

Instalación y Configuración de Módem interno TP-LINK (TM-IP5600/TM-IA5629V) Motorola SM56 Data Fax Módem

Introducción

En la siguiente guía se muestra la instalación y configuración de un módem en el sistema **NOVA 1.1.2 BAIRE** . Se podrá apreciar como se encuentra disponibles los módulos necesarios para el módem a instalar en NOVA. Pueden contactar mediante los correos electrónicos del autor para establecer mejoras, críticas constructivas, opiniones y aprender sobre las experiencias de ustedes.

Autor

Oleksis Fraga Menéndez



Universidad de Matanzas “Camilo Cienfuegos” – Facultad de Ingeniería Informática – Grupo @tis
emails: oleksis.fraga@umcc.cu, oleksis.fraga@gmail.com

Desarrollo

Para la instalación de un módem en un sistema **GNU/Linux** que no tenga soporte para determinados winmodems, se recomienda visitar la página [Linmodem support](#) . Allí encontraras utilidades como [scanModem](#) y drivers compatibles con su winmodem. En este caso el módem a instalar es de la familia SmartLink y **NOVA** posee los drivers (ungrab-winmodem y slamr) y servicios (slmodem) para su correcto funcionamiento. Solo nos hace falta software adicional como los siguientes:

Requisitos

- ppp (ppp-2.4.4)
- openssl (openssl-0.9.8j)
- wvstreams (wvstreams-4.5)
- wvdial (wvdial-1.60)

Instalación de ppp

1. descomprimir el archivo ppp-2.4.4.tar.gz
2. tar -xvf ppp-2.4.4.tar.gz
3. cd ppp-2.4.4

4. ./configure
5. make
6. make install

Instalación de openssl

1. tar -xvf openssl-0.9.8j.tar.gz
2. cd openssl-0.9.8j
3. ./config
4. make
5. make test
6. make install

Instalación de wvstreams

1. tar -xvf wvstreams-4.5.tar.gz
2. cd wvstreams-4.5
3. ./configure
4. make
5. configurar el directorio para librerías (defecto : /usr/local/lib) a /usr/lib en el fichero config.mk establecer libdir=/usr/lib
6. make install

Instalación de wvdial

1. tar -xvf wvdial-1.60.tar.gz
2. cd wvdial-1.60
3. export PKG_CONFIG_PATH=./wvstreams-4.5/pkgconfig ó
export PKG_CONFIG_PATH=/usr/lib/pkgconfig
4. make
5. make install

Configurar wvdial

/usr/local/bin/wvdialconf /etc/wvdial.conf

Salida similar a la siguiente detectando el módem:
Editing `/etc/wvdial.conf'.

Scanning your serial ports for a modem.

```
ttyS0<*>: ATQ0 V1 E1 -- failed with 2400 baud, next try: 9600 baud
ttyS0<*>: ATQ0 V1 E1 -- failed with 9600 baud, next try: 115200 baud
ttyS0<*>: ATQ0 V1 E1 -- and failed too at 115200, giving up.
ttyS1<*>: ATQ0 V1 E1 -- failed with 2400 baud, next try: 9600 baud
ttyS1<*>: ATQ0 V1 E1 -- failed with 9600 baud, next try: 115200 baud
ttyS1<*>: ATQ0 V1 E1 -- and failed too at 115200, giving up.
Modem Port Scan<*>: S2 S3
WvModem<*>: Cannot get information for serial port.
ttySL0<*>: ATQ0 V1 E1 -- OK
ttySL0<*>: ATQ0 V1 E1 Z -- OK
ttySL0<*>: ATQ0 V1 E1 S0=0 -- OK
ttySL0<*>: ATQ0 V1 E1 S0=0 &C1 -- OK
ttySL0<*>: ATQ0 V1 E1 S0=0 &C1 &D2 -- OK
ttySL0<*>: ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0 -- OK
ttySL0<*>: Modem Identifier: ATI -- SmartLink Soft Modem
ttySL0<*>: Speed 4800: AT -- OK
ttySL0<*>: Speed 9600: AT -- OK
ttySL0<*>: Speed 19200: AT -- OK
ttySL0<*>: Speed 38400: AT -- OK
ttySL0<*>: Speed 57600: AT -- OK
ttySL0<*>: Speed 115200: AT -- OK
ttySL0<*>: Speed 230400: AT -- OK
ttySL0<*>: Speed 460800: AT -- OK
ttySL0<*>: Max speed is 460800; that should be safe.
ttySL0<*>: ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0 -- OK
```

Found a modem on /dev/ttySL0.

/etc/wvdial.conf<Warn>: Can't open '/etc/wvdial.conf' for reading: No such file or directory

/etc/wvdial.conf<Warn>: ...starting with blank configuration.

