Переделывание D-Link DIR-320 в 3G роутер

Cамая распространённая проблема при покупке модема заключается в том, что нет такого модема, который бы не зависал через некоторое время (8 часов).

Задача заключается в следующем - сделать небольшой стабильный 3G роутер.  
 Для этого нужно:

* Роутер DIR-320
* 3G модем
* Внешняя антенна

Для начала эксперимента воспользуемся стационарным ПК с Linux. Настроив 3G соединение, посредством Huawei E173, запустить вызов команды ping + wget с интервалом раз в 10 минут. Соединение длилось около 4 часов, после чего модем перестал отвечать на PPP команды, AT при этом работали.  
Попробуем потушить USB порт:

1. Находим модем из списка всех USB устройств:
2. for f in `ls /sys/bus/usb/devices/\*/product`; do echo -e "$f\t`cat $f`"; done
3. Выключаем порт на 5 секунд а потом включаем:
4. echo “suspend” | sudo tee /sys/bus/usb/devices/1-1/power/level
5. sleep 5
6. echo "on" | sudo tee /sys/bus/usb/devices/1-1/power/level

В ходе эксперимента мы увидели, что все отлично перезагружается. И соединение может быть опять установлено.

Повторим эксперимент с четырьмя прошивками для роутера: D-Link, Asus, OpenWRT или DD-WRT, но ни в одной из прошивок не оказалось возможности управления питанием USB. И после перезагрузки роутера модем оставался проинициализированным.

Далее разберем роутер.  
Нужно сделать модернизацию платы и источника питания.  
  
Нужно построить дистрибутив на базе OpenWRT самостоятельно, при этом с минимумом предустановленных пакетов и сервисов, так как оперативная память роутера маленькая. Но для начала нужно выбрать целевой модем/оператора и внешней антенны.  
  
Наиболее подходящим является MTS. У них можно всю статистику и управление совершить из удобного WEB интерфейса.

* 3G модем
* Направленная антенна CDMA-450

Попытаемся подключиться к этому модему. Он отображается как CD-ROM.

Анализ dmesg показал, что CD-ROM монтируется как SCSI, и его нужно отключать именно как SCSI, никакие usb\_modeswitch тут не работают.

Первый же проход по папке с man’ами показал, что есть пакет sdparm.   
Далее приступаем к работе.

Сборка OpenWRT

Все шаги выполняются в консоли Linux. Обратите внимание, релиз Backfire выбран из-за того, что это последний релиз с ядром 2.4. Это очень важно, так как ядро 2.6 само по себе очень требовательно как к производительности процессора, так и к системным ресурсам.

* Скачиваем исходник OpenWRT:
* svn co svn://svn.openwrt.org/openwrt/branches/backfire openwrt-backfire
* Переходим в папку с исходником
* cd openwrt-backfire
* Обновляем все пакеты
* ./scripts/feeds update -a
* Добавляем из пакетов sdparm
* ./scripts/feeds install sdparm
* Открываем меню настройки
* make menuconfig
* Выбираем интересующую платформу
* Target System (Broadcom BCM947xx/953xx [2.4])
* Встраиваем интересующие нас пакеты (по умолчанию они не включены даже как модули)
* <\*> sdparm........................ Read or modify SCSI or USB disk parameters
* <\*> chat................................. Establish conversation with a modem
* <\*> comgt............................... Option/Vodafone 3G/GPRS control tool
* <\*> kmod-scsi-generic........................ Kernel support for SCSI generic
* <\*> kmod-usb-core............................................ Support for USB
* <\*> kmod-usb-ohci............................... Support for OHCI controllers
* <\*> kmod-usb-serial..................... Support for USB-to-Serial converters
* <\*> kmod-usb-storage..................................... USB Storage support
* <\*> kmod-usb2................................... Support for USB2 controllers
* Все остальные пункты оставляем без изменения
* Выходим из меню конфигуратора с сохранением
* Собираем инструменты
* make tools/install -j`grep -c processor /proc/cpuinfo`
* make toolchain/install -j`grep -c processor /proc/cpuinfo`
* Открываем меню настройки ядра Linux
* make kernel\_menuconfig
* Настраиваем MTD
* [\*] Support 2-chip flash interleave
* Выходим из меню конфигуратора с сохранением
* Собираем OpenWRT
* make -j`grep -c processor /proc/cpuinfo`
* После того как прошивка собрана, её нужно прошить в устройство.

