IRAM-COMP-037

Revision: 0
2008-01-03

Contact Author

Institut de RadioAstronomie Millimétrique

NTP server for PdB

Owner Alain Perrigouard (perrigou@iram.fr)

Keywords: ntp, pps

Approved by:	Date:	Signature:
A.Perrigouard	December 2006	

Change Record

REVISION	DATE	AUTHOR	SECTION/PAGE AFFECTED	REMARKS

Content

1 Introduction

As there is a high-precision GPS receiver on the Plateau de Bure (See http://www.iram.fr/IRAMFR/TA/backend/timefreq/pratique/index.html) it was obvious to set up a time server based on the Network Time Protocol (NTP) and synchronized on this GPS receiver.
Right now the GPS engine is used to reset a 1pps signal derived from a 5MHz signal generated either from a high quality quartz or from a maser. As time is going, the 1pps derivates from the GPS second and when the interval between both reaches 1ms, a reset is requested via a push button located at the back of the GPS receiver. The reset itself is triggered on the next GPS “one second” pulse.

The web site of the NTP project is http://www.ntp.org. There are plenty of information, an excellent documentation (copied to /computer: file:///computer/info/ntp-stable-4.2.0a-20060224/html/index.html) and the code of several reference clocks.

The NTP server for Bure is based on the clock driver refclock_true.c (clock type = 5. Radio and modem clocks by convention have addresses in the form 127.127.t.u, where t is the clock type and u is a unit number in the range 0-3 used to distinguish multiple instances of clocks of the same type).
The driver communicates with the GPS receiver TrueTime model XL-AK with a serial link at 9600bps. After init, the driver receives every second, timecode messages. However depending on the selected message types, their occurrence time may be not enough periodic to trigger a master clock. Hopefully there is the so-called “quartz or maser” digital 1pps signal already mentioned above and the GPS 1pps signal built directly from the reception of the satellites radio signals that can bring the missing accuracy. The second numbering, i.e. the second number since midnight, is directly based on the received timecode messages.

The NTP server is implemented on a Linux processor running kernel 2.6. In a native way the Linux kernel 2.6 does not provide high-precision time-keeping support. The RFC 1589 describes a model and programming interface that manages the system clock and timer functions. An implementations needs to be merged to our kernel, a version 2.6.15 already patched with RTAI-ADEOS (See file:///computer/doc/system/bure/pdbi-diskless-nfsroot.pdf).

During the implementation and the tests, the NTP server has been hosted on a VMIC 7700 microprocessor named rtaiabm1. This explains most of the command sequences beginning with the prompt:
`rtaiabm1 root:~ # …`
`On Plateau de Bure the micro has been renamed clockb.`

2 Linuxpps

The experimentation of the NTP server is done on a Single Board Computer VMIVME7700 called rtaiabm1. rtaiabm1 boots over the network using the protocol PXE handled by the server dhcp netsrv1. Its file system is available via NFS from pctcp102.

http://wiki.enneenne.com/index.php/LinuxPPS_support provides an implementation of an Application Programming Interface for the PPS facility.

The installation starts by copying a reference kernel distribution already patched for RTAI/ADEOS to pctcp102:~/linux-2.6.15.

