IRAM-COMP-019

Revision: 3
15 OCT 2008

Contact Author

Institut de RadioAstronomie Millimétrique

PdB New Generation Antenna Mount Software

Owner Alain Perrigouard (perrigou@iram.fr)

Keywords:

Approved by:	Date:	Signature:
A.Perrigouard	December 2005	

Change Record

REVISION	DATE	AUTHOR	SECTION/PAGE AFFECTED	REMARKS
1	13 Dec 2005	Perrigouard	1st release	
2	15 Nov 2007	Perrigouard	Section 2 (shell…)	
3	15 Oct 2008	Perrigouard	Section 6.5 added	

Content

1 Introduction

This document deals with the antenna mount drive software. This control package is executed on a Single Board Computer, a VMIVME 7700, which accesses the antenna drive hardware through the VME bus. The SBC is located in the first slot of a VME chassis. The chassis layout is given later in this document and the descriptions of the different VME modules are given in separate documents.
The SBC behaves exactly as a PC and for our purpose of controlling and monitoring the antenna mount a real-time version of LINUX has been installed as operating system.
The first chapters describe the user commands and interfaces. The last chapters deal with the installation of the diskless real-time operating system and the development of the application software.

2 Shell commands

The shell commands are available from the account oper. They are executed either from a local console or a secure shell window or as a command parameter in a ssh client program executed from a remote host.
In the following, in the first list, there are commands for monitoring, debugging and controlling the antenna drive software. The other commands are tasks executed once at start time or they are specific test programs.
The SBC are named ant*1.iram.fr with * standing for the antenna number from 1 to 6. For instance for antenna 5 which is in fact the first antenna with this configuration, a secure shell connection from a remote host looks like this:

```
[perrigou@bure2 perrigou]$ ssh oper@ant51
oper@ant51's password:
Last login: Mon Dec 19 04:18:26 2005 from bure5.iram.fr
ant51 oper:~ $
```

In some cases the password is not requested (from an oper session on bure2/4/5).

Normal operation commands:

- dmp. Displays the shared memory area which is defined in the include file antenna.h
- horizon Asks for an azimuth and an elevation, angles in degrees, and move the antenna to that positon in the horizon coordinate system
- incr. Initializes the incremental encoders. The command without argument displays the usage:
  Usage: incr [qualifiers]
  qualifier: AZIMUTH Init the Az encoder to the nearest pos.
  ELEVATION
  +AZIMUTH
  -AZIMUTH
  Both encoders or only one at the time may be initialized
  When the incremental encoders are initialized (with the command incr), the task checks the coherence between the different offsets, incremental preload offinc and absolute encoder offet offabs. If a discrepancy gives a position error between the absolute and incremental encoders bigger than 2PI/2^11 (4 absolute encoder steps) the problem is remotely logged to bure1b (see tail3) and the system does not switch to the incremental encoder (azimuth.absolute or elevation.absolute remains equal to 1). Only the deprecated command incr2 may force the incremental encoder use and once it is done it is strongly recommanded to update the absolute encoder offset with the command offAbs.
- incr2. Very similar to the command incr. However, the command incr2 does not move the antenna axes to their init positions that are around 177deg in azimuth and 44deg in elevation which necessitate the absolute encoders. The command incr2 moves only in slew mode (1/4 of the full speed) the axis(es) to be initialized, from the current position(s) until the encoder zero pulse(s) is(are) encountered and, as a consequence, the incremental encoder counter(s) is(are) preloaded.
- io. The command without argument displays the usage:

Usage: io output_nber mask value
output_nber 1 or 2
mask hexadecimal number [0, 0xFFFF]
value hexadecimal number [0, 0xFFFF]

The name given to each bit for the 2 VME output registers are recorded within the 2 enum out1 and out2 found in the include file antenna.h:

enum out1{O_El, O_okB2, O_Vxup, O_Az, O_proj, O_Vxdn, O_Vant, O_Vcar, O_SA, O_AzB, O_ElB, O_SH, O_SZ, O_gene, O_uB6, O_lB6};

enum out2{O_PB=1, O_gener, O_Ptrck=5, O_Pgene};

For example, to set the bits O_proj and O_gene (without changing the other bits):
io 1 2010 2010
To clear the bit O_proj:
io 1 10 0

- offAbs For each axis and when the incremental encoder is initialised (azimuth.absolute = 0 or elevation.absolute = 0), the command calculates new values of offabs such that, both encoders (absolute or incremental) provide the same axis positions, expressed in tenths of arcsecond and, of course, apart from the different resolutions. The new estimated offabs are proposed for update when they differ from the current used values. When update is accepted, the current value is changed in memory and the corresponding reference is updated in the config.ant*1 file (in fact the new value is appended in /home/introot/piedestal/src/config.ant*1 with the current update time).

- offInc <azOff> <elOff> the 2 parameters are integer values expressed in tenths of arcsecond. They represent the offset between the target in the sky and the achieved antenna positon.
  In order to aim perfectly to the target those values are substracted to azimuth.offinc and elevation.offinc, the values preloaded to the incremental encoders at initialization time.
  azimuth.offinc and elevation.offinc are changed in memory and appended to the configuration file /home/introot/piedestal/src/config.ant*1 with the current update time.
  If the incremental encoders are already initialized this operation has no immediat effect: The incremental encoders needs to be initialised again with the command incr.
  In addition azimuth.offabs and elevation.offabs are changed in memory and the new estimated values are appended also in the configuration file.

  Related comment:
  When the the incremental encoders are initialized (with the command incr), the task checks the coherence between the different offsets, incremental preload offinc and absolute encoder offet offabs. If a discrepancy gives a position error between the absolute and incremental encoders bigger than 2PI/2^11 (4 absolute encoder steps) the problem is remotely logged to bure1b (see tail3) and the system does not switch to the incremental encoder (azimuth.absolute or elevation.absolute remains equal to 1). Only the deprecated command incr2 may force the incremental encoder use and once it is done it is strongly recommanded to update the absolute encoder offset with the command offAbs.

  Related command available from bure1b:
  /control/command/bin/offInc <antenna_number> antenna_number integer from 1 to 6
  This command reads the offset IAZ and IEL from the file ~oper/pdbi-data/base/general.an<antenna_number>, executes on the remote processor ant<antenna_number>1b the command offInc with the offsets in "/10 units, and finally clears IAZ and IEL in the file general.an<antenna_number>

- remote. The command turns the axes in slew mode with the velocities equal to zero. The antenna will look like a weather cock, free to move in the wind flow. The command dmp should show vx equal to 0 for the azimuth and the elevation.

At boot time, when the axes control software is started (the key of the drive amplifier cabinet being in local), use this command before turning into remote the drive amplifier cabinet to avoid dangerous and hard jump.

- setTime. This command resets the counter utSec, the number of seconds since midnight UT. The command requests this elapsed time to the server clock. The operation is trigged on the 1s interrupt generated from the 1s pulse on the time bus and the answer generated by the server clock should be received inside the same 1s period.
  SetTime is executed at boot time within the script initialize_antenna.sh. It can be executed later at any time in order to reset the counter utSec. The current value of utSec is displayed with the command dmp, section synchro.
- stop. This commands stops smoothly the antenna if the axes are spinning. When the antenna is at rest, the command will maintain the axes in fixe positions. In strong wind , the command may be very demanding on the servos and the command remote may be an alternative.

Background commands or specific test programs:

- config. In principle this command is executed only once at boot time within the script initialize_antenna.sh. The command is executed after the command init which a more general task. The command reads the file $INTROOT/src/config.<hostname> with hostname, the name of the micro. For instance config.ant51 is the configuration file for the micro ant51 in charge of the drives of antenna 5.
  Each line starting with the name of an item of the structure antenna_s (defined in antenna.h) followed by a value compatible with its type is valid and is used to set that item. The rest of the line are treated as comment.
  Example:
  elevation.kp 8000 El proportional factor
  azimuth.init 177. (deg) Az: Init start pos. for inc. encoder
  Any other line that does not start with the name of an item of the stucture antenna.s is assumed to be a comment line.
- inclino. This task is executed in background and is started with the script initialize_antenna.sh. The task reads every second the inclinometer tilts and temperature through a RS232 line connected to port COM1. The values are saved in the shared memory area and can be displayed with the command dmp section inclino.
- init. This task initializes the shared memory area with values valid for all antennas. The task also enables the 16 bit DAC which is used to command the velocities, checks the incremental encoder VME modules to request or not their initializations, starts the VME I/O register operations, and finally enables the ADC VME module reading. This task is executed at boot time when the script initialize_antenna.sh is executed. It should no be executed later.
- subrefRecv. Program to test the communications from subrefXmit. The program subrefRecv running on the piedestal micro will accept a connection from subrefXmit only when this later would have been compiled with the option SIMULATION.
- subrefXmit. This task is executed in background and is started at boot time with the script initialize_antenna.sh. In principle the task achieves a TCP/IP connection on port 1052 with a server running in the cabin.
  When compiled with the option SIMULATION, the task will connect to a server running in the same processor. This server will be subrfRecv started for test purpose.
- tcpServer. A server running in background and started at boot time with the script initialize_antenna.sh. The server accepts a TCPIP connection on port 1051 normally performed by the task interp running on bure1.
  Periodically the server receives a buffer declared as int request[7], returns another buffer declared as Mess_t mess and which contains the status of the antenna drives and finally, depending on the values of the elements of the array request[7], receives blocks of data saved in the sections of type Synchro_t, Astro_t, Source_t, Offset_t, Pointing_t, Refraction_t and Io_t of the shared memory area.

