



Application Note

AN_552

Consistent COM Port Enumeration on Linux

Version 1.0

Issue Date: 03-08-2023

This Application Note shows how to utilize udev rules on a Linux system to ensure consistent COM port allocation for an FTDI device.

Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify, and hold FTDI harmless from any and all damages, claims, suits, or expense resulting from such use.

Future Technology Devices International Limited (FTDI)

Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom

Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758

Web Site: <http://ftdichip.com>

Copyright © Future Technology Devices International Limited

Table of Contents

1 Introduction	3
1.1 Background	3
2 A Working Example	4
2.1 Standard Enumeration.....	4
2.2 Obtaining Device Properties	5
2.3 Example udev rule.....	6
2.4 Other uses for udev rules	8
3 Conclusion	9
4 Contact Information	10
Appendix A – References	11
Document References	11
Acronyms and Abbreviations.....	11
Appendix B – List of Tables & Figures	12
List of Figures	12
Appendix C – Revision History	13

1 Introduction

Many FTDI customers utilize multiple devices on Linux hosts, enumeration order however is not guaranteed within the Linux kernel and 'COM Ports' may be assigned in an arbitrary order.

In certain scenarios, it will benefit the customer to ensure a given device is locked to a given 'COM port' entry on the system.

This Application note describes a method for ensuring the consistent enumeration of a device on a Linux operating system.

1.1 Background

The Linux Virtual COM Port (VCP) driver (ftdi_sio) which is included within the Linux Kernel will automatically attach to an appropriate device and create a COM port for this device within the '/dev' folder on the Linux system.

Device drivers are attached to a given device in Linux via the 'udev' device manager, this manages user space (the memory area application software and some drivers execute) events that occur when new hardware devices are attached or removed from the system. It is also responsible for loading device drivers when required.

The udev subsystem allows users to create rules which can be used to match on properties like the kernel subsystem, the device name, the physical location of the device, or properties like the device's serial number. Rules can also be used to name a device or specify a custom name that will always be the same, regardless of the order devices are discovered by the system.

2 A Working Example

In this example a [US232R-100](#) cable and a [FT4232H Mini-Module](#) are utilized on a Ubuntu 20.04 Linux Operation system. The US232R-100 cable provides a USB to RS232 bridge and the FT4232H Mini-Module provides a 4 channel USB to UART bridge to the host PC.

A user may have different intentions for the 4 channels of the FT4232H Mini-Module, for example they may wish to utilize channels A and B of the device as standard UART channels, whilst using Channel C with an RS485 line driver and channel D with a RS422 line driver.

By utilising a udev rule, we can map the device entries in the '/dev' folder so they are easily recognisable by the user e.g. having an RS232 entry for the US232R-100 and RS485/RS422 entries for channels C and D of the FT4232H Mini-Module.

2.1 Standard Enumeration

When the US232R-100 cable and an FT4232H Mini-Module are attached to the Linux host the udev system will automatically assign these devices the first available COM ports. **Figure 1** below shows the dmesg command output in a terminal for these two devices after they have been enumerated.

```
[ 110.811278] usb 1-1: new high-speed USB device number 2 using ehci-pci
[ 111.100361] usb 1-1: New USB device found, idVendor=0403, idProduct=6011, bcdDevice= 8.00
[ 111.100371] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 111.100421] usb 1-1: Product: FT4232H MiniModule
[ 111.100424] usb 1-1: Manufacturer: FTDI
[ 111.100427] usb 1-1: SerialNumber: FTSC1G80
[ 112.036222] usbcore: registered new interface driver usbserial_generic
[ 112.036245] usbserial: USB Serial support registered for generic
[ 112.167514] usbcore: registered new interface driver ftdi_sio
[ 112.167538] usbserial: USB Serial support registered for FTDI USB Serial Device
[ 112.167580] ftdi_sio 1-1:1.0: FTDI USB Serial Device converter detected
[ 112.167627] usb 1-1: Detected FT4232H
[ 112.170810] usb 1-1: FTDI USB Serial Device converter now attached to ttyUSB0
[ 112.170919] ftdi_sio 1-1:1.1: FTDI USB Serial Device converter detected
[ 112.170959] usb 1-1: Detected FT4232H
[ 112.175497] usb 1-1: FTDI USB Serial Device converter now attached to ttyUSB1
[ 112.175814] ftdi_sio 1-1:1.2: FTDI USB Serial Device converter detected
[ 112.176033] usb 1-1: Detected FT4232H
[ 112.180279] usb 1-1: FTDI USB Serial Device converter now attached to ttyUSB2
[ 112.180648] ftdi_sio 1-1:1.3: FTDI USB Serial Device converter detected
[ 112.180767] usb 1-1: Detected FT4232H
[ 112.184259] usb 1-1: FTDI USB Serial Device converter now attached to ttyUSB3
[ 181.896700] usb 2-2: new full-speed USB device number 3 using ohci-pci
[ 182.204774] usb 2-2: New USB device found, idVendor=0403, idProduct=6001, bcdDevice= 6.00
[ 182.204781] usb 2-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 182.204783] usb 2-2: Product: USB to Serial Converter
[ 182.204784] usb 2-2: Manufacturer: FTDI
[ 182.204785] usb 2-2: SerialNumber: LT66XJVA
[ 182.214836] ftdi_sio 2-2:1.0: FTDI USB Serial Device converter detected
[ 182.214892] usb 2-2: Detected FT232RL
[ 182.226867] usb 2-2: FTDI USB Serial Device converter now attached to ttyUSB4
```

