

**IrDA, TCP/IP over IrDA, HP
Jornada-PC & iPAQ-PC connection
with Irda, ...**

Diego Chaparro
dchaparro@acm.org
Raúl Rodríguez
rrodrigu@gsyc.escet.urjc.es

Spanish version: <http://../>

24th, January, 2002

Índice

1. Introduction	3
2. Configuring <i>IrDA</i> in PC	4
2.1. Kernel configuration	4
2.2. Software	5
2.3. Module configuration	5
2.4. Making devices	5
2.5. Checking the infrared device	6
3. Connecting two PCs with <i>TCP/IP</i> over <i>IrDA</i>	7
4. Connecting a PC and a <i>HP Jornada 540</i> with <i>TCP/IP</i> over <i>IrDA</i>	8
4.1. Configuring the PC	8
4.2. Configuring the <i>HP Jornada 540</i>	9
4.3. Testing the connection	9
5. Connecting a PC and a <i>Compaq iPAQ</i> using <i>eSquirt</i>	9
5.1. Configuring the infrared port in the PC	9
5.2. Configuring infrared in the <i>Compaq iPAQ</i>	10
5.3. Installing <i>eSquirt</i> in the PC	10
5.4. Installing <i>eSquirt</i> in the <i>Compaq iPAQ</i>	10
5.5. Testing <i>eSquirt</i>	11
6. Links	11

1. Introduction

This article explains how to use infrared to communicate different kinds of devices. These devices are PCs with *Debian GNU/Linux* and some *HP Jornada 540* and *Compaq iPAQ*, a kind of *PDA*s. We have tested the connection between two PCs, a PC and a *HP Jornada*, and a PC and a *Compaq iPAQ*.

To test infrared, we have used infrared special devices connected to the serial port. These devices are known as *IR Dongles*, and in these tests we have used the *ACTISYS-IR 220L plus IR Dongles*.

2. Configuring *IrDA* in PC

In PCs (non laptops), there is no infrared port, so to use infrared in PCs you can use a *dongle*. A *dongle* is a device connected to the serial port that allow you to use infrared from PC.

There are several types of *dongles*. We have used the *ACTISYS-IR 220L plus dongle*. You can get it from the *actisys* web page¹. This model supports perfectly *Debian GNU/Linux*.

2.1. Kernel configuration

In our test, we have used a kernel from 2.4.x series, but it should work with series 2.2.x too. In this case, you probably have to apply a patch to the kernel.

You have to enable some options in kernel, in the *IrDA (infrared) Support* section. You should to choose these options as modules, not including them into the kernel. This option is better because the modules are only loaded when you are going to use the infrared device.

The basic *IrDA* options you should to enable are:

- CONFIG_IRDA
- CONFIG_IRLAN
- CONFIG_IRCOMM
- CONFIG_IRDA_ULTRA
- CONFIG_IRDA_OPTIONS

The options to support an infrared device attached to the serial port are:

- CONFIG_IRTTY_SIR
- CONFIG_IRPORT_SIR

The option to support the *ACTISYS-IR 220L plus IR dongle* we have used in our tests is:

- CONFIG_ACTISYS_DONGLE

When you have selected this options, you will have to save changes and compile the kernel with the new options:

- make dep
- make clean
- make bzImage
- make modules
- make modules_install

¹<http://www.actisys.com/>

2.2. Software

The next step is to install the software we need. The basic software is the `irda-common` package. This package offers some tools like `irmanager` or `irattach` that we need to configure our device.

You can also install the `irda-tools` package. This package has tools like `irdadump` or `irdaping`. They are very important tools for debugging and testing.

These packages, `irda-common` and `irda-tools` are available in the *Debian GNU/Linux* distribution and in other distributions too.

2.3. Module configuration

We have to configure the modules to be automatically loaded when the kernel needs them. You have to configure the aliases in the `/etc/modules.conf` file (in *Debian GNU/Linux*).

We create the `/etc/modutils/irda` file (if it doesn't exist) and write the following:

```
#modutils/irda
alias tty-ldisc-11 irtty
alias char-major-161 ircomm-tty
alias char-major-60 ircomm_tty
alias char-major-10-187 irnet
#for dongle
alias irda-dongle-2 actisys
alias irda-dongle-3 actisys+
```

Then, you have to run from a shell the `update-modules` command to update the `/etc/modules.conf` file.

2.4. Making devices

We need the devices that the *dongle* is going to use to communicate with other devices. You have to execute the following commands:

```
mknod /dev/ircomm0 c 161 0
mknod /dev/ircomm1 c 161 1
mknod /dev/irlpt0 c 161 10
mknod /dev/irlpt1 c 161 11
mknod /dev/irnet c 10 187
```

2.5. Checking the infrared device

Once you have completed the previous steps, you have your PC configured to be able to use the infrared device attached to the serial port.

You can check if everything is ok running the next command:

```
irattach /dev/ttyS0 -d actisys -s
```

"/dev/ttyS0" means the serial port in which we have attached the device (if you use another serial port, you need to change this parameter).