Bure1b commands:
The following commands are in the oper account path defined with the variable PATH and as a consequence can be executed directly without specifying any relative path.

- coo [qualifiers] direction [qualifiers]
  direction: [coordinate_ref] angle_spec angle_spec
  or DAY long_spec lat_spec validation_time

  with coordinate_ref:= EQUATORIAL
  or ECLIPTIC
  or HORIZONTAL [default]
  angle_spec:= angle[,proper_motion]
  proper_motion:= Source yearly proper motion defined at epoch J2000 in RA, DEC expressed in seconds of time /year for RA and in "/year for DEC.

  long_spec:= angle 1st_derivative 2nd_deriv. (s)(s/h)(s/h/h)
  lat_spec:= angle 1st_derivative 2nd_deriva. (")("/h)("/h/h)
  validation_time:= UT time (h)
  long_spec and lat_spec are valid from time to time+1

  qualifier: EPOCH epoch(year)
  PARALLAX annual_parallax(")
  TELESCOPE telescope_number(1-6)
  ANTENNA antenna_number(1-6)
  LSR velocity
  HELIOCENTRIC velocity
  EARTH velocity
  NULL_VELOCITY (does not need radial velocity)
  SUN (don't care about sun avoidance)
  +AZIMUTH
  -AZIMUTH
  one of the qualifiers ANTENNA or TELESCOPE is required

- dri [qualifiers] drift_parameters [qualifiers]
  drift_parameters: speed tilt start
  speed: angle per second
  tilt: angle/meridian
  start: drift start UT time (s)

  qualifier: TELESCOPE telescope_number(1-6)
  ANTENNA antenna_number(1-6)
  HORIZON set the offset coordinate system to horizon.
  The default value is the preset coordinate system.
  RELATIF The start time is relatif to the current time
  For speed>0 and tilt=0D El or lat increases, Az or lon stays constant
  and for tilt=90D Az or lon decreases, El or lat stays constant.
  One of the qualifiers ANTENNA or TELESCOPE is required

- ini [qualifiers]
  qualifier: AZIMUTH Init the Az encoder to the nearest pos.
  ELEVATION
  +AZIMUTH
  -AZIMUTH
  SUBREFLECTOR
  TELESCOPE telescope_number(1-6)
  ANTENNA antenna_number(1-6)
  one of the qualifiers ANTENNA or TELESCOPE is required
  default initialization: AZ EL SUBREFLECTOR

- io [qualifiers] io_parameters [qualifiers]
  io_parameters: nber mask value
  One of the qualifiers ANTENNA or TELESCOPE is required

- off [qualifiers] offset_direction [qualifiers]
  offset_direction: angle_spec angle_spec
  qualifier: TELESCOPE telescope_number(1-6)
  ANTENNA antenna_number(1-6)
  HORIZON set the offset coordinate system to horizon.
  The default value is the preset coordinate system.
  One of the qualifiers ANTENNA or TELESCOPE is required
  DELAY offset the delay tracking center

- OffInc. Updates the encoder offsets
  Usage: offInc antenna_number
  Antenna _number (1-6)
  /control/command/bin/offInc <antenna_number> antenna_number integer from 1 to 6
  This command reads the offset IAZ and IEL from the file ~oper/pdbi-data/base/general.an<antenna_number>, executes on the remote processor ant<antenna_number>1b the command offInc with the offsets expressed in "/10 units, and finally clears IAZ and IEL in the opertor's file general.an<antenna_number>

- pla [qualifiers]
  qualifier: planet required
  TELESCOPE telescope_number (1-6)
  ANTENNA antenna_number (1-6)
  one of the qualifiers ANTENNA or TELESCOPE is required
  planet: MERCURY, VENUS, MARS, JUPITER, SATURN, URANUS, NEPTUNE, MOON, PLUTO, CERES, PALLAS, JUNO, VESTA

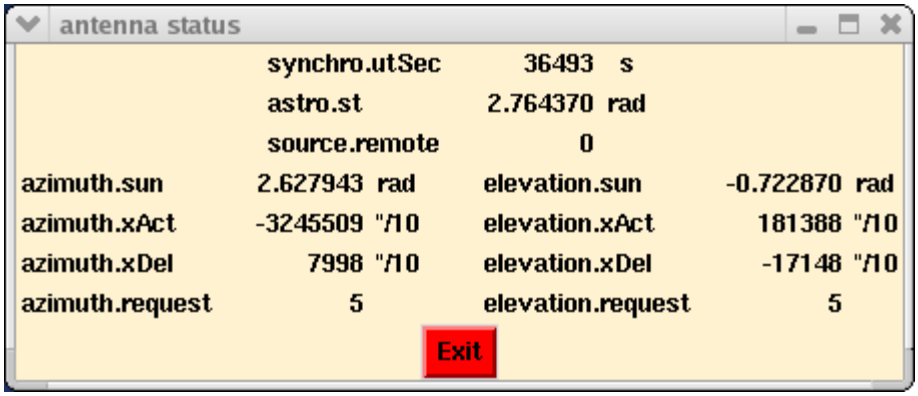
- put [qualifiers] ANTENNA antenna_number [qualifiers]
  with antenna_nber (1-6)
  qualifier: OFFSAZ angle zero azimuth encoder
  OFFSEL angle zero elevation encoder
  COLLVT angle vertical collimation
  COLLHO angle horizontal collimation
  EPSXZA angle mount verticality (westwards)
  EPSYZA angle mount verticality (southwards)
  TILTEL angle non perpendicularity (el/az)
  EXCCAZ angle Az excentricity, cosine term
  EXCSAZ angle Az excentricity, sine term
  EXCCEL angle El excentricity, cosine term
  EXCSEL angle El excentricity, sine term
  HOMOEL angle homology elevation (main axis)
  HVTIL0 angle homology subreflector vertical tilt
  HVTIL1 angle
  HTRAN0 length homology subreflector vertical translation
  HTRAN1 length
  HFOCU0 length homology subreflector focus translation
  HFOCU1 length
  ANTENNA qualifier required

- set_sun
  sends to all antennas the sun coordinates

3 Graphical user interfaces

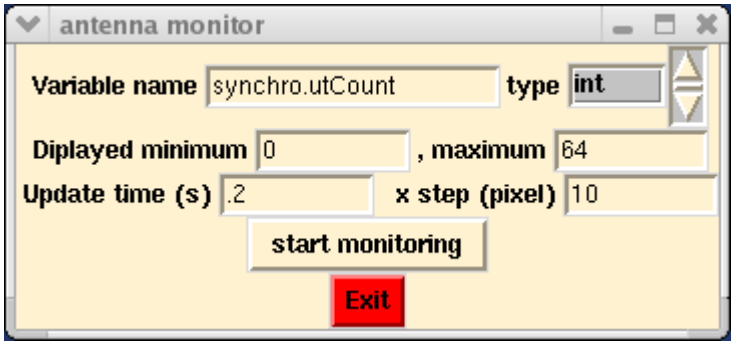
3.1 status.py

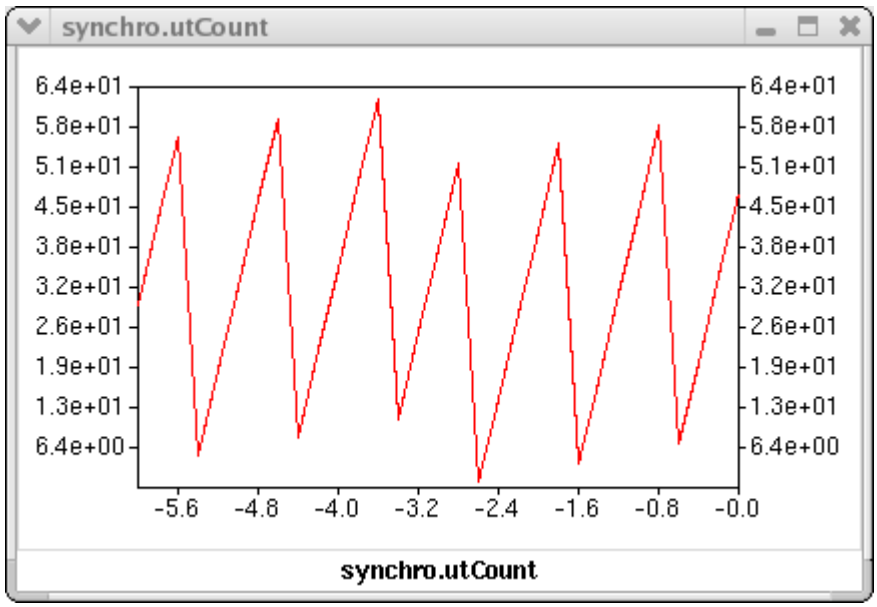
This graphical interface displays essential information about the antenna drives.
From its design and as it is a python script, this GUI is very versatile and can be edited to display any variable from the share memory area. The variables, members of the structure antenna.s are referred by their names of the form [struture-name.]member. For instance snchro.utSec or elevation.xAct.
The display is updated every second.



