

# WIS-GO Multimedia

Du möchtest dich gerne für unsere Hilfe erkenntlich zeigen 🙏 . Gerne. Wir bedanken uns bei dir für deine Spende! ☐

[Spenden](#)

Zum frei verfügbaren [Apt-Repository](#)



GITLAB:

Am besten ihr ladet euch einfach das fertige Image von meinem SF-Projekt herunter.

## Intro

SF Project: <https://sourceforge.net/projects/wis-go7007/>

WIS's Go7007 MPEG-4 encoder chip is used in many video encoder boxes including the Plector ConvertX (PX-M402U) which I own. The best part about this chip is it's fully supported under Linux if you have the appropriate driver (which I might add is open source!). To see what devices are supported visit this page: <http://oss.wischip.com>

The problem is that WIS hasn't updated their driver in a while so it won't work on most modern Linux distributions. Thanks to some talented programmers the driver will work with their patches. I've patched the wis-go7007-linux-0.9.8 source so it works on Fedora Core 5/6, Fedora 7/8/9 and Ubuntu Dapper/Edgy/Feisty/Gutsy/Hardy. I've also patched the gorecord program with an audio/video sync patch and most recently made it possible to record huge video files instead of being limited to 1GB recordings.

- All the patches are available in the **patches** directory of the source tarball

## Driver Versions

- [wis-go7007-linux-0.9.8-2](#)
  - Applied a patch from [here](#) to support linux >= 2.6.24
- [wis-go7007-linux-0.9.8-1](#)
  - New versions of the driver and gorecord.
    - Backwards compatibility fixes
    - No filesize limits for gorecord
    - Udev rules now work on Ubuntu
- [wis-go7007-linux-0.9.8-patched](#)
  - The first „patched“ release
    - 2.6.22 kernel support
    - Many patches to improve the overall driver package

- [wis-go7007-linux-0.9.8](#)
  - The initial driver package from Micronas USA

## Installation

Installation is a bit different for each distro so follow the instructions for your distro. You can probably just copy the commands as one big chunk into your terminal but it's better to do them one at a time.

### Fedora Core 5/6 and Fedora 7/8/9

```
# Install required development packages
su -c "yum install kernel-devel gcc ncurses-devel"
# Get fxload RPM because the Fedora repos don't carry this package
su -c "rpm -ivh
http://nikosapi.org/software/WIS_Go7007/fxload-2002_04_11-1.i386.rpm"
# Get the driver source
wget
http://nikosapi.org/software/WIS_Go7007/wis-go7007-linux-0.9.8-2.tar.bz2
# Unpack the source
tar -xjvf wis-go7007-linux-0.9.8-2.tar.bz2
# Decend into source directory
cd wis-go7007-linux-0.9.8-2
# Compile the software
make
# Install the software
su -c "make install"
```

### Ubuntu Dapper/Edgy/Feisty/Gutsy/Hardy

```
# Install required development packages (this is a big download)
sudo apt-get install linux-headers-generic fxload libncurses5-dev
# Get the driver source
wget
http://nikosapi.org/software/WIS_Go7007/wis-go7007-linux-0.9.8-2.tar.bz2
# Unpack the source
tar -xjvf wis-go7007-linux-0.9.8-2.tar.bz2
# Decend into source directory
cd wis-go7007-linux-0.9.8-2
# Compile the software
make
# Install the software
sudo make install
```

**Note:** Users of Ubuntu Feisty, Gutsy and Hardy have to follow the [instructions to enable usbfs](#) before the device will work.

## Ubuntu Hardy with kernel <= 2.6.46-17

The above instructions allow to compile the driver but it is not working. The driver crashed during the initialization of the ALSA sound input. For a detailed outline of the errors attempting to load the drivers, see this thread: <http://ubuntuforums.org/showthread.php?t=791842>.

However, recompiling the kernel, after verifying that the required drivers are included, leads to working setup on Ubuntu with kernel 2.6.24. You can, e.g., follow the [instructions by Benjamin Schleiner](#)

The exact cause is not clear, however:

- the wis-go7007 readme lists certain kernel configuration flags which have to be set and are not on the stock kernel, most notably CONFIG\_SND.
- the stock kernel provides ALSA 1.0.16 (cat /proc/asound/version.h), whereas the include files state that version 1.0.15 is used (more /usr/include/alsa/version.h).

### Other solution for Hardy 8.04 (maybe better?)

It is not necessary to recompile the kernel! CONFIG\_SND is not enabled in the kernel, but also driver and modules are provided via linux-ubuntu-modules (or LUM)

There's a simple workaround to compile the wis-go7007 driver: Here, to simplify, I suppose that the current kernel version is 2.6.24-19-generic. If not, replace everywhere needed.