Проверка прошивки

После прошивки и перезагрузки роутера подключите к нему модем и соедините роутер посредством Ethernet с вашим ПК. Настройте на интерфейсе вашего ПК IP адрес, выполнив команду:

ifconfig eth0 inet 192.168.1.2/24

Потом подключитесь к роутеру по Telnet. Для этого из консоли выполните следующую команду:

telnet -l root 192.168.1.1

Вход будет совершен автоматически без запроса пароля. Это в корне не верно. Чтобы исправить ситуацию, задайте пароль, выполнив команду в консоли:

passwd

Выполните команду «*exit*», для выхода из **telnet** сессии.  
Выполните вход по **ssh**:

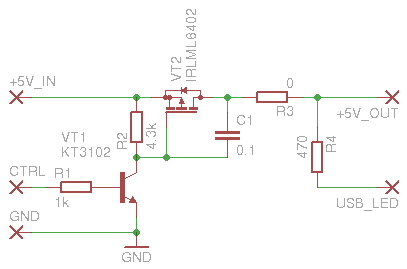
ssh root@192.168.1.1

Далее, удостоверимся, что модем был распознан ядром, выполнив команду в консоли:

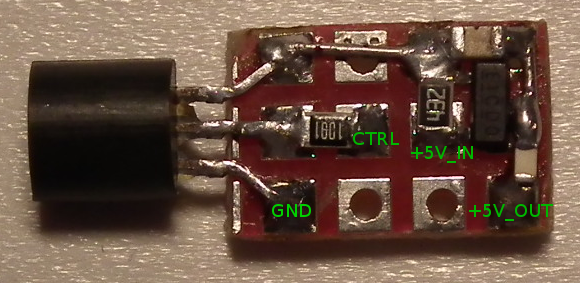
grep "^P\:.\*22de.\*6803" /proc/bus/usb/devices

