

## Настройка lirc для управления сервером mpd (Debian, ядро 3.4.104)

Общая идеология связки "пульт ДУ - сервер mpd" такая (на примере отличного debian-сервера от slovenia, ядро 3.4.104 Wheezy <http://www.igorpecovnik.com/2013/12/24/cubietruck-debian-wheezy-sd-card-image/>):

- в системе живет виртуальный девайс, обрабатывающий событие прихода сигнала от нашего пульта
- получаем код от пульта
- по коду (цифровому) определяем привязанную к нему команду (в текстовом виде)
- передаем эту команду клиенту mpc, а тот транслирует ее серверу

*Дисклаймер.*

*Чтение данного опуса не освобождает юзера ушастого от чтения мануалов. Более того - чтение мануалов есть святая его (юзера) обязанность.*

### Чтение кодов пульта

#### 1.

1а) Лобовой вариант

Ищем наш девайс в `/proc/bus/input/devices`:

```
.....  
P: Phys = RemoteIR/input1  
.....
```

Значит, наш девайс - `/dev/input/event1`

*ЗЫ это - чисто конкретно мой девайс в моем железе-софте, у вас он может быть другим .*

(для общей проверки работоспособности можно запустить `evtest`)

1б) Вариант с `udev` (см. <http://wlinux.ru/post/2773/>)

Находим номер инпута для ИК-приемника

```
cat /proc/bus/input/devices
```

(Допустим, вернулось `input1`)

Теперь получаем все параметры устройства

```
udevadm info -a -p /sys/class/input/input1
```

Из параметров пригодится

```
ATTR{name}=="sunxi-ir"
```

Теперь можно писать правило для `udev` (`/etc/udev/rules.d/10-lirc.rules`):

```
KERNEL=="event*", ATTRS{name}=="sunxi-ir", SYMLINK="lirc0"
```

Теперь после загрузки будет создан виртуальный девайс `/dev/lirc0` и он будет привязан к DEVICE в конфиге `hardware.conf`

**2.** Ставим `lirc`, `mrc`. Настраиваем конфиг по 'железу', редактируя файл:  
`/etc/lirc/hardware.conf` (привожу только измененный блок)

```
...  
DRIVER="devinput"  
...  
DEVICE="/dev/input/event1" # event1 мы ранее уже определили (/dev/lirc0 для  
работы с udev)  
MODULES="sunxi-ir"
```

### Расшифровка протокола и привязка имен кнопок к ИК\_кодам

**3.** Скачиваем шаблон кодов для нашего пульта, для моего это `NEC.conf` - там важная инфа в заголовке по таймингам ИК-посылок. <http://lirc.sourceforge.net/remotes/> - шаблоны для очень многих пультов.

**4.** Запись в файл кодов кнопок, привязанных к именам кнопок. Для этого сначала нужно узнать, какие имена допускаются (т.е. получить т.н. "пространство имён"):

```
irrecord -l
```

Теперь можно писать коды ИК-посылок, привязывая их сразу к именам кнопок:

```
irrecord -H devinput -d /dev/input/event1 NEC.conf
```

(тут `event1` - мы его определили ранее, вместо `NEC.conf` - подставить свой шаблон)

Работа `irrecord` идет в режиме диалога - сначала запрашивается имя кнопки, которая будет нажата (см. "пространство имен"), потом нажимается эта кнопка.

После завершения получим выходной файл - он повторяет имя шаблона с дополнительным расширением `.conf` (т.е. в моем примере это будет файл `NEC.conf.conf`). Копируем его в `/etc/lirc/lircd.conf` и попутно удаляем вторые слова в кодах кнопок.

### Привязка собственно команд и действий.

**5.** Создаем файл `/etc/lirc/.lircrc` и прописываем его в `/etc/rc.local`:

```
irxexec -d /etc/lirc/.lircrc
```

Этот файл и есть связующее звено между названиями кнопок (которые, в свою очередь, связаны с кодами пульта файлом `lircd.conf`) и собственно командами серверу `mrcd`. Примеры командных блоков (см. мануалы по `mrc`, `mrcd`):

Простая однократная команда

```
begin
prog = irexec
button = KEY_ENTER
repeat = 0
config = mpc pause
end
```

Команда-переключатель (например, если одна кнопка используется как Пуск и Пауза)

```
begin
prog = irexec
button = KEY_PLAY
repeat = 0
config = mpc play &
config = mpc pause
end
```

Команда пошаговой перемотки

```
begin
prog = irexec
button = KEY_FASTFORWARD
repeat = 0
config = mpc seek +2%
end
```

*Понятное дело, что командами серверу `trd` данный функционал не ограничивается и так можно делать все что угодно - дергать ножками `subietruck`, исполнять любые программы и скрипты.*