3.2 monitor.py

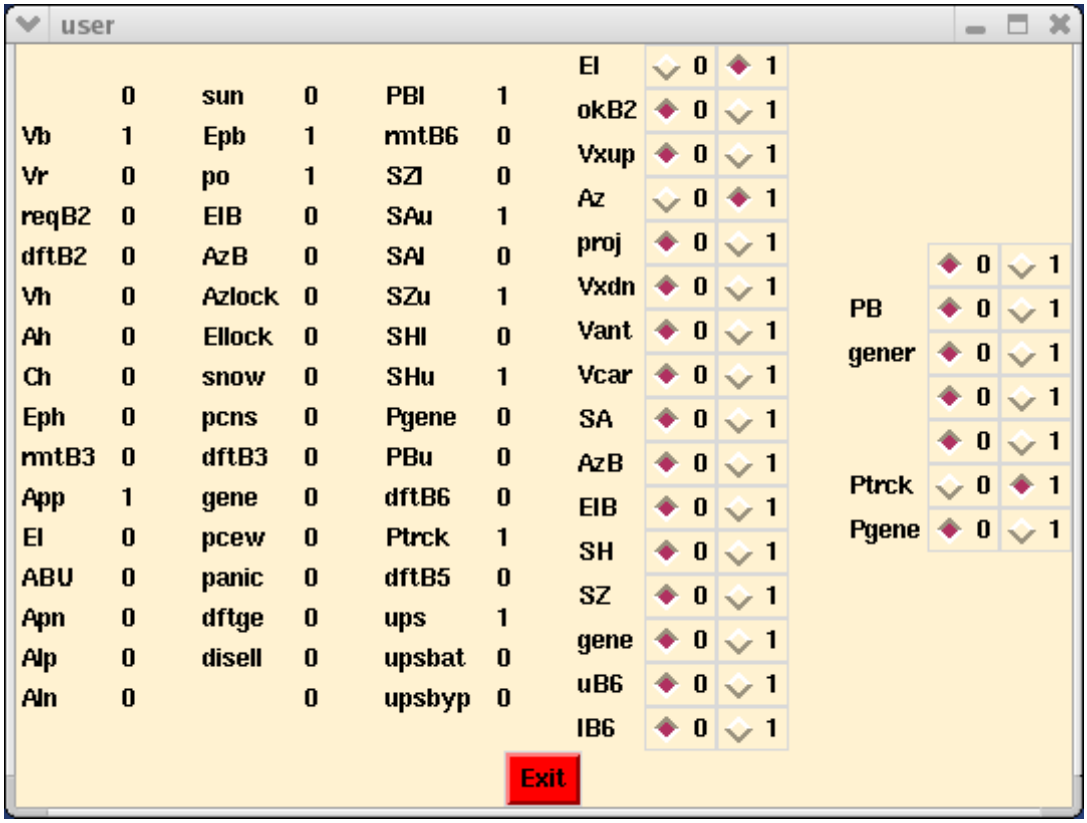
monitor.py may be used to monitor one or several variables of the shared memory area defined with the new type Antenna_t (structure antenna_s) declared in antenna.h.
Valid variables may be, for instance, synchro.utCount, azimuth.xAst or astro.st
The type of the variable has to be selected. It can be either int or float or double.
Update time is the sampling period and x step (pixel) is the extent of each sample along the horizontal axis.
When the start monitoring button is clicked, a new window is created with the variable name as title. This window will be updated every “update time”, the plot updated from the right and cropped vertically between the selected “displayed minimum” and “maximum”.
Every “update time” a horizontal segment of “x step (picel)” is displayed on the right side of the plot while pushing the curve to the left. As a consequence, as the plot area is 300 pixel wide, the plot covers a time extend equal to 300 * ”update time” / ”xstep” in seconds.





3.3 user.py

Graphical interface to monitor the VME input registers in1 in2 and in3 and to control the output registers out1 and out2 (the names in1/2/3 or out1/2 refer to enum declarations in antenna.h):



The display is updated every second. Even the radio buttons of the output registers are updated when those registers are updated beside this GUI for instance by using the local command io or by receiving a request from bure1 through the task tcpServer.
Each time the state of a radio button is changed by a user click, the corresponding variable in the share memory area is updated (see dmp) and the VME output register written.

4 XEphem

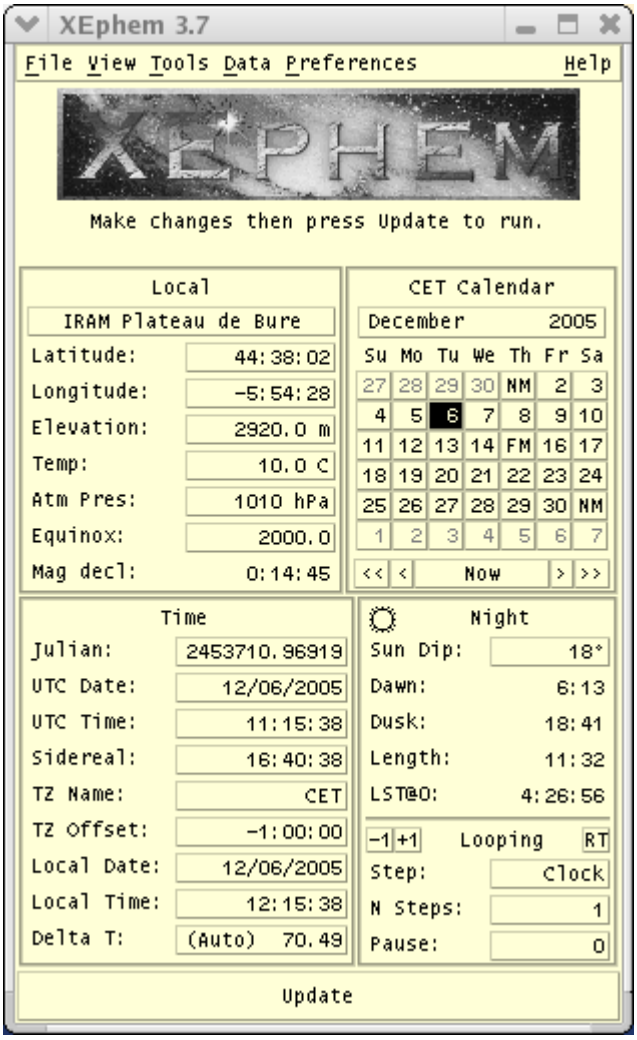
XEphem is the well known interactive astronomical ephemeris software package.
Here is an example of interfacing the antenna drive control system to XEphem by use of a simple XML-like communication protocol called INDI, standing for Instrumental-Neutral Distributed Interface.
For a first test only the equatorial J2000 and horizon positions of the antenna are displayed in the “xephem INDI Control Panel” window. There is not yet the possibility to command the antenna from XEphem. The position of the antenna may also be displayed with a specific marker in the sky views, horizon or equatorial.

Prerequisite:

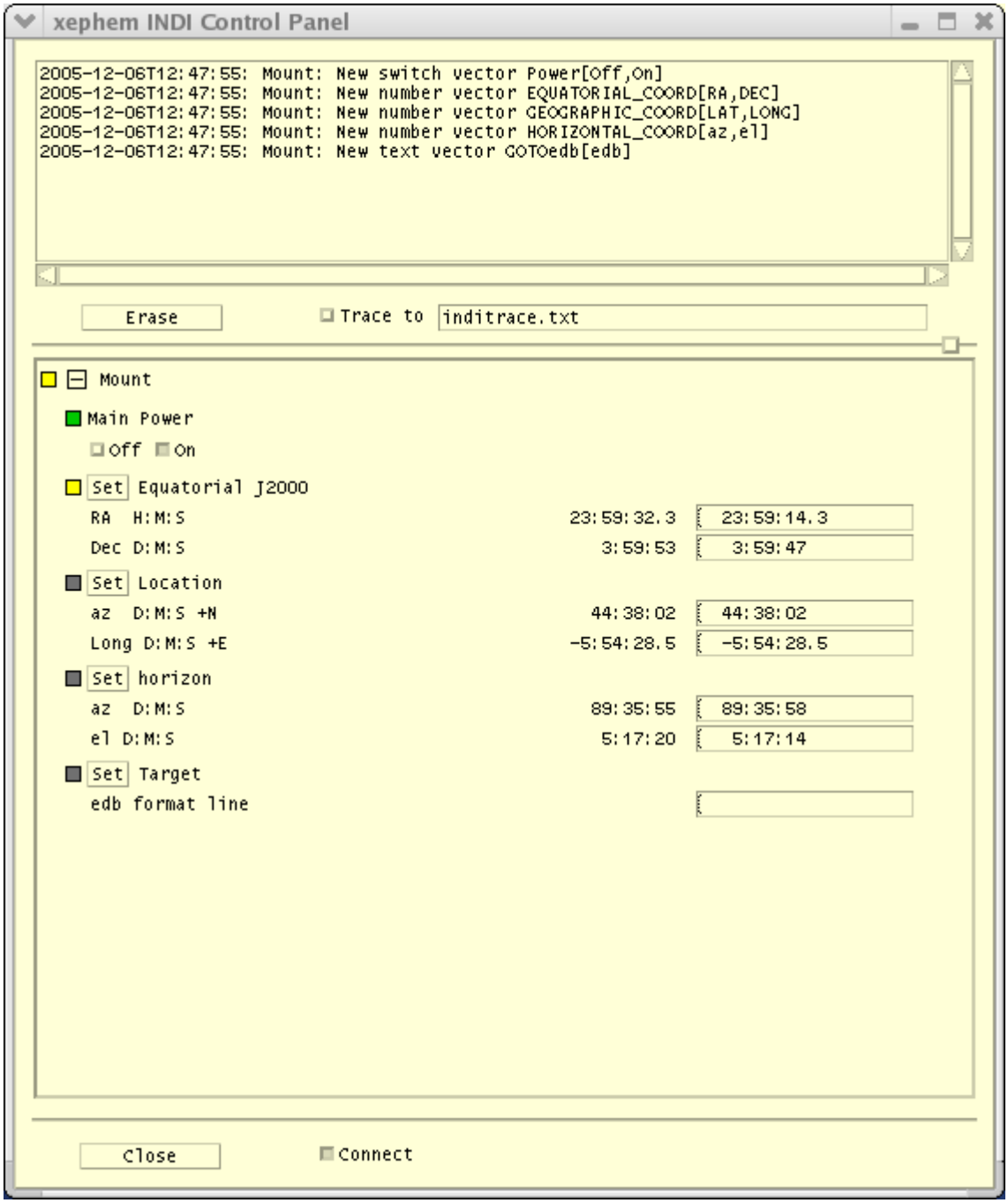
```
ant51 root:/home/perrigou/devel/PdB/LINUX/piedestal/antennaRTAI/indi #
./indiserver ./mountPdB > x.err 2>&1 &
```

To start xephem

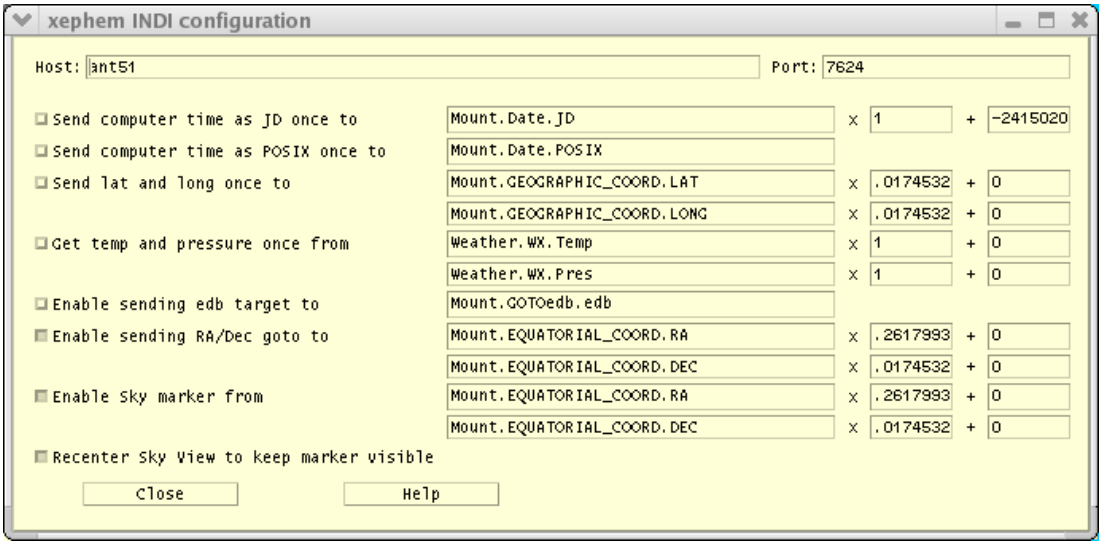
```
ant51 oper:~ $ xephem
```



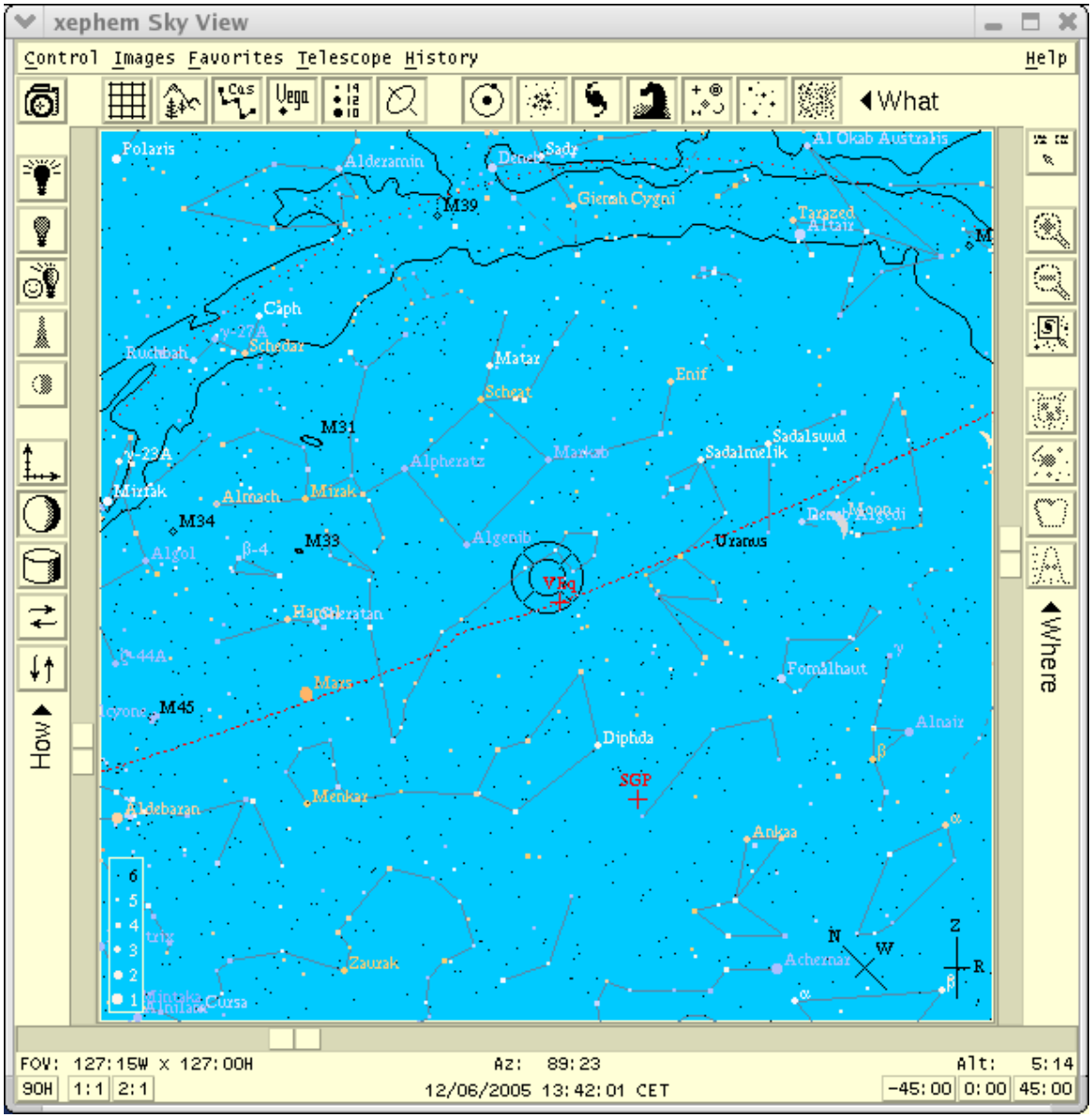
To display the INDI Control Panel, first display the sky view:
View > Sky View …
And from the new “xephem Sky View” display:
Telescope > INDI Panel …
In the “xephem INDI Control Pane”l display, click connect at the bottom of the window and expand the mount:



From the xephem Sky View display:
Telescope > Configure …



In this window one can select “Enable Sky marker from” to be able to see the antenna position in the sky view with a 2 concentric circle marker:



5 System

The antenna drive control system runs on a VME processor, full PC compatible. It is a disk-less PC with a real-time Linux system. The following shows the different steps of the configuration of this system environment, i.e. the Linux kernel, the real-time or non real-time modules, the file system.