Copy linuxpps-current.tar.gz from http://ftp.enneenne.com/pub/projects/linuxpps/Snapshot/ into ~/ntp.

```
pctcp102 perrigou:~/ntp $ tar xzf linuxpps-current.tar.gz

pctcp102 perrigou:~ $ cd ntp/linuxpps-current/contrib/linux-2.6/
```

```
pctcp102 perrigou:~/ntp/linuxpps-current/contrib/linux-2.6 $  ./ntp-
pps-2.6.15-rc7.sh ~/linux-2.6.15/
patching your kernel tree into "/home/perrigou/linux-2.6.15/"...
patching file drivers/Kconfig
patching file drivers/Makefile
patching file drivers/pps/Kconfig
patching file drivers/pps/Makefile
patching file drivers/pps/clients/Kconfig
patching file drivers/pps/clients/Makefile
patching file drivers/serial/8250.c
patching file include/linux/netlink.h
 done.
copying "pps.h" into "/home/perrigou/linux-2.6.15//include/linux/"...
done.
copying "timepps.h" into "/home/perrigou/linux-
2.6.15//include/linux/"... done.
copying "kapi.c" into "/home/perrigou/linux-2.6.15//drivers/pps/"...
done.
copying "pps.c" into "/home/perrigou/linux-2.6.15//drivers/pps/"...
done.
copying "procfs.c" into "/home/perrigou/linux-2.6.15//drivers/pps/"...
done.
copying "clients/ktimer.c" into "/home/perrigou/linux-
2.6.15//drivers/pps/clients"... done.
kernel patched! Now you can use your new LinuxPPS API! :)

pctcp102 perrigou:~ $ cd ~/linux-2.6.15
pctcp102 perrigou:~/linux-2.6.15 $ make xconfig
Device Drivers/PPS support
        [v]PPS support
                [v]PPS procfs support
                [v]PPS debugging messages
                PPS clients support
                [.]kernel timer client (Testing client, use for debug)
                [v]8250 serial support
Device Drivers/Character devices/Serial drivers
        [v] 8250/16550 and compatible serial support
                (2) Maximum number of 8250/16550 serial ports
Save and quit

pctcp102 perrigou:~/linux-2.6.15 $ make

pctcp102 perrigou:~/linux-2.6.15 $ su -c "make install; make
modules_install"

# cp /lib/modules/2.6.15/source/include/linux/timepps.h /usr/include
# cp /lib/modules/2.6.15/source/include/linux/timex.h /usr/include

pctcp102 perrigou:~/ntp/linuxpps-current/test $ make
```

The new exported file system is created with the script createNFSroot.sh (see file:///computer/doc/system/bure/pdbi-diskless-nfsroot.pdf) and the new kernel is copied to netsrv1:

```
pctcp102 perrigou:~ $ scp /boot/vmlinuz-2.6.15
root@netsrv1:/tftpboot/bzImage-2.6.15.adeos.pps
```

Boot rtaiabm1 and tests of the linuxpps API:
In the distribution there are ktimer and ppstest.
ktimer is a client which generates PPS signals by programming kernel timers.
ppstest tests the internal generated pps source. The source id is given as argument, here id = 2:

```
rtaiabm1 root:~ # insmod
/lib/modules/2.6.15/kernel/drivers/pps/clients/ktimer.ko

rtaiabm1 root:~ # cat /proc/pps/sources
id      mode    echo    name
----    ------  ----    ----------------
 0      1111    no      pps_8250_0
 1      1111    no      pps_8250_1
 2      1151    yes     ktimer


rtaiabm1 root:~ # /home/perrigou/ntp/linuxpps-current/test/ppstest 2
found PPS source #2 "ktimer" on ""
Assert timestamp: 1165460898.556162675, sequence: 186
Assert timestamp: 1165460899.556224675, sequence: 187
Assert timestamp: 1165460900.556287675, sequence: 188
Assert timestamp: 1165460901.556349675, sequence: 189
....
```

3 Timecode messages

The GPS receiver may be programmed to generate timecode messages on a serial port.
The function F08 requests to output the GPS Time-of year, once per second and well synchronized with the GPS second.
The function F28 requests to output event times. This events is the so-called “quartz or maser” 1pps. It is generated once per second but it is weakly linked to the GPS second.