After that, you can check if all the necessary modules are been loaded running the `lsmod` command. You obtain something like this:

- actisys
- irtty
- irda

You can also use the `/etc/init.d/irda` script to do the last step, to load the modules when we need them, and to download them we don't need them anymore.

To check if these devices send or receive something, you can configure two PCs, put the *IR dongles* in front of each one and run the `irdadump` command. You should see every packet that send and receive every device.

3. Connecting two PCs with *TCP/IP* over *Ir-DA*

It is very easy to connect two PCs with *TCP/IP* over *IrDA*. We need to load the modules and put the *dongles* in front of each one.

First, we have to verify we have the following options in our kernel (as modules or included into the kernel):

- `irnet`
- `ppp_generic`
- `ppp_async`
- `ppp_deflate`

Then, we run the next command in PC1:

```
pppd /dev/irnet 9600 local noauth dirIPpc1:dirIPpc2
```

And the following in PC2:

```
pppd /dev/irnet 9600 local noauth dirIPpc2:dirIPpc1
```

You have to change *dirIPpc1* and *dirIPpc2* for the PC1's *IP* address and PC2's *IP* address.

When you execute these commands, they should create a network device in PC1 and PC2 called *ppp* with the specified *IP* address. The *9600* parameter means the connection speed. You can change this value to a higher value (perhaps 115200).

Now we have a *TCP/IP* connection between two PCs over infrared devices. You can test with *ping*, *ssh* and *http* connections, ...

4. Connecting a PC and a *HP Jornada 540* with *TCP/IP* over *IrDA*

You can also connect a PC and a *HP Jornada* with *TCP/IP* over *IrDA*. You need to configure the PC with the infrared device as the *ppp server* in the connection, and then to configure the *HP Jornada* to connect to this server using the infrared port.

4.1. Configuring the PC

We create a `/usr/sbin/cebox.sh` file with the following content:

```
#!/bin/sh
pppd call cebox
```

This file is executed in PC to run the *ppp server*. A `/etc/ppp/cebox.chat` file contains the next:

```
AT OK
AT OK
AT OK
AT OK
AT OK
ATDT CONNECT
```

This file contains the authentication options for the PC and the *HP Jornada*.

Finally, you have to make a link between `/dev/irnine` and `/dev/ircomm0`:

```
ln /dev/ircomm0 -s /dev/irnine
```

and to create the next file (named `/etc/ppp/peers/cebox`):

```
/dev/irnine 115200 nocrtscts
connect /usr/sbin/chat -v -f /etc/ppp/cebox.chat
noauth
local
dirIPpc:dirIPjornada
ms-dns servidorDNS
```

You have to change *dirIPpc* for the PC's *IP* address and *dirIPjornada* for the *HP Jornada*'s *IP* address. You also have to change *servidorDNS* for your DNS server's *IP* address.

4.2. Configuring the *HP Jornada 540*

You have to follow the next steps to configure correctly the *HP Jornada*. First, make a new modem connection. You have to select **Start | Settings** in your *HP Jornada*. Then, choose **Connections** and pick in **Modem**.

Next, in **New connection...**, choose the **Generic IrDA** modem. Choose the modem speed (115200) and pulse on the **Advanced** button. You only have to change the **Flow Control** to *Software*.

Then, pulse the **OK** button to go to the previous screen and pulse **Next**. In the **Phone number** entry, you have to type "0" until the box is full. Pulse the **Next** button and, finally, the **Finish** button.

Now, we have already configured the *HP Jornada* to be the client in the *TCP* connection.

4.3. Testing the connction

First, we execute `/usr/sbin/cebox.sh` in a shell in the PC. Now, the PC is ready to be the *ppp server* and to accept *TCP/IP* connections.

Next, we have to run the client in the *HP Jornada*. We have to select **Start | Programs | Connections**. In the folder will appear the connection created in the previous section. When you pick in the icon, the *ppp* connection starts and we have connected the PC and the *HP Jornada* with a *TCP/IP* connection over *IrDA*.

To test the connection, you can run a web server in the PC and get its pages from the *Internet Explorer* in the *HP Jornada*, or you can configure a proxy server in the PC and visit everywhere in Internet.

5. Connecting a PC and a *Compaq iPAQ* using *eSquirt*

eSquirt is a simple program developed by *HP Labs*² for the *Cooltown*³ project and it is used to transfer data between devices using the infrared ports.

In this test, we have used a PC with an *Actisys-IR 220L plus IR Dongle* connected to its serial port, and a *Compaq iPAQ* with *Familiar 0.4*⁴ distribution.

To be able to communicate the PC and the *Compaq iPAQ* using *eSquirt*, first we need to configure the infrared ports in both, and then install *eSquirt* in both of them too.

5.1. Configuring the infrared port in the PC

You should follow the same steps shown in the 2 section.

²<http://www.hpl.hp.com/>

³<http://www.cooltown.hp.com/>

⁴<http://familiar.handhelds.org/>

5.2. Configuring infrared in the *Compaq iPAQ*

The *Familiar 0.4* distribution we have used has kernel version 2.4.3, and it has some packages for using *IrDA*. These packages are `irda-common` and `irda-modules-2.4.3.rmk2-np1`.