Make sure the following packages are installed:

```
linux-headers-lum-2.6.24-19-generic
linux-headers-2.6.24-19-generic
```

Get rid of the incorrect sound kernel headers (which should never be included): for example, rename

```
/usr/src/linux-headers-2.6.24-19-generic/include/sound
```

to

```
/usr/src/linux-headers-2.6.24-19-generic/include/sound-blacklisted
```

In the wis-go7007 source, add the following line at the beginning of kernel/Makefile to take the right LUM also headers into account:

```
EXTRA_CFLAGS += -I /usr/src/linux-headers-lum-2.6.24-19-generic -include
sound/config.h
```

Compile and install normally.

# Testing

## Recording

Plug in your (\*supported\*) video capture device and try the following command:

- for the composite input use: -input 0
- for the svideo input use: -input 1

`gorecord -input 0 -duration 20 test-video.avi` This will record video for 20 seconds then stop. The video file will be named `test-video.avi`

## Playback

Playback is a bit tricky because the video is encoded to a so called „proprietary“ format (mpeg-4). Which basically means it's not encoded as theora or using some other non-proprietary video codec.

## Fedora

- Enable the Livna repository:

For FC5:

```
su -c "rpm -Uvh http://rpm.livna.org/livna-release5.rpm"
```

For FC6:

```
su -c "rpm -Uvh http://rpm.livna.org/livna-release-6.rpm"
```

For Fedora 7:

```
su -c "rpm -Uvh http://rpm.livna.org/livna-release-7.rpm"
```

For Fedora 8:

```
su -c "rpm -Uvh http://rpm.livna.org/livna-release-8.rpm"
```

- Install mplayer:

```
su -c „yum install mplayer“
```

- Play the video:

```
mplayer test-video.avi
```

## Ubuntu

Should be as easy to install as:

```
sudo apt-get install mplayer
```

Then play the video:

```
mplayer test-video.avi
```

## Fun

### GoGomini

I wrote a very simple little frontend to gorecord that can be found here: [here](#)

## Video Streaming

You can use wis-streamer to stream video from your go7007-based device to your local network or even onto the internet. There is a cool piece of software on [this page](#) that lets you easily control wis-streamer and there's also a patched version of wis-streamer that will work with the go7007 devices.

## Recording audio only

- To record only audio, first you must initialize the device:

```
gorecord -adevice /dev/dsp -vdevice /dev/video0 -nowrite
```

Note: /dev/dsp must not be the audio input of the device, use your sound card's input otherwise this won't work.

- Fire up audacity or any other audio recording software that can use an alsa input and start recording!
  - For audacity, in the preferences window the device will look something like this: „ALSA: Plector PX-M402U: (hw:1,0)“
  - Remember to use a **2 channel (stereo)** recording

## Making long recordings

**Note:** This shouldn't be needed anymore when using the new version of gorecord because there is no longer is a filesize limit. The „%d“ option is still provided but must be used with the -maxsize option to have any effect.

Because of the filesize limitation by the **old** gorecord program, you can't make really long recordings that are stored in one big avi file. The workaround is to use gorecord's %d option to continue recording to a new file after hitting the limit. We can then put the files back together again into one big avi using mencoder.

```
gorecord [options] filename-"%d".avi
mencoder -oac copy -ovc copy filename-*.avi -o final_video.avi
* Note: The individual files might not be playable until you string them
back together with mencoder.
```

# Troubleshooting

## Ubuntu usbfs fix

In order to enable for the fxload program to load the firmware onto your device you must have usbfs (or usbdevfs) mounted on /proc/bus/usb. To enable this /etc/init.d/mountdevsubfs.sh must be modified.

There is a simple patch that can be applied which is included in the latest driver source package, apply it as follows:

```
# Backup /etc/init.d/mountdevsubfs.sh
cp /etc/init.d/mountdevsubfs.sh ~/Desktop/
# Decend into source directory
cd wis-go7007-linux-0.9.8-2
# Apply the patch
wd=`pwd`; sudo patch -d /etc/init.d -p0 -i "$wd"/patches/mountdevsubfs.sh-usbfs.patch
# Mount the usbfs
sudo /etc/init.d/mountdevsubfs.sh start
```

## Device not recognized

If when you plug in the device it isn't recognized, the **dmesg** command will say something like this:

```
usb 7-2: new high speed USB device using ehci_hcd and address 9
usb 7-2: configuration #1 chosen from 1 choice
```

If it is recognized it should say something like this:

```
usb 7-2: new high speed USB device using ehci_hcd and address 8
usb 7-2: configuration #1 chosen from 1 choice
Linux video capture interface: v2.00
go7007-usb: probing new G07007 USB board
go7007: registering new Plextor PX-M402U
wis-saa7115: initializing SAA7115 at address 32 on WIS G07007SB
usbcore: registered new interface driver go700
```

This normally means that udev/hotplug isn't doing it's job (or you don't have it setup).