Figure 1 - dmesg output after enumeration

From the above figure we can see that the FT4232H Mini Module has been attached to the /dev entries ttyUSB{0-3}, and the US232R-100 has been attached to ttyUSB4. **Figure 2** shows these devices listed within the /dev folder in the operating system.

```
ftdi@ftdi-VirtualBox:~$ ls /dev/ttyUSB*
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3 /dev/ttyUSB4
```

Figure 2 - Assigned COM ports for attached devices

2.2 Obtaining Device Properties

In order to write an effective udev rule to properly map the COM ports to more easily identifiable entries within the /dev folder we need to obtain device attributes {ATTRS} for a given device. These {ATTRS} are used by the udev subsystem to identify devices and will be utilized within a udev rule for the same purpose.

The following command can be used in a terminal to list a devices ATTRS{x} when the ftdi_sio driver is loaded:

```
sudo udevadm info -a -n /dev/ttyUSBx
```

Figure 3 below shows the output of this command for the /dev/ttyUSB0 entry:

```
ftdi@ftdi-VirtualBox:~$ sudo udevadm info -a -n /dev/ttyUSB0
[sudo] password for ftdi:
Udevadm info starts with the device specified by the devpath and then
walks up the chain of parent devices. It prints for every device
found, all possible attributes in the udev rules key format.
A rule to match, can be composed by the attributes of the device
and the attributes from one single parent device.

looking at device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0/ttyUSB0/tty/ttyUSB0':
  KERNEL=="ttyUSB0"
  SUBSYSTEM=="tty"
  DRIVER==" "

looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0/ttyUSB0':
  KERNELS=="ttyUSB0"
  SUBSYSTEMS=="usb-serial"
  DRIVERS=="ftdi_sio"
  ATTRS{port_number}=="0"
  ATTRS{latency_timer}=="16"

looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0':
  KERNELS=="1-1:1.0"
  SUBSYSTEMS=="usb"
  DRIVERS=="ftdi_sio"
  ATTRS{supports_autosuspend}=="1"
  ATTRS{bInterfaceProtocol}=="ff"
  ATTRS{authorized}=="1"
  ATTRS{interface}=="FT4232H MiniModule"
  ATTRS{bNumEndpoints}=="02"
  ATTRS{bInterfaceNumber}=="00"
  ATTRS{bInterfaceClass}=="ff"
  ATTRS{bInterfaceSubClass}=="ff"
  ATTRS{bAlternateSetting}==" 0"

looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1':
  KERNELS=="1-1"
  SUBSYSTEMS=="usb"
  DRIVERS=="usb"
  ATTRS{configuration}==" "
  ATTRS{ltm_capable}=="no"
  ATTRS{devpath}=="1"
  ATTRS{maxchild}=="0"
  ATTRS{bConfigurationValue}=="1"
  ATTRS{busnum}=="1"
  ATTRS{version}==" 2.00"
  ATTRS{bNumConfigurations}=="1"
  ATTRS{authorized}=="1"
  ATTRS{tx_lanes}=="1"
  ATTRS{bNumInterfaces}==" 4"
  ATTRS{urbnum}=="22"
  ATTRS{bcdDevice}=="0800"
  ATTRS{serial}=="FTSC1G80"
  ATTRS{removable}=="unknown"
  ATTRS{quirks}=="0x0"
  ATTRS{bMaxPower}=="150mA"
  ATTRS{bDeviceProtocol}=="00"
  ATTRS{product}=="FT4232H MiniModule"
  ATTRS{devnum}=="2"
  ATTRS{avoid_reset_quirk}=="0"
  ATTRS{bMaxPacketSize0}=="64"
  ATTRS{idVendor}=="0403"
  ATTRS{bMAttributes}=="a0"
  ATTRS{idProduct}=="6011"
  ATTRS{manufacturer}=="FTDI"
  ATTRS{bDeviceSubClass}=="00"
  ATTRS{speed}=="480"
  ATTRS{bDeviceClass}=="00"
  ATTRS{rx_lanes}=="1"
```

