# Инструкция по установке драйверов в Linux для настольных считывателей и конвертеров IronLogic

Поддерживаемые модели считывателей и конвертеров:
– Z-2 (мод. RD_ALL)
– Z-2 (мод. RD_ALL) (с режимом «Клавиатура»)
– Z-2 (мод. MF)
– Z-2 (мод. MF-I)
– Z-2 (мод. E HTZ RF)
–Z-397 (мод. USB Guard)
– Z-397 (мод. USB)
Прошивки в считывателях и конвертерах должны быть только заводских версий.
Основное внимание уделено Astra Linux, но также актуально для Ubuntu и Linux Mint.

## Установка и настройка FTDI устройств

Считыватели и конвертеры с чипом FTDI (Z-2 (мод. RD_ALL), Z-2 (мод. MF), Z-2 (мод. E HTZ RF), Z-397 (мод. USB Guard), Z-397 (мод. USB)) должны иметь VID/PID по умолчанию для FTDI (VID 0x403, PID 0x6001) и должны иметь строку описания "Manufacturer" равную "ILogic" или "IL".

Считыватель (конвертер) IronLogic в Astra Linux работает через преобразователь FTDI, поэтому для его настройки обычно требуется загрузить модуль ядра ftdi_sio и указать системе идентификаторы устройства (VID и PID).

### Основные этапы установки

1. **Подключение устройства и получение информации о нём.**

Подключите считыватель к USB-порту и выполните команду:

```
lsusb
```

В выводе должна отобразиться информация о вендор-ID и продукт-ID идентификаторах (idVendor и idProduct). Например: **0403:6001** (Vendor ID: 0403, Product ID: 6001)

Сообщение в терминале:

```
Bus 002 Device 002: ID 8087:8001 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:8009 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 011: ID 046d:c050 Logitech, Inc. RX 250 Optical Mouse
Bus 003 Device 016: ID 0403:6001 Future Technology Devices International, Ltd IronLogic RFID Adapter [Z-2 USB]
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

2. **Проверка и загрузка модуля ядра ftdi_sio.**

Драйвер ftdi_sio обычно встроен в ядро. Проверьте, загружен ли он командой:

```
lsmod | grep ftdi_sio
```

Если нет, отключаем ридер и добавляем модуль ядра ftdi_sio командой:

```
sudo modprobe ftdi_sio
```

Если устройство не появилось, привяжите его вручную по VID/PID (узнайте их из lsusb):

```
# Пример для VID 0403 и PID 6001
sudo modprobe ftdi_sio
echo "0403 6001" | sudo tee /sys/bus/usb-serial/drivers/ftdi_sio/new_id
```

3. **Указание системе, что устройство является FTDI.**

В Ubuntu (Astra Linux) для этого выполните команду:

```
sudo bash -c "echo 0403 6001 > /sys/bus/usb-serial/drivers/ftdi_sio/new_id"
```

Здесь 0403 — идентификатор вендора, 6001 — идентификатор продукта.

**4. Проверка подключения к системе и привязки драйвера.**

Подключите устройство и выполните команду, чтобы убедиться, что система его обнаружила:

```
sudo dmesg | tail
```

Сообщение в терминале:

```
[793134.996159] usb 3-1: new full-speed USB device number 20 using xhci_hcd
[793135.131301] usb 3-1: New USB device found, idVendor=0403, idProduct=6001
[793135.131303] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[793135.131305] usb 3-1: Product: USB IronLogic RFID Adapter
[793135.131306] usb 3-1: Manufacturer: ILogic
[793135.131307] usb 3-1: SerialNumber: IL02KQQ9
[793135.135463] ftdi_sio 3-1:1.0: FTDI USB Serial Device converter detected
[793135.135502] usb 3-1: Detected FT232RL
[793135.135641] usb 3-1: FTDI USB Serial Device converter now attached to ttyUSB0
```

Также можно проверить, что драйвер привязался к устройству, с помощью команды:

```
usb-devices | grep ftdi
```

В терминале отобразится:

```
I:  If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=ftdi_sio
```

**5. Настройка прав доступа.**

Для работы с USB считывателями дайте доступ к порту. По умолчанию к устройствам /dev/tty* имеют доступ пользователи из группы dialout. В этом можно убедиться, набрав:

```
ls -l /dev/ttyUSB0
```

В ответ появится в терминале:

```
crw-rw---- 1 root dialout 188, 0 фев 25 20:56 /dev/ttyUSB0
```

Откуда можно увидеть, что доступ открыт на чтение и запись для пользователя root и для группы dialout. Добавьте себя в эту группу:

```
sudo usermod -a -G dialout $USER
sudo reboot
```

где $USER - имя пользователя, например nick.

Перезагружать систему не обязательно, достаточно выйти из системы и снова войти.

**6. Для работы в режиме набора на клавиатуре нужен доступ /dev/uinput.**

Чтобы дать доступ:

– Создайте новую группу (если таковой ещё не существует, вы можете проверить это с помощью getent group uinput).

```
sudo groupadd uinput
```

– Добавьте своего пользователя в новую группу.

```
sudo usermod -aG uinput $USER
```

где $USER - имя пользователя.

– Создайте файл правил udev для установки разрешений. Создайте новый файл в /etc/udev/rules.d, например, /etc/udev/rules.d/99-uinput.rules, со следующим содержимым:

```
KERNEL=="uinput", MODE="0660", GROUP="uinput", OPTIONS+="static_node=uinput"
```

Создать файл можно командой:

```
sudo nano /etc/udev/rules.d/99-uinput.rules
```

◦ MODE="0660" предоставляет доступ на чтение/запись владельцу (root) и указанной группе.

◦ ОПЦИИ+="static_node=uinput" часто необходимы для обеспечения правильного управления узлом в разных системах.

– Перезагрузите правила udev и инициируйте изменения.

```
sudo udevadm control --reload-rules
sudo udevadm trigger --subsystem-match=misc
```

– Выйдите из системы и снова войдите в систему, чтобы изменения в группе вступили в силу.

**7. Настройка параметров порта.**

Установите скорость передачи и прочие параметры порта командой:

```
stty -F /dev/ttyUSB0 cs8 cstopb -ixon raw speed 9600
```

**8. Проверка считывания данных со считывателя.**

Выполните команду:

```
cat /dev/ttyUSB0
```

Сообщение в терминале:

```
Em-Marine[0000] 125,59903
No card
```

**9. Сохранение настроек и прав доступа после перезагрузки.**

Чтобы пользователь мог работать с ридером, нужно настроить udev-правила. Нужно создать скрипт, который будет автоматически настраивать устройство. Например, можно использовать такой подход:

Создайте файл правил udev:

```
sudo nano /etc/udev/rules.d/99-z2-readers.rules
```

Вставьте строки (VID/PID скорректируйте под ваше устройство):

```
text
# Z-2 USB Reader
SUBSYSTEM=="usb", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001", MODE="0666",
GROUP="users"
```

Если устройство определяется как TTY (COM-порт), найдите его (dmesg | grep tty), скорее всего ttyUSB0, и добавьте правило для него:

```
text
KERNEL=="ttyUSB*", MODE="0666", GROUP="users"
```

После создания файла выполните:

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

Переподключите считыватель.

**Дополнительные рекомендации**

**Проверка групп пользователей.** Убедитесь, что пользователь добавлен в группы dialout и uinput, так как это может быть необходимо для работы с устройством.

Если возникают проблемы, проверьте последние сообщения ядра с помощью команды:

```
sudo dmesg | tail
```

## Установка и настройка считывателей Z-2 (мод. RD_ALL) (с режимом «Клавиатура») и Z-2 (мод. MF-I)

Считыватели **Z-2 (мод. RD-All)** (VID 3168/PID 0100) и **Z-2 (мод. MF-I)** (VID 3168/PID 0106) в вашей системе автоматически загружается драйвером ядра cdc_acm и создаёт файл последовательного порта /dev/ttyACM0. Это поведение по умолчанию для устройств, реализующих протокол USB CDC ACM (виртуальный COM-порт).

В зависимости от того, какое приложение вы планируете использовать, есть **два основных подхода**:

- **Использовать устройство как обычный последовательный порт** (через /dev/ttyACM0) — если ваша программа поддерживает работу с COM-портом.
- **Отвязать драйвер** cdc_acm **и использовать устройство через libusb** — если программа требует прямого доступа через libusb.

Ниже подробно описаны оба варианта с учётом особенностей **Astra Linux** (Special Edition или Common Edition).

**1. Проверка текущего состояния.**

Подключите устройство и выполните:

```
ls -l /dev/ttyACM*
```

Если присутствует /dev/ttyACM0 – драйвер cdc_acm активен.

Узнайте, какой драйвер привязан к устройству:

```
lsusb -t
```

Вы увидите что-то вроде:

```
text
/:  Bus 01.Port 1: Dev 1, Class=root_hub, ...
   |__ Port 2: Dev 3, If 0, Class=Communications, Driver=cdc_acm, ...
   |__ Port 2: Dev 3, If 1, Class=CDC Data, Driver=cdc_acm, ...
```

**2. Вариант 1 – работа через последовательный порт (ttyACM0).**

Если ваше приложение умеет работать с последовательным портом, **ничего дополнительно устанавливать не нужно**. Достаточно настроить права доступа к устройству.

– **Настройка прав доступа.**

Для Z-2 (мод. RD-All) создайте правило udev для автоматического назначения прав:

```
sudo nano /etc/udev/rules.d/99-z2-rdall.rules
```

Вставьте:

```
Z-2 RD-All reader
SUBSYSTEM=="tty", ATTRS{idVendor}=="3168", ATTRS{idProduct}=="0100", MODE="0666", GROUP="dialout"
```

Для Z-2 (мод. MF-I) создайте правило udev для автоматического назначения прав:

```
sudo nano /etc/udev/rules.d/99-z2-mfi.rules
```

Вставьте:

```
Z-2 MF-I reader
SUBSYSTEM=="tty", ATTRS{idVendor}=="3168", ATTRS{idProduct}=="0106", MODE="0666", GROUP="dialout"
```

Добавьте своего пользователя в группу dialout (если ещё не добавлены):

```
sudo usermod -a -G dialout $USER
```

где $USER - имя пользователя, например nick.

Перезагрузите правила udev и переподключите устройство:

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

Теперь вы можете открывать /dev/ttyACM0 с правами обычного пользователя.

– **Настройка параметров порта.**

Установите скорость передачи и прочие параметры порта командой:

```
stty -F /dev/ttyACM0 cs8 cstopb -ixon raw speed 9600
```

– **Проверка считывания данных со считывателя.**

Для проверки считывания идентификаторов выполните команду:

```
cat /dev/ttyACM0
```

Сообщение в терминале:

```
Em-Marine[7900] 108,19751
No card
```

**3. Вариант 2 – использование через libusb (отвязка от cdc_acm).**

Если программа требует libusb (например, обращается к устройству через VID/PID, а не через последовательный порт), необходимо **отвязать** драйвер cdc_acm, чтобы ядро не управляло устройством, и передать управление libusb.

– **Установка libusb.**

```
sudo apt update
sudo apt install libusb-1.0-0 libusb-1.0-0-dev
```

– **Ручная отвязка драйвера.**

Найдите путь к интерфейсу устройства в sysfs. Например:

```
ls /sys/bus/usb/devices/*/idVendor | xargs grep -l 3168
```

Это покажет каталог, например /sys/bus/usb/devices/1-2. Внутри него будут подкаталоги для интерфейсов (1-2:1.0, 1-2:1.1 и т.д.).

Отвяжите каждый интерфейс от драйвера cdc_acm:

```
echo -n "1-2:1.0" | sudo tee /sys/bus/usb/drivers/cdc_acm/unbind
echo -n "1-2:1.1" | sudo tee /sys/bus/usb/drivers/cdc_acm/unbind
```

(Замените 1-2:1.0 на реальные пути.)

После этого устройство исчезнет из /dev/ttyACM* и станет доступным для libusb.

– **Автоматическая отвязка через udev.**

Чтобы отвязка происходила при каждом подключении, дополните правило udev:

Для Z-2 (мод. RD-All) создайте или отредактируйте /etc/udev/rules.d/99-z2-rdall.rules:

```
Z-2 RD-All reader
ACTION=="add", SUBSYSTEM=="usb", ATTR{idVendor}=="3168", ATTR{idProduct}=="0100", RUN+="/bin/sh -c 'for iface in /sys/bus/usb/devices/$kernel/*/../driver; do echo -n $(basename $(dirname $iface)) > $iface/unbind; done'"
```

Для Z-2 (мод. MF-I) создайте или отредактируйте /etc/udev/rules.d/99-z2-mfi.rules:

```
Z-2 MF-I reader
ACTION=="add", SUBSYSTEM=="usb", ATTR{idVendor}=="3168", ATTR{idProduct}=="0106", RUN+="/bin/sh -c 'for iface in /sys/bus/usb/devices/$kernel/*/../driver; do echo -n $(basename $(dirname $iface)) > $iface/unbind; done'"
```

Перезагрузите правила udev и переподключите устройство:

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

– **Настройка прав доступа для libusb.**

Даже после отвязки драйвера для работы через libusb требуются права на USB-устройство. В том же правиле для Z-2 (мод. RD-All) /etc/udev/rules.d/99-z2-rdall.rules укажите:

```
text
SUBSYSTEM=="usb", ATTR{idVendor}=="3168", ATTR{idProduct}=="0100", MODE="0666", GROUP="plugdev"
```

В том же правиле для Z-2 (мод. MF-I) /etc/udev/rules.d/99-z2-mfi.rules укажите:

```
text
SUBSYSTEM=="usb", ATTR{idVendor}=="3168", ATTR{idProduct}=="0106", MODE="0666", GROUP="plugdev"
```

Добавьте пользователя в группу plugdev:

```
sudo usermod -a -G plugdev $USER
```

где $USER - имя пользователя, например nick.

– **Проверка доступности через libusb.**

Используйте утилиту lsusb -v -d 3168:0100 или lsusb -v -d 3168:0106 – она должна показывать полную информацию без ошибок доступа.

Если ваше приложение использует libusb, теперь оно должно видеть устройство.

**4. Особенности Astra Linux (мандатный доступ).**

Astra Linux Special Edition использует мандатное управление доступом (PARSEC). В этом случае простых прав udev может быть недостаточно. Чтобы приложение могло обращаться к устройству:

– Установить мандатные метки на файл устройства (например, через pdpl-file):

```
sudo pdpl-file 0:0 /dev/ttyACM0   # для последовательного порта
```

Для libusb доступ идёт через /dev/bus/usb/..., поэтому нужно применить метки к соответствующему файлу (или использовать контекст безопасности приложения).

– Если вы работаете в замкнутой среде, возможно, потребуется разрешить доступ к USB-устройству в политике мандатного доступа (через графическую утилиту fly-admin-smc или командой setfacl). Для libusb также может потребоваться запуск приложения с определённым мандатным уровнем.

– В случае проблем попробуйте временно отключить мандатный контроль для теста (не рекомендуется в боевой среде):

```
sudo setenforce 0   # если используется SELinux (в Astra может быть свой механизм)
```

**5. Что делать, если устройство работает как ttyACM0, но приложение требует libusb?**

Некоторые программы могут ожидать, что устройство будет не последовательным портом, а USB-устройством с libusb. Тогда отвязка cdc_acm обязательна.

Однако существуют и программы, которые умеют работать с ttyACM0 напрямую. В таком случае **не отвязывайте** драйвер, а просто используйте порт.

**6. Проверка работоспособности.**

После настройки выполните:

– Если используете последовательный порт:

```
cat /dev/ttyACM0
```

В терминале должны видеть сырые данные при активности устройства:

```
Mifare[F239D848] 057,55368 1K (0004,08)
No card
```

– Если используете libusb: запустите тестовую программу.

Если устройство не отвечает, проверьте, что ни одно другое приложение не держит порт открытым.

**Заключение**

В большинстве случаев для **Z-2 (мод. RD-All)** (VID 3168/PID 0100) и **Z-2 (мод. MF-I)** (VID 3168/PID 0106) в Astra Linux достаточно **использовать** /dev/ttyACM0 с правильными правами. Если же ваше ПО требует libusb, отвяжите драйвер cdc_acm с помощью udev или вручную. Учитывайте мандатные ограничения Astra Linux, при необходимости настраивая политики доступа.