5.1 Fedora C3 installation

Connect a hard disk of a small capacity (e.g. a disk Quantum Fireball SE 3.2A of 3GB) and a CD drive to the SBC IDE controller.
Boot from the CDROM FC3-i386-disc1.iso Dec04 and install Fedora Core 3. Here after the installation dialog:

CD Found
    test: Skip

Language Selection
    English (default)

Keyboard Configuration
    U.S. English

Installation Type
    Install Fedora Core
    Workstation

Disk Partitionning Setup
    Manually partition with Disk Druid
    Delete all existing partitions
    Add with Disk Druid:
        - hda1: Mount Point:
            Size: 3079MB
            Partition Type: Linux ext3
            Format: yes

Network Devices
    no DHCP
    eth0 193.48.252.251/255.255.255.0
    Set hostname manually: rtaiabm1.iram.fr
        - Gateway: 193.48.252.2
        - First DNS: 193.48.252.22

FireWall configuration
    Enable firewall
    Allow incoming SSH

Enable SELinux: Active

Additional language support:
    English (USA)

Timezone Selection
    Europe/Paris

Set root password

Customize software package to be installed
remove Applications/ Office/productivity
        Applications/ Sound and Video

Applications/ Games and Entertainment

The size of the directory /, also called root, is 2.177GB for this installation.

Edit the sshd configuration (/etc/ssh/sshd_config) to authoriz/limit access.
Reinforce the firewall by editing its setting (/etc/sysconfig/iptables).

Modify the X11 server configuration in order to run gnome. In /etc/X11/xorg.conf, change the device section to add Option "noaccel"

```
Section "Device"
         Identifier  "Videocard0"
         Driver      "i810"
         VendorName  "Videocard vendor"
         BoardName   "Intel 815"
         Option      "noaccel"
EndSection
```

Tkinter is not installed in the default installation. This package is needed for developing the graphical user interfaces written in python. Get tkinter-2.3.4-11.i386.rpm from the Fedora distribution:
rtaiabm1 root:~ # rpm –ivh tkinter-2.3.4-11.i386.rpm

5.2 Disk less configuration

5.2.1 tftp/dhcp setting on netsrv1:

Transfer the kernel to netsrv1 used as tftp server. From rtaiabm1:

```
root@rtaiabm1 boot]# scp vmlinuz-2.6.9-1.667 root@netsrv1:/tftpboot
```

On netsrv1. configuration of pxe and dhcpd:
create /tftpboot/pxelinux.cfg/rtaiabm1.FC3

```
[root@netsrv1 root]# cat /tftpboot/pxelinux.cfg/rtaiabm1.FC3
label linux
  kernel vmlinuz-2.6.9-1.667
  append root=/dev/hda1 vga=775
```

vga=775 in order to have smaller characters on the console!!!

```
[root@netsrv1 root]# cd /tftpboot/pxelinux.cfg/
[root@netsrv1 pxelinux.cfg]# ln -s rtaiabm1.FC3 C130FCFB
```

C130FCFB is the IP address (193.48.252.251) in hexadecimal.

Add in /etc/dhcpd.conf:

```
host rtaiabm1 {
        hardware ethernet 00:20:38:01:43:E1;
        filename "pxelinux.0";
        fixed-address rtaiabm1.iram.fr;
        option routers 193.48.252.2;
        option subnet-mask 255.255.255.0;
        option domain-name-servers 193.48.252.22;
        option domain-name "iram.fr";
}
```

00:20:38:01:43:E1 is the MAC address of the VMIC Single Board Computer.

With this setting the SBC downloads its kernel and get its network configuration from netsrv1 but still uses /dev/hda1for its file system.

For a configuration based on a network file system (exported from the server pctcp101), a special kernel has to be built. It is done on pctcp101:
Information in http://download.fedora.redhat.com/pub/fedora/linux/core/3/i386/os/RELEASE-NOTES-en

Get from http://fedora.redhat.com/ kernel-2.6.9-1.667.src.rpm

```
[root@pctcp101 ~]# rpm -ivh kernel-2.6.9-1.667.src.rpm

[root@pctcp101 ~]# cd /usr/src/redhat/SPECS

[root@pctcp101 SPECS]# rpmbuild -bp --target=noarch kernel-2.6.spec

[root@pctcp101 ~]# cd /usr/src/redhat/BUILD/kernel-2.6.9/linux-2.6.9
```

There is .config which is a copy of configs/kernel-2.6.9-x86_64.config

```
[root@pctcp101 linux-2.6.9]# make xconfig
```

Seeting to build a kernel able to boot from the network:

Device Drivers
  Networking Support
    Networking options
      TCP/IP networking
        IP: kernel level autoconfiguration
          IP: DHCP support
  Network device support
    Ethernet (10 or 100Mbit)
      EISA, VLB, PCI and on board controllers
        Intel(R) PRO/100+ support
          Use Rx Polling (NAPI)
File systems
  Network File System
    NFS file system support
      Provide NFSv3 client support
      Root file system on NFS

```
root@pctcp101 linux-2.6.9]# make
[root@pctcp101 linux-2.6.9]# scp arch/i386/boot/bzImage
root@netsrv1:/tftpboot/bzImage-2.6.9.1.Net
[root@pctcp101 linux-2.6.9]# mkdir /home/rtaiabm1/lib/modules/2.6.9-prep
[root@pctcp101 linux-2.6.9]# rm -Rf /lib/modules/2.6.9-prep/
[root@pctcp101 linux-2.6.9]# ln -s /home/rtaiabm1/lib/modules/2.6.9-prep
/lib/modules/2.6.9-prep
[root@pctcp101 linux-2.6.9]# make modules_install
```

5.2.2 pctcp101 as NFS server

Transfer the SBC hard disk file system to the NFS server.
From rtaiabm1:

```
[root@rtaiabm1 /]# rsync -a --rsh=ssh --progress --exclude /proc --exclude
/home . pctcp101:/home/rtaiabm1
```

On pctcp101 modification of /home/rtaiabm1/etc/fstab

```
[root@pctcp101 ~]#  cat /home/rtaiabm1/etc/fstab
pctcp101:/home/rtaiabm1/ /                nfs    rsize=8192,wsize=8192 1 2
none                     /dev/pts         devpts gid=5,mode=620  0 0
pctcp101:/home/          /home            nfs    rsize=8192,wsize=8192 1 2
none                     /dev/shm         tmpfs  defaults              0 0
none                     /proc            proc   defaults              0 0
none                     /sys             sysfs  defaults              0 0
```

Add /proc:

```
[root@pctcp101 ~]# mkdir /home/rtaiabm1/proc
```

Add /home:

```
[root@pctcp101 ~]# mkdir /home/rtaiabm1/home
```

Change of the tftp config on netsrv1 to consider the new NFS kernel:
Create /tftpboot/pxelinux.cfg/rtaiabm1.FC3

```
[root@netsrv1 root]# cat /tftpboot/pxelinux.cfg/rtaiabm1.FC3.Net
label linux
  kernel bzImage-2.6.9.1.Net
  append root=/dev/nfs nfsroot=193.48.252.231:/home/rtaiabm1 vga=775
  ipappend 1

[root@netsrv1 root]# cd /tftpboot/pxelinux.cfg/
[pxelinux.cfg]# rm C130FCFB
[root@netsrv1 pxelinux.cfg]# ln -s rtaiabm1.FC3.Net C130FCFB
```

Reboot after changing the Bios (F2): Boot menu: Disable the hard Drive (!+Hard Drive) with Shift 1, Enable MBA UNDI (Bus0 Slot5) LAN 1, F10
Configuring MBA Ctrl+Alt+B, Boot Method: PXE

From now on, the SBC boots without hard disk , from the network with netsrv1 as dhcp/tftp server and pctcp101 as NFS server. The network file system corresponds to a Fedora Core 3 distribution and the kernel is a 2.6.9-1

```
[root@rtaiabm1]# rsync -a /home/rtaiabm1/lib/modules/2.6.9-prep
/home/rtaiabm1/lib/modules
```

Modification to allow normal reboot on diskless station:

```
rtaiabm1 root:~ # cd  /etc/rc0.d
rtaiabm1 root:/etc/rc0.d # rm K20nfs K75netfs S00killall
rtaiabm1 root:~ # cd  /etc/rc6.d
rtaiabm1 root:/etc/rc6.d # rm K20nfs K75netfs S00killall
```

5.3 Installation of rtai-3.2 with kernel 2.6.9

RTAI (Real Time Application Interface) is a hard real-time extension of the Linux kernel. Lot of documents may be found at http://www.rtai.org and http://www.aero.polimi.it/~rtai/ .
The New Generation PdB antenna drive software is based on this extension. The construction of the modules, the real-time extension of the Linux kernel plus all the user interfaces are described here after.

```
rtaiabm1 root:/home # mkdir adeos
```