Modem configuration written to /etc/wvdial.conf.

ttySL0<Info>: Speed 460800; init "ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0"

Adaptando la configuración de /etc/wvdial.conf

[Dialer Defaults]

Modem = /dev/ttySL0

Baud = 460800

Init1 = ATZ

Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0

Modem Type = Analog Modem

PPPD Path = /usr/local/sbin/pppd

Phone = 123456

Username = username

Password = password

Ask Password = 0

ISDN = 0

```
Dial Command = ATDT
Stupid Mode = 1
Compuserve = 0
Auto DNS = 1
```

Cargamos los módulos en el kernel

```
modprobe ungrab-winmodem
modprobe slamr
```

Comprobamos con:

```
lsmod | grep ungrab
```

```
Salida > ungrab_winmodem      9600  0
```

```
lsmod | grep slamr
```

```
Salida > slamr                431272  0
```

Adaptando la configuración de /etc/conf.d/slmodem

Habilitar el módulo a cargar (MODULE = slamr) quedando así:

```
# Config file for /etc/init.d/slmodemd

DEV=/dev/ttySL0
COUNTRY=USA
GROUP=dialout
# the group should be kept in sync with:
# /etc/devfs.d/slmodem (devfs)
# /etc/udev/rules.d/55-slmodem.rules (udev)

# The following symlink will be created if uncommented
LN_DEV=/dev/modem

# Raise priority to reduce modem dropouts
NICE=-6

# ALSA Options:

# The following sets the ALSA (alsasound) init script to
# be a dependancy of the slmodem one. It does also provides
# ALSA support.
#MODULE=alsa

# The modem hardware slot
# use "modem:0", "modem:1", etc.
```

```

# usually modem:1 is used
HW_SLOT=modem:1

# Non-ALSA OPTIONS:

MODULE=slamr
# or
# MODULE=slusb

# Include extra slmodemd options here:
# -r = enables ring detection (needed for Hylafax faxgetty to answer)
# -l 5 = Logging Level
SLMODEM_OPTS=""

```

Para el correcto funcionamiento del servicio slmodem deben estar cargados los módulos ungrab_winmodem y slamr, por lo que se puede modificar el fichero de configuración del servicio en **/etc/init.d/slmodem** como sigue en los bloques de inicio (start) y parada (stop):

```

start() {

    checkconfig || return 1
    if [ ! -c /dev/ppp ]; then mknod /dev/ppp c 108 0; fi

    # either if we use alsa or not, the only thing we need is
    # ttySL0, which is created by slmodemd when started (points to a pts)

    if [ "${MODULE}" = "alsa" ]; then
        loadsalsa
    else
        modprobe ungrab-winmodem
        loadsmodule
    fi

    result=${?}

    test ! -z "${LN_DEV}" && /bin/ln -s ${DEV} ${LN_DEV} 2> /dev/null

    eend ${result}
}

stop() {
    ebegin "Shutting down slmodemd"
    start-stop-daemon --stop --quiet --pidfile /var/run/slmodemd.pid
    result=${?}
    [ -e /var/run/slmodemd.pid ] && rm /var/run/slmodemd.pid
    unlink ${LN_DEV} 2> /dev/null
    eend ${result}
    if [ ! "${MODULE}" = "alsa" ]; then

```

```

    ebegin "Waiting for ${MODULE} modem driver unload..."
    for a in $(seq 0 2); do
        sleep 1
        echo -n "."
        modprobe -r ungrab-winmodem 2> /dev/null && result=0
        modprobe -r slamr 2> /dev/null && result=0
    done
    if [ "$SRC_NOCOLOR" != "yes" ]; then echo; fi
    eend ${result}
fi
}

```

Iniciamos el servicio

```
/etc/init.d/slmodem start
```

```
Salida > * Starting slmodemd for /dev/ttySL0 ...          [ ok ]
```

Levantamos la conexión

```
/usr/local/bin/wvdial
```

para cancelar la conexión CTRL+C

Configuración del nivel de ejecución default

Agregamos el servicio al nivel de ejecución

```
rc-update add slmodem default
```

Comprobamos con:

```
rc-status
```

```
Salida > Runlevel: default
```

```

...
slmodem          [ started ]
...

```

Otras utilidades

Nos permite mostrar una lista de los módulos disponibles:

```
find /lib/modules/$(uname -r)/ -type f -iname '*.o' -or -iname '*.ko' | more
```

Conclusión

Pudimos observar el proceso de instalación y configuración de los módulos y software necesario para nuestro módem. El proceso con otros winmodems debe ser parecido. Antes recomiendo buscar toda la documentación posible de nuestro módem para lograr una mejor instalación. Espero que también colabores con la comunidad de NOVA a través de los foros en <http://www.nova.uci.cu/website/foros> .