Figure 3 - udevadm info output for /dev/ttyUSB0

We can see a variety of attributes associated with the device attached to /dev/ttyUSB0 in **Figure 3** above. Particularly of note is the devices Product feild (FT4232H MiniModule), Vendor ID (0403), Product ID (6011), Serial Number (FTSC1G80) and Interface Number (00).

These represent the main attributes to be used within a udev rule to identify the FTDI devices we are interested in mapping to new COM port entries.

Note: the interface number is particularly useful for multi channel devices as this maps directly to the Channel of the device. For example interface = 00 represents Channel A of the FT4232H, interface = 01 represents Channel B, etc.

If the VCP driver has been detached (unbound or unloaded) from the device it won't show up in the /dev folder but you can still examine it's attributes using a /sys/bus/usb/devices link. For example to view the attributes of a device onbus 1, port 2:

```
sudo udevadm info --attribute-walk /sys/bus/usb/devices/1-2
```

2.3 Example udev rule

The ATTRS{x} discovered in Section 2.2 can be made to match any of the following in 3 levels, Whole Device (FT4232H), port on a device (FT4232H Channel B) or driver (/dev/ttyUSBx). This means that only ATTRS{x} which are grouped together in the output of the '**udevadm info**' command can be used for a given device rule (for example it is not possible to use ATTRS{serial} and ATTRS{bInterfaceNumber} in the same rule). Matching is done as per a logical AND but wildcards are allowed.

The following procedure can be followed to create a udev rule, please ensure the FTDI devices are disconnected from your system during this process:

1. Open a terminal window in Linux.
2. Navigate to the **/etc/udev/rules.d** directory.
3. Using the preferred Linux text editor, create a new file with the following name:

```
99-usbftdi.rules
```

(**note:** The "sudo" command can be used here to create files.)

4. Enter your desired udev rule, then save and close the file.
5. Once the udev rule has been created issue the following command to reload the systems udev rules:

```
sudo udevadm control --reload-rules && udevadm trigger
```