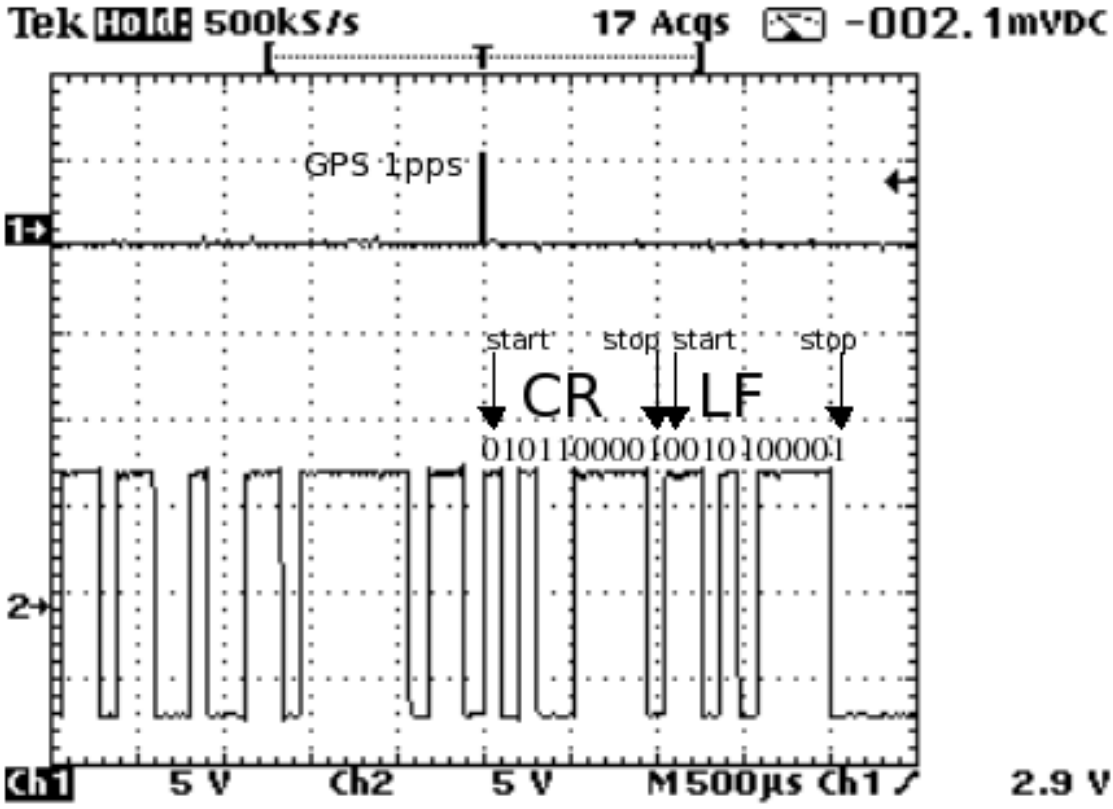
3.1 F08 timecode message

The format is
<SOH>DDD:HH:MM:SSQ<CR><LF>
Where
<SOH> : ASCII Start-of-header character (Hex 01)
<CR> : ASCII Carriage Return character (Hex OD)
<LF> : ASCII Line Feed character (Hex 0A)
DDD : day-of-year
HH : hours
MM : minutes
SS : seconds
Q : time quality character (“space” or “.” or “*” or “#” or “?”)

Example obtained with the communication program minicom on Linux. Serial port setup:9600bits/s, 8bits, no parity bit, 1stop bit.
F08
.341:16:55:28
.341:16:55:29
.341:16:55:30
….

On the scope one can check the following:
The 1pps pulse is 20 us width.
The message lasts about 17 ms.
The carriage return character <CR> start bit begins on the second which corresponds to the rising edge of the 1pss pulse with a delay measured between 10 us and 70 us.

These numbers are in agreement with the specifications in the TrueTime documentation XLAKMAN.DOC pages 1.6 and 3.43: The delay should be +0 to +1 bit time so less than 100us.