Get rtai-3.2.tar.bz2 from http://www.rtai.org
Get linux-2.6.9.tar,bz2 from http http://www.kernel.org/pub/linux/kernel/v2.6

```
rtaiabm1 root:/home/adeos # tar xjf rtai-3.2.tar.bz2
rtaiabm1 root:/home/adeos # tar xjf linux-2.6.9.tar.bz2
rtaiabm1 root:/home/adeos # cd linux-2.6.9
rtaiabm1 root:/home/adeos/linux-2.6.9 # patch -p1  <../rtai-
3.2/base/arch/i386/patches/hal-linux-2.6.9-i386-r9.patch
patching file Documentation/adeos.txt
patching file Makefile
patching file adeos/Kconfig
patching file adeos/Makefile
patching file adeos/generic.c
patching file adeos/x86.c
patching file arch/i386/Kconfig
patching file arch/i386/kernel/Makefile
patching file arch/i386/kernel/adeos.c
```

```
patching file arch/i386/kernel/apic.c
patching file arch/i386/kernel/cpu/mcheck/p4.c
patching file arch/i386/kernel/cpu/mtrr/cyrix.c
patching file arch/i386/kernel/cpu/mtrr/generic.c
patching file arch/i386/kernel/cpu/mtrr/state.c
patching file arch/i386/kernel/entry.S
patching file arch/i386/kernel/i386_ksyms.c
patching file arch/i386/kernel/i8259.c
patching file arch/i386/kernel/io_apic.c
patching file arch/i386/kernel/irq.c
patching file arch/i386/kernel/process.c
patching file arch/i386/kernel/smp.c
patching file arch/i386/kernel/smpboot.c
patching file arch/i386/kernel/time.c
patching file arch/i386/kernel/timers/timer_pit.c
patching file arch/i386/kernel/timers/timer_tsc.c
patching file arch/i386/kernel/traps.c
patching file arch/i386/mm/fault.c
patching file arch/i386/mm/ioremap.c
patching file include/asm-i386/adeos.h
patching file include/asm-i386/io_apic.h
patching file include/asm-i386/mach-default/do_timer.h
patching file include/asm-i386/mach-default/irq_vectors.h
patching file include/asm-i386/mach-visws/do_timer.h
patching file include/asm-i386/mach-visws/irq_vectors.h
patching file include/asm-i386/msi.h
patching file include/asm-i386/pgalloc.h
patching file include/asm-i386/smp.h
patching file include/asm-i386/spinlock.h
patching file include/asm-i386/system.h
patching file include/linux/adeos.h
patching file include/linux/init_task.h
patching file include/linux/preempt.h
patching file include/linux/sched.h
patching file init/Kconfig
patching file init/main.c
patching file kernel/Makefile
patching file kernel/adeos.c
patching file kernel/exit.c
patching file kernel/fork.c
patching file kernel/panic.c
patching file kernel/printk.c
patching file kernel/sched.c
patching file kernel/signal.c
patching file kernel/sysctl.c
patching file mm/vmalloc.c

rtaiabm1 root:/home/adeos/linux-2.6.9 # make xconfig
Loadable module support
   [v] Module unloading
   [ ] Module versioning support (EXPERIMENTAL)
Processor type and features
   Pentium-Pro
   []Symmetric multi-processing support
Device Drivers/networking support>Networking support/Networking options
   [v] TCP/IO networking
      [v] IP: Kernel level autoconfiguration
         [x]IP: DHCP Support
Device Drivers/networking support>Network device support/Ethernet (10 or
100Mbit)
   [v] EISA,VLB,PCI, and on board controllers
      [v] Interl(R) PRO/100+ support
Graphics support
```

```
   [x] support for frame buffer device
      [x] VGA 16-color graphics support
      [x] VESA VGA graphics support
      [x] Intel 810/815 support (EXPERIMENTAL)
File system, Network File System
   [x] NFS file system support
      [x] provide NFSv3 client support
   [x]Root File System on NFS

rtaiabm1 root:/home/adeos/linux-2.6.9 # make
rtaiabm1 root:/home/adeos/linux-2.6.9 # make modules_install

rtaiabm1 root:/home/adeos/linux-2.6.9 # scp arch/i386/boot/bzImage
root@netsrv1:/tftpboot/bzImage-2.6.9.adeos
```

On netsrv1 /tftpboot/pxelinux.cfg:

```
[root@netsrv1 pxelinux.cfg]# ln -s rtaiabm1.2.6.9.adeos C130FCFB
```

Reboot

```
rtaiabm1 root:~ # uname -a
Linux rtaiabm1.iram.fr 2.6.9-adeos #1 Wed Jun 29 11:25:58 CEST 2005 i686 i686
i386 GNU/Linux

rtaiabm1 root:/home/adeos # mkdir build-3.2 && cd build-3.2
rtaiabm1 root:/home/adeos/build-3.2 # make -f /home/adeos/rtai-3.2/makefile
srctree=/home/adeos/rtai-3.2 xconfig
General/
        Dclick(Installation directory): /usr/realtime-3.2
        Dclick(Linux source tree) and enter /lib/modules/2.6.9-adeos/build

rtaiabm1 root:/home/adeos/build-3.2 # make
rtaiabm1 root:/home/adeos/build-3.2 # make install
```

Test:

```
rtaiabm1 root:/usr/realtime-3.2/testsuite/kern/latency # ./run
*
* Type ^C to stop this application.
*
## RTAI latency calibration tool ##
# period = 100000 (ns)
# avrgtime = 1 (s)
# check overall worst case
# do not use the FPU
# start the timer
# timer_mode is oneshot

RTAI Testsuite - UP latency (all data in nanoseconds)
RTH|    lat min|    lat avg|    lat max|   overruns|  freq.cntr
RTD|      -2343|        314|       9485|          0|      10000
RTD|      -2374|        246|      10109|          0|      10000
RTD|      -2374|        228|      10109|          0|      10000
RTD|      -2374|        228|      10109|          0|      10000

rtaiabm1 root:~ # ln -s /usr/realtime-3.2 /usr/realtime
```

5.4 Installation of the vme universe driver

This diver has been developed for RTAI from a version distributed by VMIC
The original version is in CVS PdB/LINUX/drivers/vme/vme_universe

```
rtaiabm1 root:~ # mkdir -p /home/devel/PdB
rtaiabm1 root:~ # export CVSROOT=:pserver:perrigou@netsrv1.iram.fr:/CVS/PdB
rtaiabm1 root:~ # cvs login
rtaiabm1 root:~ # cd /home/devel/PdB
rtaiabm1 root:/home/devel/PdB # cvs co LINUX

rtaiabm1 root:/home/devel/PdB # cd LINUX/drivers/vme/vme_universe_rtai
rtaiabm1 root:/home/devel/PdB/LINUX/drivers/vme/vme_universe_rtai # make install
rtaiabm1 root:/home/devel/PdB/LINUX/drivers/vme/vme_universe_rtai # cd test/iram_sync/
rtaiabm1 root:/home/devel/PdB/LINUX/drivers/vme/vme_universe_rtai/test/iram_sync # make

root:/home/devel/PdB/LINUX/drivers/vme/vme_universe_rtai/test/iram_sync (maybe create
/dev/rtf0: # ./create_node.sh)
# ./run     OK!!!!
```

5.5 Reorganization of the exported file system

This reoganization is mandatory to support several Single Board computers.

Copy the NFS root file system into /partition/nfsroot/common

```
[root@pctcp101 ~]# mkdir –p /partition/nfsroot/common
[root@pctcp101 ~]# cd /home/rtaiabm1
[root@pctcp101 rtaiabm1]# tar c * | (cd /partition/nfsroot/common/; tar x)
```

Special actions

- /partition/nfsroot/common/etc/hosts must contains all hosts (to setup the hostname)
- eth0 must not be reconfigured

  [root@pctcp101 ~]# cd /home/nfsroot/common/etc/syconfig/network-scripts
  [root@pctcp101 network-scripts]# mv ifcfg-eth0 old.ifcfg-eth0
- replace /etc/mtab by a symbolic link to /proc/mount
  [root@pctcp101 ~]# ln -s /proc/mounts mtab
- disable kudzu (needed RW permissions)
  [root@pctcp101 ~]# chkconfig kudzu off

/partition/nfsroot/common is exported in principle in Read Only access.