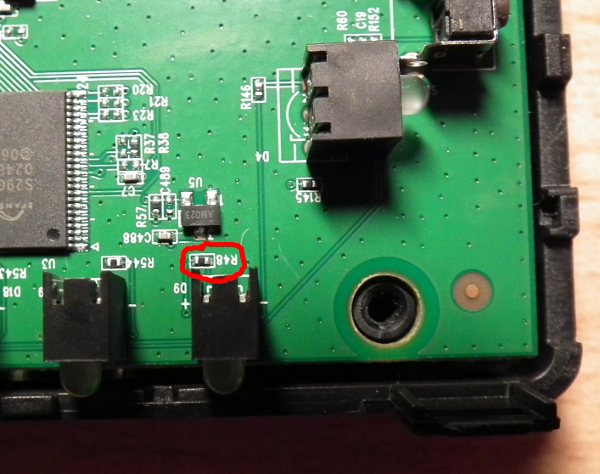
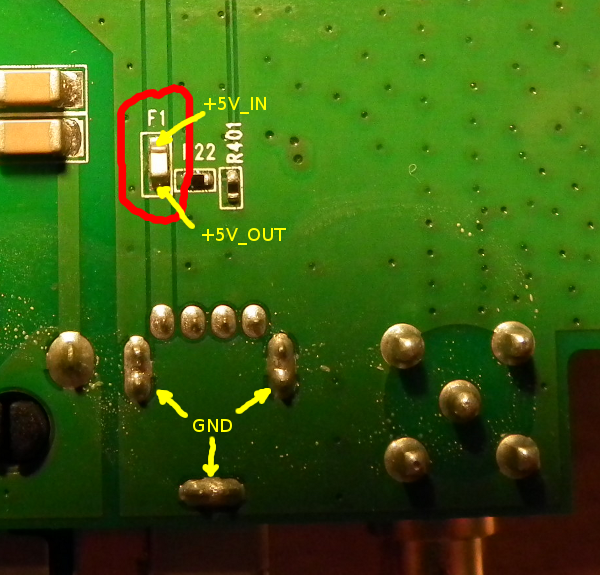
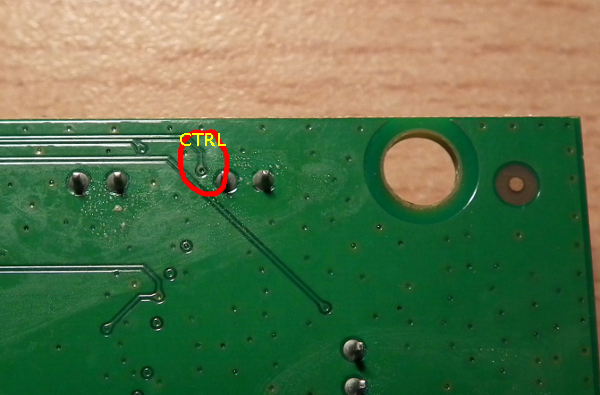
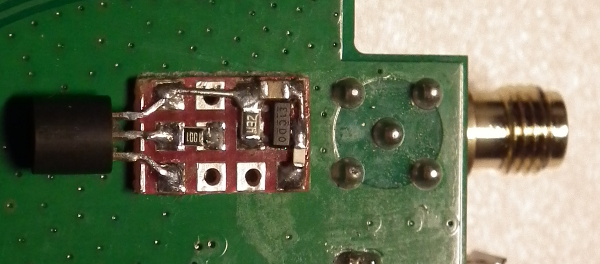
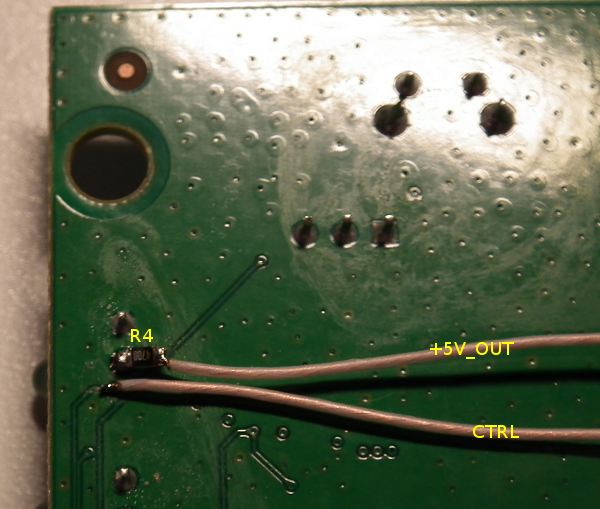
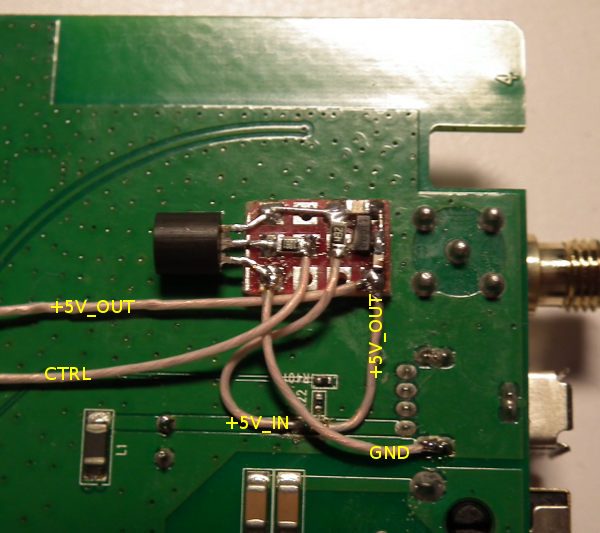
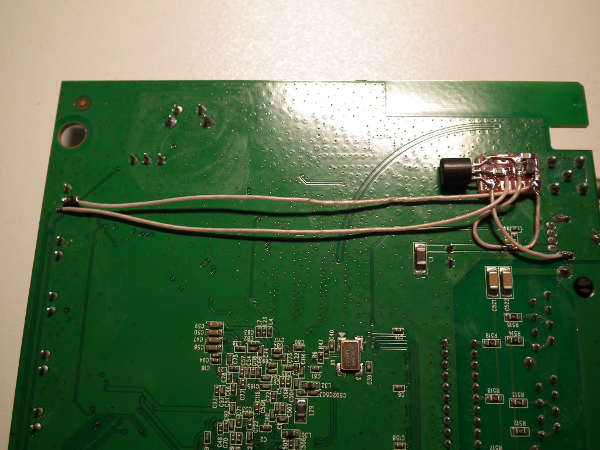
Результатом выполнения этой команды, будет строка с VID и PID модема. В противном случае, проверьте, что было сделано не по инструкции.