F08 message versus GPS 1pps

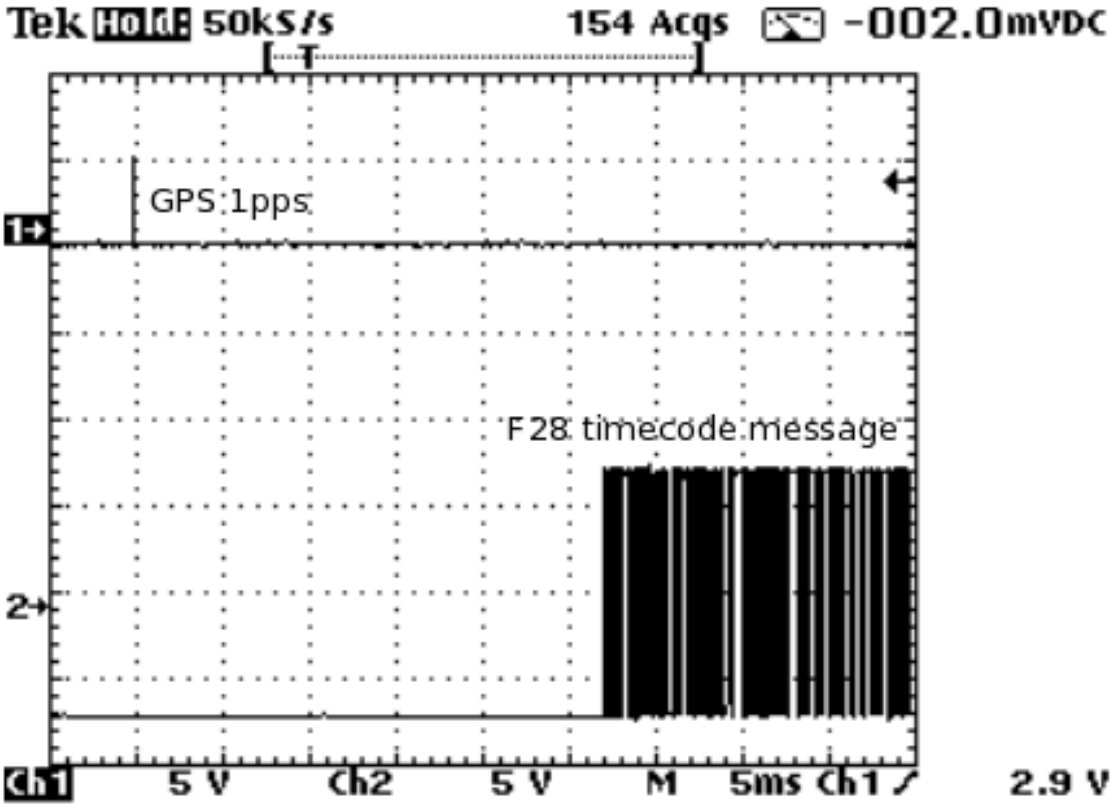
3.2 F28 timecode message

The format is
<SOH>DDD:HH:MM:SS.<NSEC>><CR><LF>
Where
<SOH> : ASCII Start-of-header character (Hex 01)
<CR> : ASCII Carriage Return character (Hex OD)
<LF> : ASCII Line Feed character (Hex 0A)
DDD : day-of-year
HH : hours
MM : minutes
SS : seconds
<NSEC>: 9-digit subsecond string

Example obtained with the communication program minicom on Linux. Serial port setup:9600bits/s, 8bits, no parity bit, 1stop bit.
F28
341:16:56:46.007844483
341:16:56:47.007844086
341:16:56:48.007843719
…

On the scope one can check the following:
The message lasts about 26ms.

The message occurs about 15ms after the “quartz or maser” 1pps pulse and the jitter is quite more important: +-5ms. Between the GPS 1pps and the F28 message there are 20ms+-5ms but just after a reset of the “quartz or maser” 1pps, i.e. when the fraction part of the event time in the F28 message is very close to 0 or 1 (<0.001 or >0.999).



F28 message versus GPS 1pps

The occurrence of this F28 message cannot be used directly to drive the NTP reference clock.
If this message is needed for monitoring the drift between the GPS second and the 1pps derived from the quartz or maser 5MHz, we need another precise digital signal for the NTP reference clock.
As the “maser or quartz” 1pps differs of less than 1ms from the GPS 1pps signal, any of these 2 pps signals may be used for the synchronization.
On the scope we saw also that the drift of one 1pps pulse versus the other is well continuous without any jitter or noise.
As the NTP server will be used for the observations and the synchronization between the different processors members of the interferometer, the so-called “maser or quartz” 1pps will be declared as reference digital signal and will be, in consequence, connected to the NTP server.

