Howto install legOS, the universal tuntap driver and the gateway.

Howto install legOS 2.5, the binutils and egcs.

To use legOS on the RCX you have to install legOS, the binutils and the egcs (gcc).
The gcc must to be recompiled to act as a crosscompiler for the RCX H8 microcontroller.

You need the files:
egcs-1.1.2.tar.bz2 (or newer)
binutils-2.5.1.tar.gz (or newer)
legOS-0.2.5.tar.gz

Login as root and create the following directory:

```
mkdir /usr/local/crossgcc
```

If you cannot login as root, create a directory of your choice and use that instead.

Move binutils and egcs into a temporary directory and unpack them with

```
tar xvzf egcs-1.1.2.tar.bz2.
```

In the same directory as you unpacked egcs and binutils create the following 2 directories:
build-binutils
build-egcs

Build the binutils with:

```
cd build-binutils
../binutils-2.9.1/configure --target=h8300-hitachi-hms --prefix=/usr/local/crossgcc --exec-prefix=usr/local/crossgcc/h8300-hitachi-hms
```

(all of this on one line).

```
make all
make install
```

Remember to use the directory you created at the beginning
/usr/local/crossgcc part.

If you get errors try to do the following to sucessfully build the binutils:

```
ln -s /usr/src/linux/include/linux /usr/include
ln -s /usr/src/linux/include/asm /usr/include
```

Now, include the path to the binutils into the PATH environment variable with:

```
export PATH=/usr/local/crossgcc/h8300-hitachi-hms/bin:$PATH
```

Installing and configuring the cross compiler.

```
cd ../build-egcs
../egcs-1.1.2/configure --target=h8300-hitachi-hms --prefix=/user/local/crossgcc --exec-prefix=/usr/local/crossgcc/h8300-hitachi-hms --enable-target-optspace
```

(all of this on one line).

Remember to use the directory you created at the beginning
/usr/local/crossgcc part.

In the build-egcs directory, open the Makefile with an editor and find the line that starts with CFLAGS. Add "-Dinhibit_libc" to the end of that line. Save and close the editor.

type:

```
make cross LANGUAGES="c c++"
```

This will take some time on slower computers.
At the end of the compilation you will be presented with an error message that looks something like this:

```
"configure error: installation or configuration problem.
C compiler cannot create executables"
```

This error can safely be ignored as legOS provides the necessary replacements.

Install the compiler with:

```
make LANGUAGES="c c++" install
```

How to Install legOS
unpack legOS-0.2.4.tar.gz to a location of your choise with

```
tar xvzf legos-0.2.4
cd legOS
```

open the Makefile.common file with an editor, and look for the line
that starts with "TOOLPREFIX=". Set this to:

```
TOOLPREFIX=/usr/local/crossgcc/h8300-hitachi-hms/bin/h8300-hitachi-hms-
```

Remember to use the directory you created at the beginning
/usr/local/crossgcc part.

To finish the installation of legOS type:

```
make
```

To verify that legOS installed correctly look for these files.
firmdl3 and dll in legOS/util
legOS.srec in legOS/boot.

If the files are present you have sucessufully installed legOS.

How to Install Lnpd

Download these 3 files:
dllx.tgz
lnpd+liblnp.tgz
lnpd.static

They can all be found on several pages on the net.

In the legOS root directory do the following

```
mkdir host
```

move the 3 files into the host directory.
unpack dllx.tgz and lnpd+liblnp.tgz with:

```
tar xvzf dllx.tgz
tar xvzf lnpd+liblnp.tgz
```

```
cd dllx
```

open the Makefile with an editor
Edit the 2 lines LIBLNPDIR and LIBLNPINC so that they point to the place where you installed lnpd+liblnp/liblnp.
They look like this:

```
LIBLNPDIR=$(HOME)/legOS/host/lnpd+liblnp/liblnp
LIBLNPINC=$(HOME)/legOS/host/lnpd+liblnp/liblnp
```

save and close the editor.

```
make depend
make
```

```
cd ../lnpd+liblnp
make realclean
make depend
make
```

Set the LD_LIBRARY_PATH with:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH: /home/username/legOS/host/lnpd+liblnp/liblnp
```

To use the commands available in lnpd you must include
liblnp.h in your programs. As i have all my programs using
liblnp.h in legOS/lnpd+liblnp/application i include them with:

```
#include "../liblnp/liblnp.h"
```

How to install the universal tun/tap driver.
This driver seems to be a part of the 2.4.x kernel (e.g. suse 7.2).
Unfortunately on my system it doesn't seem to work properly.
To install the driver get the driver from the universal tun/tap driver homepage.
Unpack the driver.
Follow the instructions in the readme file.
You have to make sure that your kernel has been configured.
Otherwise the installtion of the driver will be aborted and you will get the message that your kernel has not been configured.
To configure the kernel use make oldconfig to keep your old settings.
You have to have the kernel sources for that operation.
After your kernel has been properly configured, the installation will run without any problems.

Howto install the gateway and the patched tcp/ip enhanced kernel.

After installation of the lnpd, legOS and the compiler just replace the original files with the modified files and put the newer files to their respective drawers.
Start the gateway with &. (newer versions offer the ability to configure the ip-addresses via parameters)
Now try to ping the RCX.

This should give you ping results around 700 ms, which is totally normal.

Howto write code for the patched kernel.
Writing code for the patched kernel is a bit more complicated.
This is because this is an early version and because the version of the uip stack used calls the code itself.
Right now there is a file called httpd.c in the kernel directory. This is the place where your own code should go.
This means that you have to recompile the legOS kernel each time you are changing something.
Although this is not an ideal situation its not as bad as its sounds.
The situation will change with in the next version of legOS.
For examples on how to write code for the uip stack, please visit Adam Dunkels website.