get.vial.today /manual/linux-udev.html

6 - Configuring udev rules for VIA and Vial on Linux

1. User manual
2. Configuring udev on Linux

In order for your keyboard to be detected by the VIA and Vial apps on Linux, you need to set up a custom `udev` rule. Both use the `hidraw` linux driver to directly communicate configurations to your keyboard. In most situations it should be safe to use a generalized `udev` rule for all `hidraw` devices, but in other cases where you prefer or are required to restrict access on a device basis, it is possible to create a `udev` rule specifically for your keyboard.

Vial has a magic number that is inserted into the USB serial attribute, so you will only need one rule for all Vial keyboards that will be safe to use in conjunction with other `hidraw` device rules.

The following guides will show you how to implement these `udev` rules. You will require root access to create the files that are necessary.

Universal Vial `udev` rule

For a universal access rule for any device with Vial firmware, run this in your shell while logged in as your user (this will only work with `sudo` installed):

```
export USER_GID=`id -g`; sudo --preserve-env=USER_GID sh -c 'echo "KERNEL==\"hidraw*\", SUBSYSTEM==\"hidraw\", ATTRS{serial}==\"*vial:f64c2b3c*\", MODE=\"0660\", GROUP=\"$USER_GID\", TAG+=\"uaccess\", TAG+=\"udev-acl\"" > /etc/udev/rules.d/99-vial.rules && udevadm control --reload && udevadm trigger'
```

This command will automatically create a `udev` rule and reload the `udev` system.

Manually

Write this text to `/etc/udev/rules.d/99-vial.rules` in a text editor:

```
KERNEL=="hidraw*", SUBSYSTEM=="hidraw", ATTRS{serial}=="*vial:f64c2b3c*", MODE="0660", GROUP="users", TAG+="uaccess", TAG+="udev-acl"
```

You can replace the group with any that you require on your operating system, but the `users` group should work on most distributions of Linux.

In order for the rule to take effect, you must reload `udev`.

Generalized VIA `udev` rule

On a kernel level, there is no way to detect if a keyboard is compatible with VIA, so you have to allow access to all `hidraw` devices. If you require more device security, you are able to configure access on a device-specific basis.

For a rule that allows users to access all `hidraw` devices, run this in your shell while logged in as your user (this will only work with `sudo` installed):

```
export USER_GID=`id -g`; sudo --preserve-env=USER_GID sh -c 'echo
"KERNEL==\"hidraw*\", SUBSYSTEM==\"hidraw\", MODE=\"0660\",
GROUP=\"$USER_GID\", TAG+=\"uaccess\", TAG+=\"udev-acl\"" >
/etc/udev/rules.d/92-viia.rules && udevadm control --reload && udevadm
trigger'
```

This command will automatically create a `udev` rule and reload the `udev` system.

Manually

Write this text to `/etc/udev/rules.d/92-viia.rules` in a text editor:

```
KERNEL=="hidraw*", SUBSYSTEM=="hidraw", MODE="0660", GROUP="users",
TAG+="uaccess", TAG+="udev-acl"
```

You can replace the group with any that you require on your operating system, but the `users` group should work on most distributions of Linux.

In order for the rule to take effect, you must reload `udev`.

Device-specific `udev` rules

Finding the vendor and product IDs

Configuring device specific `udev` rules will require knowing the USB vendor and product IDs of your keyboard. The easiest way to find them is using the `lsusb` command. Most major Linux distributions include `lsusb`, but you may need to install it. `lsusb` will list information about all of your connected USB devices. Here is a sample output for the Keychron Q2 Revision 0110:

```
Bus 001 Device 049: ID 3434:0110 Keychron Keychron Q2
```

The first four hex numbers next to `ID` is the vendor ID, and the next four hex numbers after the colon is the product ID. In this case, the Keychron Q2's vendor ID is `3434` and it's product ID is `0110`.

Where is my keyboard in `lsusb`?

Depending on your keyboard, the device name may be missing, or may not be the name of your keyboard at all. This is due to the fact that a majority of QMK compatible keyboards are not made by major companies with vendor IDs registered with the USB-IF. As a result, the IDs are sometimes set to arbitrary or unofficial values, or use the IDs supplied by the microcontroller's USB interface.

To find out which device is your keyboard, run `lsusb` with your keyboard plugged in and then run it again while disconnected. You should be able to find the different device between the two outputs.

Creating the `udev` rule

To create a rule for your specific keyboard, use this template:

```
# Name of your keyboard
KERNEL=="hidraw*", SUBSYSTEM=="hidraw", ATTRS{idVendor}=="XXXX",
ATTRS{idProduct}=="XXXX", MODE="0660", GROUP="users", TAG+="uaccess",
TAG+="udev-acl"
```

Replace the vendor and product ID with the ones you found for your keyboard using lsusb. You may also specify a different group if required. Write this rule to `/etc/udev/rules.d/99-vial.rules` as root. In order for the rule to take effect, you must reload `udev`.

The following is an example `udev` rule for the Keychron Q2:

```
# Keychron Q2
KERNEL=="hidraw*", SUBSYSTEM=="hidraw", ATTRS{idVendor}=="3434",
ATTRS{idProduct}=="0110", MODE="0660", GROUP="users", TAG+="uaccess",
TAG+="udev-acl"
```

Reloading `udev`

In order for your new `udev` rules to take effect, you must reload the `udev` rules. The standard way to do this is to run the following command as root:

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

If this command doesn’t work, or it’s not available, check that you are root or using a privilege escalation tool like `sudo`, and make sure to read your distribution’s handbook/manual.