4 NTP server

The reference clock reads the timecode messages triggered by the TrueTime function F28. Those messages provide the “quartz or maser” 1pps event time but their occurrences are of no help for the reference clock. The same 1pps is used in conjunction with the PPS API. The 1pps is connected to the data carrier detect (DCD) line of the NTP server serial port. It is converted in RS232 levels (-12V, 12V) and at each pulse it is equal to 12V for one bit time (26us calculated for 38.4 kbps seems to work also at 9600bps). DCD is pin 1 and ground is pin 5 on a DB9 connector. The timecode messages, read on the receive line of the same serial

port, are used by the reference clock code just to number the 1pps events for which the PPS API provides the timestamp.

Get perrigou@pctcp00:ntp-stable-4.2.0a-20060224.tar.gz from http://www.ntp.org

```
pctcp102 perrigou:~ $ tar xzf ntp-stable-4.2.0a-20060224.tar.gz
pctcp102 root:~ # mkdir /usr/localAP
pctcp102 root:~ # chown perrigou:computer /usr/localAP
```

Modification of ~/ntp-stable-4.2.0a-20060224/ntpd/refclock_true.c to include the PPS support and to use the F28 timecode messages.

```
pctcp102 perrigou:~/ntp-stable-4.2.0a-20060224 $ ./configure --
prefix=/usr/localAP --disable-all-clocks --disable-parse-clocks --
enable-TRUETIME
pctcp102 perrigou:~/ntp-stable-4.2.0a-20060224 $ make
pctcp102 perrigou:~/ntp-stable-4.2.0a-20060224 $ su -c "make install"
>make.install
```

Configuration of ntpd:

```
rtaiabm1 root:~ # cat /etc/ntp.conf
server 127.127.5.0
fudge 127.127.5.0 stratum 0 flag4 1
# Enables writing of statistics records
statistics loopstats peerstats clockstats
# Indicates the full path of a directory where stats files should be
created
statsdir /tmp/
# Configures setting of generation file set names
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
# specifies the complete path and name of the file used to record the
frequency
of the local clock oscillator
driftfile /tmp/ntp.drift
```

Manual start of the ntp server with debug option:

```
rtaiabm1 root:~ # ln -s /dev/ttyS0 /dev/true0
rtaiabm1 root:~ # ln -s /dev/ttyS0 /dev/pps0
rtaiabm1 root:~ # /usr/localAP/bin/ntpd -D 2
```

4.1 Tests

4.1.1 Trace the NTP server

On each line, the fields are (left to right): the host name, the host stratum, the time offset between that host and the local host (as measured by ntptrace; this is why it is not always zero for "localhost"), the host synchronization distance, and (only for stratum-1 servers) the reference clock ID. All times are given in seconds:
Hereafter a steady trace with only one peer, the local reference clock, at least 2 hours after start up.

```
rtaiabm1 root:~ # ntptrace
localhost: stratum 1, offset -0.000055, synch distance 0.002812, refid 'TRUE'
```

4.1.2 Standard NTP query program

Print a list of the peers known to the server as well as a summary of their state. This is equivalent to the peers interactive command.
Information includes the address of the remote peer, the reference ID (0.0.0.0 if this is unknown), the stratum of the remote peer, the type of the peer (local, unicast, multicast or broadcast), when the last packet was received, the polling interval, in seconds, the reachability register, in octal, and the current estimated delay, offset and dispersion of the peer, all in milliseconds. The character at the left margin of each line shows the synchronization status of the association and is a valuable diagnostic tool. The encoding and meaning of this character, called the tally code.

* sys.peer
The peer has been declared the system peer and lends its variables to the system variables.

```
rtaiabm1 root:~ # ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*TRUETIME(0)     .TRUE.           0 l   59   64  377    0.000   -0.058   0.005
```

4.1.3 Monitoring files

Clockstats
Enables recording of clock driver statistics information.
The first two fields show the date (Modified Julian Day) and time (seconds and fraction past UTC midnight). The next field shows the clock address in dotted-quad notation, the final field shows the last timecode received from the clock in decoded ASCII format.

```
rtaiabm1 root:/tmp # more clockstats
54077 0.049 127.127.5.0 342:00:00:00.000039986
54077 1.049 127.127.5.0 342:00:00:01.000039711
54077 2.057 127.127.5.0 342:00:00:02.000039436
```