Модернизация аппаратного обеспечения роутера

Для управления питанием USB порта нужна свободная ножка процессора. У этого роутера масса светодиодных индикаторов на передней панели и один из них можем приспособить под наши нужды. Выбираем пал на индикатор со значком принтера, так как он соответствует светодиоду USB на proc-fs (/proc/diag/led/usb).  
  
Управление питанием будет осуществлять дополнительная плата с минимально необходимым набором деталей. Вот её схема:  
  
Оба транзистора работают в качестве ключей. При подаче логической единицы на вход «CTRL», транзистор VT1 откроется и далее также откроется транзистор VT2 и питающее напряжение подастся на USB порт.   
Не обязательно использовать печатную плату, можно – монтажную.

Вот так выглядит конечный вариант:

  
*Зелёными надписями отмечены одноименные порты на схемы. Резистор R4, припаян непосредственно к ножке светодиода.*  
  
Пошаговая инструкция по модификации:

* Обесточьте роутер и разберите его
* Уберите не нужный R48 – этот резистор соединяет светодиод с ножкой процессора  
  
* Удалите предохранитель F1 – этот предохранитель, должен обеспечить защиту в случае короткого замыкания на линиях питания USB.
* Надписями желтого цвета отмечены места подключения платы управления  
  
* Удалите защитное покрытие с проводника, ведущего сигнал от процессора к резистору R48. К этому проводнику, должен быть подпаян порт «CTRL» платы управления  
  
* Приклейте плату управления в наиболее удобное для вас место.   
  
* Припаяйте резистор R4 непосредственно к аноду светодиода, как указанно на фотографии. Также обратите внимание на подключение соединительных проводов  
  
* Закончите монтаж соединительных проводов  
  
* Итоговый результат  
  
* Соберите и включите роутер

Модернизация программного обеспечения роутера

Подключитесь к роутеру по ssh, и все следующие шаги будут выполняться на нём.

Создание утилит для управления USB

Перейдите в папку /etc/ppp.