Creation of the Read Write directories, specific to each processor:
The mounted directories /var and /tmp are generated for each SBC by executing a script. For instance for rtaiabm1: [root@pctcp101 ~]# /partition/nfsroot/create_NFS_directories.sh rtaiabm1
Basically, for host=rtaiabm1, this script does

```
directories="tmp var"
for dir in $directories
do
        (cd common ; tar c $dir) | (cd $host ; tar x)
done
```

/partition/nfsroot/common/etc/rc.sysinit is edited to add at line 529 just after the root and /proc mounts:

```
echo “mount RW directories”
/mount_rw.sh
```

```
[root@pctcp101 ~]# cat /partition/nfsroot/common/mount_rw.sh
#!/bin/bash
host=`hostname`
nfsroot="pctcp101:/partition/nfsroot/"
directories="tmp var"
 for dir in $directories
do
        mount -n -o rw,rsize=8192,wsize=8192,udp,nolock $nfsroot/$host/$dir
$dir
done
```

On pctcp101 the export file is edited to contain the following:

```
[root@pctcp101 ~]# cat /etc/exports
…
/home/   rtaiabm1(rw,no_root_squash)
/partition/nfsroot/common/   rtaiabm1(ro,no_root_squash)
/partition/nfsroot/rtaiabm1   rtaiabm1(rw,no_root_squash)
```

On netsrv1 (dhcp/tftp server) creation of /tftpboot/pxelinux.cfg/nfsroot-rtaiabm1-2.6.9.adios to indicate the new file system:

```
[root@netsrv1 pxelinux.cfg]# cat nfsroot-rtaiabm1-2.6.9.adeos
# NFS ROOT for rtaiabm1 (exported from pctcp101)
label linux
  kernel bzImage-2.6.9.adeos
  append root=/dev/nfs nfsroot=193.48.252.231:/partition/nfsroot/common
ip=193.48.252.251:193.48.252.231:193.48.252.2:255.255.255.0:rtaiabm1:eth0
  ipappend 1

[root@netsrv1 root]# cd /tftpboot/pxelinux.cfg/
[root@netsrv1 pxelinux.cfg]# ln - nfsroot-rtaiabm1-2.6.9.adeos C130FCFB
```

5.6 Services dhcp/tftp/NFS servers at Bure

bure2 is the server dhcp/tftp and bure5 is NFS server for the SBCs.

On bure5:
Creation of a partition /dev/hda6 on the master disk and mount it to /nfsroot.

Copy pctcp101:/partition/nfsroot/common to bure5:/nfsroot/common .
Copy pctcp101:/home/adeos to bure5:/nfsroot/home/adeos .
Copy pctcp101:/partition/nfsroot/create_NFS_directories.sh to bure5:/nfsroot/create_NFS_directories.sh .

Add the addresses of the SBCs in /nfsroot/common/etc/hosts:

```
…
195.83.131.79        ant51.iram.fr ant51
```

Use the script /nfsroot/create_NFS_directories.sh to generate the RW directories. For instance for ant51:

```
[root@bure5 nfsroot]# ./create_NFS_directories.sh ant51
```

Edit /etc/exports

```
[root@bure5 root]# cat /etc/exports
…
/nfsroot/home/     antx51(rw,no_root_squash) antx62(rw,no_root_squash)
ant51(rw,no_root_squash) ant62(rw,no_root_squash)
/nfsroot/common/   antx51(rw,no_root_squash) antx62(ro,no_root_squash)
ant51(rw,no_root_squash) ant62(ro,no_root_squash)
/nfsroot/ant51   ant51(rw,no_root_squash)
```

Export the files systems.

On bure2:
Copy netsrv1:/tftpboot/bzImage-2.6.9.adeos to bure2:/tftpboot/

Creation of /tftpboot/pxelinux.cfg/ C353834F:
```
[root@bure2 root]# cat /tftpboot/pxelinux.cfg/C353834F
    # NFS ROOT for ant51 (exported from bure5)
    label linux
      kernel bzImage-2.6.9.adeos
      append root=/dev/nfs nfsroot=195.83.131.7:/nfsroot/common
    ip=195.83.131.79:195.83.131.7:195.83.131.1:255.255.255.128:ant51:eth0
      ipappend 1
```

dhcpd configuration
```
[root@bure2 root]# cat /etc/dhcpd.conf
…
host ant51 {
        hardware ethernet 00:20:38:01:43:E1;
        filename "pxelinux.0";
        fixed-address ant51.iram.fr;
        option routers 195.83.131.1;
        option subnet-mask 255.255.255.128;
        option domain-name-servers 193.48.252.22;
        option domain-name "iram.fr";
}
```

Restart the service dhcp

The SBC ant51 boots successfully.

6 Antenna Mount application

6.1 Cvs check out and installation

Under a non-privileged account having a CVS account on the CVS server antsrv1, download the application:
```
ant51 perrigou:~ $ mkdir -p devel/PdB/
ant51 perrigou:~ $ export CVSROOT= :pserver:perrigou@netsrv1.iram.fr:/CVS/PdB
ant51 perrigou:~ $ cvs login
ant51 perrigou:~ $ cd devel/PdB; cvs co LINUX
```

Under root:
```
ant51 root:~ # mkdir /home/introot; cd /home/introot
ant51 root: /home/introot # mkdir src bin include python
```

ant51 root:~ # cd ~perrigou/devel/PdB/LINUX/drivers/vme/vme_universe_rtai/moduleant51
root:/home/perrigou/devel/PdB/LINUX/drivers/vme/vme_universe_rtai/module # make
root:/home/perrigou/devel/PdB/LINUX/drivers/vme/vme_universe_rtai/module # cp vme_universe.ko ~perrigou/devel/PdB/LINUX/piedestal/antennaRTAI/src

ant51 root:~ # cd ~perrigou/devel/PdB/LINUX/piedestal/antennaRTAI/src
ant51 root:/home/perrigou/devel/PdB/LINUX/piedestal/antennaRTAI/src # make
ant51 root:/home/perrigou/devel/PdB/LINUX/piedestal/antennaRTAI/src # make install

6.2 Application start

ant51 root:~ # mkfifo /home/introot/share.fifo –m o+rw
ant51 root:~ # cd /home/introot/src
ant51 root:/home/introot/src # ln –s runinfo.antx .runinfo
ant51 root:/home/introot/src # ./run

run executes the script /usr/realtime/bin/rtai-load which, at its turn, installs kernel modules and linux tasks. The file /home/introot/src/.runinfo indicates the modules to install and the tasks to execute

```
ant51 root:/home/introot/src # cat .runinfo
antenna:ksched+fifos+shm+sem+math: push vme_universe_rtai; push common;
../create_node.sh; ../bin/init; ../bin/config; push it1s; push it64; push
evIt1s; ../bin/setTime; ../bin/tcpServer&; ../bin/subrefXmit&; ../bin/inclino;
popall: control_c
```

vme_universe_rtai, common, it1s, it64 and evIt1s are RTAI modules.
The tasks are in /home/introot/bin and are executed in foreground or in background. The background tasks are endless waiting for external events and their name is followed with the ampersand sign & in .runinfo. ../bin/inclino is the exception: It is also an endless loop triggered with RTAI FIFO semaphores every second.

When all the foreground tasks have been executed and the last endless taks, namely inclino has been started a kill of the task inclino will stop all the backgound tasks and will remove all the rtai modules.

In normal operation a ps, report process status, should display:

```
root      3879  3376  0 Dec09 ?        00:00:46 ../bin/tcpServer
root      3892  3376  0 Dec09 ?        00:00:09 ../bin/subrefXmit
root      3905  3376  0 Dec09 ?        00:00:25 ../bin/inclino
```

and the status of the kernel modules:

```
ant51 root:/home/introot/src # lsmod
Module                  Size  Used by
evIt1s                  7712  0
it64                    6236  0
it1s                    4724  1 evIt1s
common                  1348  3 evIt1s,it64,it1s
vme_universe_rtai     365616  4 it64,it1s
rtai_math              28196  0
rtai_sem               18336  2 evIt1s,it1s
rtai_shm                9312  1 common
rtai_fifos             38220  4 it64,it1s
rtai_up                82040  7
evIt1s,it64,it1s,vme_universe_rtai,rtai_sem,rtai_shm,rtai_fifos
rtai_hal               19732  4 vme_universe_rtai,rtai_sem,rtai_fifos,rtai_up
vmiwdtf                 4376  1
```

Application start at boot time:
/home/introot/initialize_antenna.sh is added at the end of /etc/rc.local, a script executed “after” all the other scripts.
In initialize_antenna.sh, the dedicated command lines for ant51 are:

```
if [ `hostname` == "ant51" ] ; then
        $INTROOT/src/rtai-load $INTROOT/src: &
fi
```

The command rtai-load execution installs all the RTAI modules and starts all the task as seen above in interactive mode.
The variable INTROOT is defined /home/root/.bashrc:

```
export INTROOT=/home/introot
```

6.3 RTAI kernel modules

Beside the rtai scheduler and the vme universe real-time driver, the antenna drive modules are common.ko, it64.ko, it1s.ko and evIt1s.ko

6.3.1 common.ko

This kernel module allocates a chunk of memory in kernel space that can be shared inter-intra kernel modules and Linux processes. In common.c the function rtai_kmalloc() does this allocation. The name of the allocated area is SHM_NAME defined as “ANTENNA” in the include file antenna.h.
In Linux processes, the allocation is performed with the function rtai_malloc().