To install these packages you have to execute in the *Compaq iPAQ* the next:

- `ipkg install irda-common`
- `ipkg install irda-modules-2.4.3.rmk2-np1`

After you should run `depmod -ae` because the `irda-modules-2.4.3.rmk2-np1` package adds the modules we need to use *IrDA* in the *Compaq iPAQ* to the kernel. In this way, we have configured the *Compaq iPAQ* to use the infrared port. To enable it, you have to execute the following command:

```
ifconfig irda0 up; echo 1 >/proc/sys/net/irda/discovery
```

and to disable:

```
ifconfig irda0 down; echo 0 >/proc/sys/net/irda/discovery
```

5.3. Installing *eSquirt* in the PC

To install *eSquirt* in the PC, we need to install two packages, the `eSquirt` package and the `openobex` package. The last package contains the libraries that *eSquirt* uses.

You can get the `eSquirt` package in this file⁵, and the `openobex` package in this web page⁶.

First of all, we have to install the `openobex` following the typical way: `./configure`, `make` & `make install`. Then, you have to install `eSquirt` in the same way.

After that, in the folder you have installed the `eSquirt`, you will find a `tests` directory containing some programs to test *eSquirt*.

5.4. Installing *eSquirt* in the *Compaq iPAQ*

The `eSquirt` and `openobex` installation in the *Compaq iPAQ* is more difficult because the *Compaq iPAQ* hasn't a `gcc` compiler to compile itself the source code. However, we have several options to compile `eSquirt` and `openobex` for the *Compaq iPAQ* architecture.

One option is to install a cross compiler in the PC to produce binaries for the *Compaq iPAQ* architecture. Another option is to use the *Skiff Cluster*⁷. The *Skiff Cluster* is a set of PCs with the same architecture that the *Compaq iPAQ*. These

⁵<http://devnet.hp.com/download/coolbase/97/Esquirt.zip>

⁶<http://sourceforge.net/projects/openobex/>

⁷<http://www.handhelds.org/projects/skiffcluster.html>

PCs allow us to compile our source codes. The URL is <http://skiffcluster{1-6}.handhelds.org/>⁸, and you can log in using `telnet` with user *"guest"* (without password).

First, we have to put the `eSquirt` and `openobex` packages in a PC of the *Skiff Cluster* and then log in in that PC and compile the sources in the habitual way: `./configure, make and make install`.

But you have to know some considerations to compile correctly the sources, like when you are going to execute `make install`, this command will try to install the binaries and libraries in the `/bin`, `/lib` and `/include` directories in the PC where we are compiling the sources, but we haven't permissions to do this, so you should create your own `/bin`, `/lib` and `/include` directories and modify the `Makefile` to pointing to these directories.

When you have compiled the sources, you will have to copy all produced files (and all the folders) in the *Compaq iPAQ*. You have to copy the files in the `/lib` directory in the cluster into the `/lib` directory in the *Compaq iPAQ*, the files in the `/bin` in the cluster, into the `/bin` directory, ...

At this point, we have the `eSquirt` ready to be run in the *Compaq iPAQ* architecture.

5.5. Testing *eSquirt*

To test that we have properly configured the PC and the *Compaq iPAQ*, we can test the `uobex_squirt` program that it is included in the `tests` directory. We can send an URL from the *Compaq iPAQ* to the PC. In the PC, you can execute the next:

```
./uobex_squirt
```

This command run a server that waits to receive messages with URLs.

In the *Compaq iPAQ*:

```
./uobex_squirt http://gsyc.escet.urjc.es/
```

This command send that URL using the infrared port. If everything is all right, the PC should receive a message and display the web page.

6. Links

- [Irda HomePage](#)⁹
- [Linux IrDA HomePage](#)¹⁰
- [Irda-Utills](#)¹¹
- [Linux-IrDA mailing list](#)¹²

⁸<http://skiffcluster1.handhelds.org/>

⁹<http://www.irda.org/>

¹⁰<http://irda.sourceforge.net/>

¹¹<ftp://irda.sourceforge.net/pub/irda/irda-utills/>

¹²<http://www.pasta.cs.uit.no/mailman/listinfo/linux-irda>

- Linux IR HOWTO¹³
- Linux Infrared HOWTO¹⁴
- Actisys¹⁵
- HP Labs¹⁶
- Cooltown¹⁷
- Familiar 0.4¹⁸
- openobex¹⁹
- Skiff Cluster²⁰

¹³<http://linux-embedded.com/howto/IR-HOWTO.html>

¹⁴<http://mobilix.org/Infrared-HOWTO/Infrared-HOWTO.html>

¹⁵<http://www.actisys.com/>

¹⁶<http://www.hpl.hp.com/>

¹⁷<http://www.cooltown.hp.com/>

¹⁸<http://familiar.handhelds.org/>

¹⁹<http://sourceforge.net/projects/openobex/>

²⁰<http://www.handhelds.org/projects/skiffcluster.html>