cd /etc/ppp

* Создайте usb-up.sh. Этот скрипт будет включать питание на USB порте
* $ cat usb-up.sh
* #!/bin/sh
* logger -t "USB PwrCtrl" "Turn-on USB power"
* echo 1 > /proc/diag/led/usb
* Создайте скрипт для перезагрузки USB порта
* $ cat resetusb.sh
* #!/bin/sh
* logger -t "USB PwrCtrl" "Turn-off USB power"
* echo 0 > /proc/diag/led/usb
* # Sleep in 3 sec is an optional, but may be useful for some modem types
* #sleep 3
* logger -t "USB PwrCtrl" "Turn-on USB power"
* echo 1 > /proc/diag/led/usb
* Создайте скрипт для проверки существования процесса pppd
* #!/bin/sh
* if ps | grep -q /usr/sbin/pppd; then
* logger -t "PPP watcher" "PPP alive"
* else
* logger -t "PPP watcher" "No PPP daemon"
* /etc/ppp/resetusb.sh
* fi
* Сделайте скрипты исполняемыми
* chmod a+x /etc/ppp/\*.sh
* Добавим корректную обработку подключения модема к USB. Обратите внимание на то, что в OpenWRT, по умолчанию, нет udev, а вместо этого используется hotplug «демон»
* $ cat /etc/hotplug.d/usb/11-modem
* #!/bin/sh
* # Copyright (C) 2013 Sergey Shcherbakov <shchers@gmail.com>
* case "$ACTION" in
* add)
* (grep -q "^P\:.\*22de.\*6803" /proc/bus/usb/devices) && {
* logger -t "WM-D300" "Detected ZeroCD of WM-D300 modem"
* # Checking ZeroCD status
* while [ "$(sdparm -q --command=ready /dev/scsi/host0/bus0/target0/lun0/generic)" != "Ready" ]; do
* logger -t "WM-D300" "ZeroCD yet not ready"
* # Workaround for bug with multiple calls of scripts
* (/usr/bin/lsusb | grep -q 22de:6801) && {
* logger -t "WM-D300" "Seems that script called twice and device already initialized"
* return 0
* }
* sleep 2
* done
* logger -t "WM-D300" "ZeroCD ready and will be ejected"
* /usr/bin/sdparm --command=eject /dev/scsi/host0/bus0/target0/lun0/generic
* }
* (grep -q "^P\:.\*22de.\*6801" /proc/bus/usb/devices) && [ ! -c /dev/usb/tts/0 ] && {
* logger -t "WM-D300" "Updating USB Serial driver with D200/D300 vendor and product ID"
* rmmod usbserial && insmod usbserial vendor=0x22de product=0x6801
* }
* (grep -q "^P\:.\*22de.\*6801" /proc/bus/usb/devices) && [ -c /dev/usb/tts/2 ] && ! (ifconfig 3g-wan 1>/dev/null 2>/dev/null) && {
* logger -t "WM-D300" "All modem interfaces installed, turning on WAN"
* ifdown wan && ifup wan
* }
* ;;
* remove)
* ! (grep -q "^P\:.\*22de.\*6801" /proc/bus/usb/devices) && {
* logger -t "WM-D300" "Modem disconnected, turning off WAN interface"
* ifdown wan
* }
* ;;
* esac

Внесение изменений в существующие скрипты

* Запретите WAN интерфейс, заданный по умолчанию, выполнив команду
* sed -i 's/interface\ wan$/interface\ wan\_dflt/' /etc/config/network
* Добавьте новый описатель WAN интерфейса с [настройками для 3G](http://wiki.openwrt.org/doc/uci/network#protocol.3g.ppp.over.ev-do.cdma.umts.or.gprs)
* $ cat /etc/config/network
* …
* config interface wan
* option ifname ppp0
* option device /dev/usb/tts/0
* option service evdo
* option proto 3g
* option username 'mobile'
* option password 'internet'
* option pppd\_options 'noipdefault maxfail 3'
* option connect '/etc/ppp/usb-up.sh'
* option disconnect '/etc/ppp/resetusb.sh'
* option keepalive 20
* Для того чтобы добавить автоматическое включение USB порта при инициализации роутера, нужно модифицировать стандартный скрипт /etc/init.d/usb, добавив в конце функции start() вызов скрипта /etc/ppp/usb-up.sh
* $ cat /etc/init.d/usb
* …
* start() {
* ...
* # Turn-on power on USB
* /etc/ppp/usb-up.sh
* }
* …
* Дополнительно можно модифицировать, но необязательно, Chat скрипт (/etc/chatscripts/evdo.chat).
* ABORT BUSY
* ABORT 'NO CARRIER'
* ABORT ERROR
* ABORT 'NO DIAL TONE'
* ABORT 'NO DIALTONE'
* ABORT 'NO ANSWER'
* REPORT CONNECT
* TIMEOUT 10
* '' AT
* OK ATDT#777
* CONNECT ''
* Добавляем в Cron задачу, которая будет проверять наличие процесса pppd каждые пять минут. Для этого следует вызвать команду “crontab -e” и добавить следующую строку:
* \*/5 \* \* \* \* /etc/ppp/check.sh
* Разрешаем выполнение Cron и запускаем его
* /etc/init.d/cron enable && /etc/init.d/cron start

Далее перезагрузить роутер и можете пользоваться с удовольствием.  
Без сбоя работал 41 день. Это гораздо больше, чем у обычных модемов. В зависимости от местоположения скорость достигала от 220мс до 350Кбит/с.