6.3.2 it64.ko

This kernel module allocates different VMEbus master window handles and maps them to local memory (returns pointers). These handles define 16bit address short supervisory access and 24bit standard supervisory data access.
A real-time FIFO is created.
Finally a real-time task attached to the 64Hz interrupt (vector 0xF0 level 4) is created. Its priority is 5.
When the task is resumed, the task will accept azimuth and elevation position requests through the real-time FIFO and every 16ms will perform the following:

- Read the counter of the VME Sync board and save it into synchro.utCount.
- Do the axis position interpolation in case of remote requests.
- Call the function encoder() to read the absolute encoders or the incremental encoders when the later have been initialized.
- Call the function move() for each axis to calculate the command velocities, azimuth.vx and elevation.vx.
- Write these command velocities to the 16bits DAC.
- Call the function aux() that convert 2 internal variables of the shared memory area, available in analog form through the channels 3 and 4 of the VME 16bit DAC module. It is for debug purpose and the 2 internal variables can be selected, as well as the scaling factors and the offset levels.

6.3.3 It1s.ko

This kernel module allocates a VMEbus master window handle and maps it to the local memory (it returns a pointer). This handle defines 24bit standard supervisory data access.
2 real-time FIFOs are created.
Finally a real-time task attached to the 1Hz interrupt (vector 0xF1 level 4) is created. Its priority is 4, the highest of the application real-time task.
When the task is resumed, every second, the task will perform the following:

- Increment synchro.itSec.
- Prepare the axis position interpolation in case of remote requests.
- Signal the semaphore sem1s to the module evit1s
- Post (signal) a semaphore to the task setTime.
- Post (signal a semaphore to the task inclino.
- Read the input registers saved into io.in1/2/3/4.
- Read the status register of the incremental encoder VME modules.

6.3.4 EvIt1s.ko

The modules create a real-time task, which is triggered by the semaphore sem1s. The task priority is 6, the lowest of the application real-time task.
Every second, but with a priority which let the fast events (64Hz) interrupt its execution, the following operations are performed:

- Call tcoor() to calculate the astro direct transformation applying the refraction and pointing corrections.
- Call sunLimit() which prevents the antenna to enter inside the sun avoidance circle. sunlimit() calculates azimuth.xMax/xMin and elevation.xMax/xMin.
- Call dcoor() to calculate the astro inverse transformation: Right ascension/elevation or azimuth/elevation from the encoder readings.

6.4 Linux processes

These processes have already been introduced in the shell command paragraph.
The processes setTime and inclino are triggered by the real-time FIFO semaphores posted by the module it1s.ko.
setime needs this synchronization mechanism to request and obtain the current Universal Time in seconds only valid for this period of 1 second.
inclino is an endless process which reads the inclinometer tilts and temperature every second. As this information is transferred every second to the master computer, namely bure1, and as the tilts are correlated to the azimuth position, it was necessary to synchronize tilt reading, azimuth position and data transfer on the same event: the 1s pulse of the time bus.

6.5 Input/output signals

3 VME digital input/output boards interface the external signals. The boards referenced as E31 DE IDM32 are fully described in the document “VME Antenna Control” issued on October 3, 1995. Each board has 32 opto-coupled inputs and outputs.

Input register at address 0x20008 (A24D16)

in1/00			
in1/01	I_Vb	Verins de levee en position basse	Jacks: Low positon
in1/02	*I_Vr (1)*		*Jacks: Rest position*
in1/03	*I_reqB2 (1)*		
in104	*I_deftB2 (3)*	Defaut dans la baie 2: Absence 24V, disjoncteur off …	Cabinet 3 : Default
in1/05	I_Vh	Verins de levee en position haute	Jacks: High position
in1/06	*I_Ah (3)*	Antenne en position haute	Antenna up
in1/07	*I_Ch (3)*	Chariot en position haute	Transporter up
in1/08	I_Eph	Elevation: Pre-limite haute	Elevation: High pre-limit
in1/09	I_rmtB3	Cle baie 3 en position “distant”	Cabinet 3 in remote
in1/10	I_App	Azimut: Pre-limite positive	Azimuth: Positive pre-limit
in1/11	I_El	Elevation en position limite (haute ou basse)	Elevation : high or low limit
in1/12	I_ABU	Azimut: zone positive pour codeur	Azimuth : Positive from south
in1/13	I_Apn	Azimut: Pre-limit negative	Azimuth: Negative pre-limit
in1/14	I_Alp	Azimut: Limite positive	Azimuth: Positive limit
in1/15	I_Aln	Azimut: Limite negative	Azimuth: Negative limit

Input register at address 0x2000A (A24D16)

in2/00	I_sun	Alarme capteur solaire	Sun alarm
in2/01	I_Epb	Elevation: Pre-limite basse	Elevation: Low pre-limit
in2/02	I_po	Capteurs de portes cabine recepteur ouverte	Receiver cabin door sensor. 1: open
in2/03	I_ElB	Freins d'elevation	Elevation break on
in2/04	I_AzB	Freins en azimut	Azimuth break on
in2/05	I_Azlock	Status carte controle mouvement Az baie 3	Cabinet 3 status ??
in2/06	I_Ellock	Status carte controle mouvement El baie 3	Cabinet 3 status ??
in2/07	I_snow	Capteurs de collision du reflecteur	Snow sensor
in2/08	*I_pcns (1)*		
in2/09	I_dftB3	Defauts Baie 3: disjoncteurs, alimentations …	Cabinet 3 : Default
in2/10	I_gene	Groupe electrogene en marche	Diesel engine running
in2/11	*I_pcew (1)*		
in2/12	I_panic	Arret d'urgence general	General panic stop
in2/13	I_dftge	Groupe electrogene en defaut: Pression huile …	Diesel engine default
in2/14	I_disell	Niveau gas-oil bas dans la cuve	Diesel fuel low level
in2/15			

Input register at address 0x30008 (A24D16)

in3/00	*I_PBl (3)*	8 ancres verouillees sur station	Clamps locked
in3/01	*I_rmtB6 (1)*		
in3/02	*I_SZl (3)*	Survie zenith verrouillee	Locked zenith stow pin
in3/03	*I_Sau (3)*	Survie azimut deverrouillee	Unlocked azimuth stow pin
in3/04	*I_Sal (3)*	Survie azimut verrouillee	Locked azimuth stow pin
in3/05	*I_Szu (3)*	Survie zenith deverrouillee	Unlocked zenith stow pin
in3/06	*I_SHl (2)*	Survie horizon verrouillee	Locked horizon stow pin
in3/07	*I_Shu (2)*	Survie horizon deverrouillee	Unlocked horizon stow pin
in3/08	I_Pgene	Selection de puissance dans baie 5 sur groupe	Power on diesel
in3/09	I_PBu	8 ancres deverouillees et libres	Clamps unlocked
in3/10	I_dftB6	Defaut baie 6: Disjoncteur, alimentations ...	Cabinet 6 : Default
in3/11	I_Ptrck	Selection de puissance dans baie 5 sur voie	Power on track
in3/12	I_dftB5	Defaut baie 5: Disjoncteurs, alimentations ...	Cabinet 5 : Default
in3/13	I_ups	Onduleur en fonctionnement normal	UPS OK
in3/14	I_upsbat	Onduleur alimente par batterie	UPS on batterie
in3/15	I_upsbyp	Sortie onduleur sur "Bypass"	UPS on bypass

Output register at address 0x2000E (A24D16)

out1/00	O_El	Commande vers Baie 3: Voir BR !!!	Cabinet 3 : See BR !!!
out1/01	*O_okB2 (1)*		
out1/02	*O_Vxup (1)*		
out1/03	O_Az	Commande vers Baie 3: Voir BR !!!	Cabinet 3 : See BR !!!
out1/04	O_proj	Commande des projecteurs sequence manuelle 1 puis 0	External floodlit
out1/05	*O_Vxdn (1)*		
out1/06	*O_Vant (1)*		
out1/07	*O_Vcar (1)*		
out1/08	*O_Sa (1)*		
out1/09	*O_AzB (1)*		
out1/10	*O_ElB (1)*		
out1/11	*O_SH (1)*		
out1/12	*O_SZ (1)*		
out1/13	O_gene	Commande demarrage groupe electrogene	Diesel engine start
out1/14	*O_uB6 (1)*		
out1/15	*O_lB6 (1)*		

Output register at address 0x2000C (A24D16)

out2/00			
out2/01	*O_PB (1)*		
out2/02	*O_gener (1)*		
out2/03			
out2/04			
out2/05	O_Ptrck	Commande vers baie 5 pour selection source de puissance sur voie	Cabinet 5. Power from track selection
out2/06	O_Pgene	Commande vers baie 5 pour selection source de puissance sur groupe	Cabinet 5. Power from diesel selection

(1) signal introduced since 1999 and no more used (mail from PdB January 2006)
(2) signal introduced since 1999 and no more used (mail from PdB October 2008)
(3) signal introduced since 1999 and foreseen to be removed in summer 2009 (mail from PdB October 2008)