Loopstats
Enables recording of loop filter statistics information.
The first two fields show the date (Modified Julian Day) and time (seconds and fraction past UTC midnight). The next five fields show time offset (seconds), frequency offset (parts per million - PPM), RMS jitter (seconds), Allan deviation (PPM) and clock discipline time constant.

```
rtaiabm1 root:/tmp # more loopstats
54077 35.050 -0.000057000 64.468369 0.000002388 0.212985 6
54077 100.071 -0.000054000 64.465027 0.000002554 0.213206 6
54077 165.046 -0.000050000 64.461929 0.000002982 0.209565 6
```

Peerstats
Enables recording of peer statistics information.
The first two fields show the date (Modified Julian Day) and time (seconds and fraction past UTC midnight). The next two fields show the peer address in dotted-quad notation and status, respectively. The status field is encoded in hex in the format described in Appendix A of the NTP specification RFC 1305. The final four fields show the offset, delay, dispersion and RMS jitter, all in seconds.

```
rtaiabm1 root:/tmp # more peerstats
54077 35.067 127.127.5.0 9624 -0.000057000 0.000000000 0.000990000 0.000007161
```

```
54077 100.072 127.127.5.0 9624 -0.000054000 0.000000000 0.000975000 0.000004840
54077 165.047 127.127.5.0 9624 -0.000050000 0.000000000 0.000975000 0.000004706
```

Example of debug information saved in /tmp/true0.debug

```
rtaiabm1 root:/tmp # more true0.debug
true0: clock tm, state Base, event Init
'rue0: Send 'F28
true0: receive(341:16:56:54.007841428) [22]
true0: receive(341:16:56:55.007841061) [22]
true0: receive(341:16:56:56.007840664) [22]
```

In debug mode ntpd displays 2 important lines every second:

```
true_receive: rd_lastcod 345:17:00:08.998684261 1165856409.045173999
true: true_pps up->ts 1165856409.000190000
```

`345:17:00:08.998684261` is the F29 message
`1165856409.045173999` is the message occurrence sytem time. But there is quite a log of jitter. The fraction part is not stable
`1165856409.000190000` is the “quartz or maser” 1pps signal (DCD) occurrence system time. The faction part should be very stable.

5 Redundancy

In order to increase the redundancy and diversity of our NTP server, 2 remote servers of stratum 2 have been added to the configuration: ntp.imag.fr and ntp.obspm.fr. The TrueTime reference clock is declared “prefer”, an option that is useful to persuade the server to cherish a reference clock with somewhat more enthusiasm than other reference clocks or peers.
Here after the peers used as references for the remote servers added in our configuration.

```
# ntpq -p ntp.imag.fr
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*canon.inria.fr  .GPS.            1 u  229 1024  377   18.570   -1.957   3.070
+swisstime.ee.et swisstime2.ee.e  2 u  351 1024  377   25.220   -1.397   1.300
+dns.univ-lyon1. chronos.cru.fr   2 u  359 1024  377   11.250    0.489   0.720
 brahma.imag.fr  imag.imag.fr     3 u   15   64  255    0.580    0.215   0.690
+isis.imag.fr    canon.inria.fr   2 u   92 1024  375    0.340    0.111   1.010
 NTP.MCAST.NET   0.0.0.0         16 -    -   64    0    0.000    0.000 16000.0

# ntpq -p ntp.obspm.fr
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
-ntp-p1.obspm.fr .1PPS.           1 u  361 1024  377    9.013    1.302   0.626
-saturne.obs-bes .INIT.           1 u  178 1024  377   29.905   -3.757   1.915
+horlogegps.rese .GPS.            1 u  444 1024  377    4.798    0.323   0.321
-ntp-sop.inria.f .hopf.           1 u  197 1024  377   27.016   -1.578   2.139
*chronos.cru.fr  .GPS.            1 u  258 1024  377   15.051    0.182   1.287
+soleil.uvsq.fr  horlogegps.rese  2 u  321 1024  377    8.295   -0.114   9.645
 ntp.cri.u-psud. chronos.cru.fr   2 u  383 1024  377    4.198   -0.663  23.990
```