In our example the following 99-usbftdi.rules file has been created with 5 rules:

```
# Rule to remap FTDI FT232R & FT4232H USB devices for consistent COM port
enmueration

KERNEL=="ttyUSB[0-9]*", SUBSYSTEM=="tty", SUBSYSTEMS=="usb",
ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001",
ATTRS{serial}=="LT66XJVA",SYMLINK+="ttyUSB-F232" GROUP="ray" MODE="0666"

KERNEL=="ttyUSB[0-9]*", SUBSYSTEM=="tty", SUBSYSTEMS=="usb",
ATTRS{interface}=="FT4232H MiniModule", ATTRS{bInterfaceNumber}=="00",
SYMLINK+="ttyUSB-UARTA" GROUP="ray" MODE="0666"

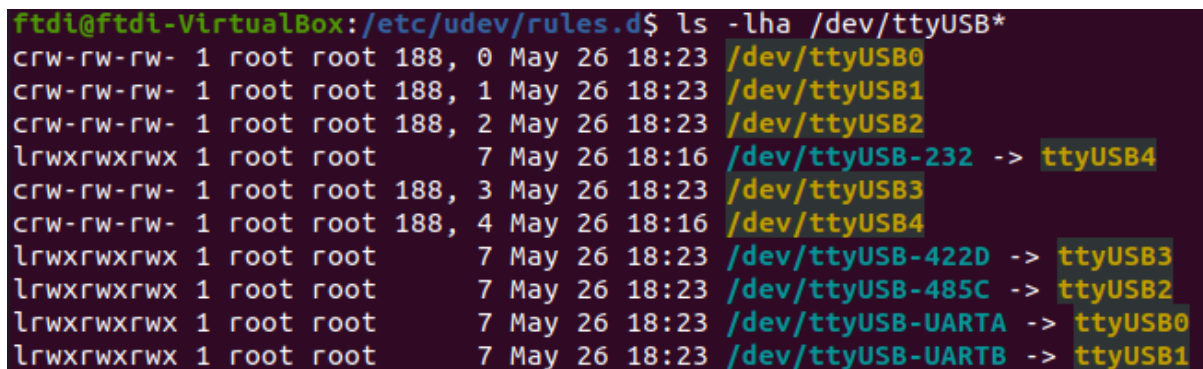
KERNEL=="ttyUSB[0-9]*", SUBSYSTEM=="tty", SUBSYSTEMS=="usb",
ATTRS{interface}=="FT4232H MiniModule", ATTRS{bInterfaceNumber}=="01",
SYMLINK+="ttyUSB-UARTB" GROUP="ray" MODE="0666"

KERNEL=="ttyUSB[0-9]*", SUBSYSTEM=="tty", SUBSYSTEMS=="usb",
ATTRS{interface}=="FT4232H MiniModule", ATTRS{bInterfaceNumber}=="02",
SYMLINK+="ttyUSB-485C" GROUP="ray" MODE="0666"

KERNEL=="ttyUSB[0-9]*", SUBSYSTEM=="tty", SUBSYSTEMS=="usb",
ATTRS{interface}=="FT4232H MiniModule", ATTRS{bInterfaceNumber}=="03",
SYMLINK+="ttyUSB-422B" GROUP="ray" MODE="0666"
```

Once the rule has been created and the udev rules have been reloaded and triggered, symbolic links will be created in the /dev folder for the COM ports assigned according to the above definitions.

We can see in **Figure 4** below that each /dev/ttyUSBx entry has now been mapped to a new entry within the /dev folder which can be used to easily identify which COM port belongs to which device, or channel of that device. For example /dev/ttyUSB4 the US232R-100 is mapped to /dev/ttyUSB-232 and /dev/ttyUSB2 channel C of the FT4232H Mini-Module has been mapped to /dev/ttyUSB-485C.



```
ftdi@ftdi-VirtualBox:/etc/udev/rules.d$ ls -lha /dev/ttyUSB*
crw-rw-rw- 1 root root 188, 0 May 26 18:23 /dev/ttyUSB0
crw-rw-rw- 1 root root 188, 1 May 26 18:23 /dev/ttyUSB1
crw-rw-rw- 1 root root 188, 2 May 26 18:23 /dev/ttyUSB2
lrwxrwxrwx 1 root root 7 May 26 18:16 /dev/ttyUSB-232 -> ttyUSB4
crw-rw-rw- 1 root root 188, 3 May 26 18:23 /dev/ttyUSB3
crw-rw-rw- 1 root root 188, 4 May 26 18:16 /dev/ttyUSB4
lrwxrwxrwx 1 root root 7 May 26 18:23 /dev/ttyUSB-422D -> ttyUSB3
lrwxrwxrwx 1 root root 7 May 26 18:23 /dev/ttyUSB-485C -> ttyUSB2
lrwxrwxrwx 1 root root 7 May 26 18:23 /dev/ttyUSB-UARTA -> ttyUSB0
lrwxrwxrwx 1 root root 7 May 26 18:23 /dev/ttyUSB-UARTB -> ttyUSB1
```

Figure 4 – Device mapping after the udev rules have been created

Users can utilize these mappings in the /dev folder to access a device with standard COM port programming techniques or via a terminal emulator such as PuTTY.

