Hess2 local module HOW-TO

Table des matières

1 Introduction

The LPNHE laptop provide a 2.6.24 LINUX kernel and a shl-3.2.3 LINUX distribution. The */home* partitions are also provided by the laptop and mounted via NFS. Indeed the 3 */home*'s directories */bigcamera /guevara* and */roche* are accessible from both the laptop and the controller card side. (sorry about using 3 account, it's a direct copy of our own system).

2 Boot the controller card

- Connect the CPCI create and the LPNHE laptop with an RJ45 cable.
- Boot the LPNHE laptop : choose UBUNTU at grub menu prompt.
- Login with **lpnhe**. The password is written on the laptop.
- Open a terminal in order to look later at the TFTPD and MOUNTD logs :

```
$ tail -f /var/log/messages
...
```

- Boot the CPCI create and look for the logs :

```
...
tftpd:  trying to get file: /tftpboot/zImage15
mountd: authenticated mount request...
```

- Connect as **root** to the controller card :

```
$ telnet 192.168.1.15
login> root
```

3 Run pilot on the controller card

- Connect as **root** to the controller card (see above)
- Probe the *tmgts* driver (the *xpc* driver is needed)
- Run the *TmgTsPilot* pilot with argument `0` or `1`

```
$ telnet 192.168.1.15
login> root
# ./loadDriver.sh
# TmgTsPilot 0
```

4 Run client on the laptop

So as to save the data received from the pilot, you can use a client over TCP.

- Open a terminal
- Run the *tmgtsclt* client
- Configure the pilot to send data to the IP `192.168.1.9` and the port 60200
- Read the data from the pilot. This will create new data files on client side.

```
$ ./tmgtsclt
^C
$ ls data_*
```

5 Compiling driver and pilot on the controller card

```
$ telnet 192.168.1.15
login> guevara

$ cd SBig/Driver/TmgTs
$ touch tmgts.c
$ make

$ cd SBig/Driver/TmgTs/Pilot
$ touch TmgTsPilot.c
$ make
```

6 Compiling the client on the laptop

```
$ su - nroche
passwd> 20hess02
$ cd src/tmgtsclt
$ make clean
$ make
```

7 Conclusion

Please note that the LPNHE laptop provide you the entire HESS2 development environment and sources. For instance you should also be able to compile our embedded server using the cross-compile tools. If needed just ask for more documentation.