The new NTP server configuration:

```
pctcp102 root:/partition/nfsroot/common/etc # cat ntp-true.conf
server 127.127.5.0 prefer # local clock
fudge 127.127.5.0 stratum 0 flag4 1
server ntp.imag.fr      # IMAG Grenoble
server ntp.obspm.fr     # Observatoire de Paris
# Enables writing of statistics records
statistics loopstats peerstats clockstats
```

```
# Indicates the full path of a directory where stats files should be created
statsdir /tmp/
# Configures setting of generation file set names
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
# specifies the complete path and name of the file used to record the frequency
of the local clock oscillator
driftfile /tmp/ntp.drift
```

The firewall has to be updated to allow the NTP 123/UDP service.
The following is added to pctcp102:/partition/nfsroot/common/etc/sysconfig/iptables :

```
#
# trust ntp protocol
#
# ntp 123
#
-A RH-Lokkit-0-50-INPUT -p udp -m udp --dport 123 -j ACCEPT
```

rtaiabm1 has also to be added in the extended access-list “allowed_host” of the router iramr1.

To start manually the server:

```
rtaiabm1 root:~ # /usr/localAP/bin/ntpd -D 2 –c /etc/ntp-true.conf
```

And some time later:

```
rtaiabm1 root:/home/perrigou/ntp-stable-4.2.0a-20060224/ntpd #
/usr/localAP/bin/ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*TRUETIME(0)     .TRUE.           0 l   33   64  377    0.000   14.770   5.035
+imag.imag.fr    192.93.2.20      2 u  247 1024  377    1.980   56.048   4.065
+syrte8.obspm.fr 134.157.254.19   2 u  564 1024  377   14.385   50.492   5.834
```

TrueTime: system peers
Imag and obspm: candidates

Here the offset of the system peer is still mediocre. The quartz which produces the 5Mhz signal is very poor.

6 Troubleshooting

GPS receiver failure
If the reception of the GPS satellite signal disappears, the NTP server can continue to work for quite a long time. The F28 timecode messages will be still generated, using instead the internal TrueTime own quartz. However, the timecode messages will not provide anymore the information of the drift of the “quartz or maser” 1pps with respect to the GPS 1pps and the reset of the “quartz or maser” will be useless.

A way to force the NTP server to discard the TrueTime RS232 receive signal and the “quartz or maser” 1pps digital signal (RS232 DCD line), without modifying the configuration file, would be just to disconnect the serial line on the front panel of the VMIVME 7700 NTP server. The running NTP server would automatically, but after few minutes, discard the TrueTime local reference clock and would declare one remote server as system peer.

After a cold restart
Following an UPS failure or a cold restart of the GPS receiver, the RS232C line may be reset.
Either connect a VT100 type terminal to the RS232C socket at the back of the GPS TrueTime receiver or directy simulate such a VT100 terminal on clockb (under root) still connected to the RS232C line.

VT100 type terminal connection: Disconnect the RS232C line which links the 1PPS VME module to the RS232C socket at the back of the GPS receiver and connect instead a VT100. Use a crossed cable. The default setting should be 1200 bauds, no parity, 1 stop bit
From clockb with minicom: With the default configuration file minirc.dfl:

```
clockB root:/etc # more minirc.dfl
# Machine-generated file - use "minicom -s" to change parameters.
pr port             /dev/ttyS0
pu baudrate         1200
pr bits             8
pr parity           N
pr stopbits         1
pu rtscts           No
pu xonxoff          No
```

just run minicom :

```
pctcp00 root:~ # minicom
Welcome to minicom 2.00.0
OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Mar  7 2005, 10:29:09.
Press CTRL-A Z for help on special keys
Serial port setup  /dev/ttyS0 9600 8N1
```

Once connected(with VT100 or minicom), type CTRL+C to kill the possible running stream of data and then F28 followed by CR(or enter) to define the right stream. The sequence should look like:

```
. . .
080:15:21:11.999999877
080:15:21:12.999999877     (or any stream)
CTRL+C
OK
ERROR 05 NO SUCH FUNCTION   (eventually)
F28
080:15:25:51.999999877
080:15:25:52.999999877
```

Don’t forget to connect again the 1PPS VME module to the GPS RS232C socket at the back of the GPS receiver.