATTRS{idVendor}=="10c4",  
 ATTRS{idProduct}=="ea60", ATTRS{serial} == "001",  
 SYMLINK="tty\_esp32", GROUP="dialout", MODE="0666"

\*\* ที่ตัวหนังสือ สีแดง ให้แก้เป็นค่าที่เราได้มาจากการทำขั้นตอนๆแรก\*\*

- ขั้นตอนที่ 7 : Save File และ ปิด Text Editor

\*\*\* อุปกรณ์อีกตัวที่ซ้าก็ทำแบบเดียวกัน เพียงแค่เปลี่ยน

KERNELS=="1-1.2" อย่าให้ Port ซ้ำกัน เช่น ให้เป็น 1-1.1

\*\*\*