If you are using the latest driver, try running `go7007_firmware_load` to give you a clue as to why the firmware might not be loading.

## Activating the device manually

So here's a little bash script to recognize the device and load the firmware for your PX-M402U or PX-TV402U.

```
devices="093b:a002|093b:a004|0eb1:6666|0eb1:6668"
device=`lsusb | grep -iE $devices`
bus=`echo $device | cut -d " " -f 2`
device_num=`echo $device | cut -d " " -f 4 | sed 's:///'`
type=`echo $device | cut -d " " -f 6 | tr A-Z a-z`
[ $type == "093b:a002" ] && hex="/lib/firmware/ezusb/hpi_PX-M402U.hex"
[ $type == "093b:a004" ] && hex="/lib/firmware/ezusb/hpi_PX-TV402U.hex"
[ $type == "0eb1:6666" ] && hex="/lib/firmware/ezusb/hpi_LR192.hex"
[ $type == "0eb1:6668" ] && hex="/lib/firmware/ezusb/hpi_StarTrek.hex"
/sbin/fxload -t fx2 -I $hex -D /proc/bus/usb/"$bus"/"$device_num"
```

Here's the easy way to get this script working (The script contains some more troubleshooting info):

- **Note:** The `go7007_firmware_load` script is now included under `udev/` in the source package

wget [http://nikosapi.org/software/WIS\\_Go7007/go7007\\_firmware\\_load](http://nikosapi.org/software/WIS_Go7007/go7007_firmware_load)

```
chmod +x ./go7007_firmware_load
sudo ./go7007_firmware_load
```

## GoRecord fails to find the ALSA audio node

At least for Ubuntu 8.04 with Kernel 2.6.24, GoRecord from the 0.9.8.2 provided here fails to find the ALSA audio input provided by the `snd_go7007` module. A simple fix is to change line 217 in `gorecord.c`

```
if (!strcmp(gopath, canonpath))
```

to

```
if (!strncmp(gopath, canonpath, strlen(gopath))
```

On my system, for `gopath` and `canonpath` I have this values:

```
gopath:
/sys/devices/pci0000:00/0000:00:1d.7/usb5/5-1/5-1.1/5-1.1.4/5-1.1.4:1.0
canonpath:
/sys/devices/pci0000:00/0000:00:1d.7/usb5/5-1/5-1.1/5-1.1.4/5-1.1.4:1.0/sound/card1
```

The canonpath points to an actual subdir of gopath. The fix just verifies that gopath matches the prefix of canonpath.

## wis-streamer fails to find the ALSA audio node and emulated OSS device node

This is the same problem as the last one and the fix is similar. Use `strncmp` instead of `strcmp` in `WISInput.cpp` line 187.

In addition, the next test in `wis-streamer` fails to find the emulated OSS device node. To fix it, replace the while loop of `WISInput.cpp` in line 206 with the while loop of `gorecord.c` in line 231.

## Thanks

1. Chris P.
  - Created the patch to fix compilation issues on the 2.6.22 kernels
2. [bender647](#)
  - Created the patch to fix compilation issues on the 2.6.24 kernels

## Resources

1. <http://home.comcast.net/~bender647/go7007/>
  - A collection of patches for the driver
2. <http://home.comcast.net/~bender647/go7007/gorecord.html>
  - A tutorial on how to make DVDs of video recorded from the Plextor ConvertX boxes

## Footnote

I don't know where the best place for this is, but I've found an bug in the `wis-sony-tuner.c` module which doesn't seem to have made it into any of the patches. Not needed for `gorecord`, but other v4l experiments may result in system instability:

In the `VIDIOC_G_TUNER` ioctl, where it says:

```
memset(t, 0, sizeof(*tun) );
```

it should be:

```
memset(tun, 0, sizeof(*tun));
```

Unfortunately, the bug results in system instability if the VIDIOC\_G\_TUNER ioctl is ever invoked.-Bifferos 06:13, 16 July 2008 (EDT)

From:

<https://www.deepdoc.at/dokuwiki/> - DEEPDOC.AT - enjoy your brain

Permanent link:

[https://www.deepdoc.at/dokuwiki/doku.php?id=server\\_und\\_serverdienste:wis-go\\_multimedia&rev=1747557073](https://www.deepdoc.at/dokuwiki/doku.php?id=server_und_serverdienste:wis-go_multimedia&rev=1747557073)

Last update: **2025/11/29 22:06**