2.4 Other uses for udev rules

Udev rules can also be utilized to unbind the VCP driver from a given device so that users can access the device via our [D2XX](#). This can be particularly useful if you have multiple devices and require to utilize the D2XX and VCP drivers simultaneously. Please see below for an example of a udev rule which will detach the ftdi_sio driver from a FT4232H Mini-Module:

```
# Rule to unbind the ftdi_sio driver from a FT4232H MiniModule
ACTION=="add", SUBSYSTEMS=="usb", ATTRS{idVendor}=="0403",
ATTRS{idProduct}=="6011", ATTRS{product}=="FT4232H MiniModule",
KERNEL=="ttyUSB*" \
RUN+="/bin/sh -c '[ ! -z \"$DEVNAME\" ] && ls -d
/sys/bus/usb/drivers/ftdi_sio/*/${basename $DEVNAME} | cut -d \"/\\" -f 7 | tee
/sys/bus/usb/drivers/ftdi_sio/unbind | /usr/bin/logger -t d2xx || exit 0"
```

Similar rules can also be created to unbind the ftdi_sio driver from a given channel of a device. This can be beneficial if a user wishes to utilize both the D2XX and VCP drivers simultaneously. For example the following rule unbinds Channel A of an FT4232H, leaving Channels B-D to create a COM port.

```
# Rule to unbind the ftdi_sio driver from Channel A of a FT4232H MiniModule
ACTION=="add", SUBSYSTEMS=="usb", ATTRS{idVendor}=="0403",
ATTRS{interface}=="FT4232H MiniModule", ATTRS{bInterfaceNumber}=="00",
RUN+="/bin/sh -c 'echo $kernel > /sys/bus/usb/drivers/ftdi_sio/unbind'"
```

Finally, support for devices not included by default in the ftdi_sio driver can be added via a rule similar to the following, in this example the device uses a default FTDI VendorID (0403) and a custom ProductId (FAC6):

```
# Rule to add support to ftdi_sio for device VID: 0403, PID: FAC6
ACTION=="add", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="FAC6",
RUN+="/sbin/modprobe ftdi_sio" RUN+="/bin/sh -c 'echo 0403 FAC6 > /sys/bus/usb-
serial/drivers/ftdi_sio/new_id'"
```


3 Conclusion

This application note has shown how a udev rule can be created to easily map allocated COM ports on Linux operations systems to a consistent entry so that they are easier to identify for end users.

4 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited
(Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan, R.O.C.
Tel: +886 (0) 2 8797 1330
Fax: +886 (0) 2 8751 9737

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com

Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited
(USA)
7130 SW Fir Loop
Tigard, OR 97223-8160
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited
(China)
Room 1103, No. 666 West Huaihai Road,
Shanghai, 200052
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com

Web Site

<http://ftdichip.com>

Distributor and Sales Representatives

Please visit the Sales Network page of the [FTDI Web site](#) for the contact details of our distributor(s) and sales representative(s) in your country.

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Future Technology Devices International Ltd (FTDI) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640

Appendix A – References

Document References

[FT4232H Mini-Module](#) Development Module

[US232R-100](#) Cable

http://reactivated.net/writing_udev_rules.html

Acronyms and Abbreviations

Terms	Description
ftdi_sio	FTDI Virtual Com Port driver for Linux Operating Systems
sudo	Linux "Substitute User" command, gives administrator privileges
kernel	Code that forms the heart of a Linux Operating System
dmesg	Linux "Message of the day" – shows a log of events for the Linux PC
udev	Linux device manager subsystem
RS422/RS485/RS232	Standardized serial communication protocols
USB	Universal Serial Bus
VCP	Virtual COM Port (driver)
Terminal	A Linux application similar to Windows' "Command Prompt"

Appendix B – List of Tables & Figures

List of Figures

Figure 1 - dmesg output after enumeration	4
Figure 2 - Assigned COM ports for attached devices	4
Figure 3 - udevadm info output for /dev/ttyUSB0.....	5
Figure 4 – Device mapping after the udev rules have been created	7

Appendix C – Revision History

Document Title: AN_552 Consistent COM Port Enumeration on Linux
Document Reference No.: FT_001496
Clearance No.: FTDI#583
Product Page: <https://www.ftdichip.com/>
Document Feedback: [Send Feedback](#)

Revision	Changes	Date
1.0	Initial Release	03-08-2023