The Yebes PC-EVN:
Configuration and usage

P. de Vicente

Informe Técnico IT-OAN 2010-8

Revision history

Version	Date	Author	Updates
1.0	18-05-2010	P. de Vicente	First version

Contents

1 Introduction

This report summarizes the steps undertaken at the 40 m radiotelescope to make the PC-EVN work with Debian Lenny. This PC originally run under Debian Etch, and was configured by G. Molera from Metsähovi Observatory in 2007.

   The PC was upgraded to Debian Lenny and all software had to be reinstalled and tuned. This report is a summary of all operations performed to setup a working system and should be regarded as a collection of recipes compiled for future users to look up. Further information can be obtained from the Metsähovi web page:

```
http://www.metsahovi.fi/en/vlbi/vsib-tools/installguide
```
Most of the programs have been developed by A. Mujunen and J. Wagner.

2 The PC and the VSI-C card

The PC-EVN at Yebes is a Dell computer with two disks (350 Gb and 500 Gb), two 1 Gb ethernet cards and one VSI-B card manufactured by Metsähovi and installed in the PCI bus. The VSI-B card is a VSI-H interface which receives the VSI flux from the Mark5B working in cascade mode (Fig. 1)
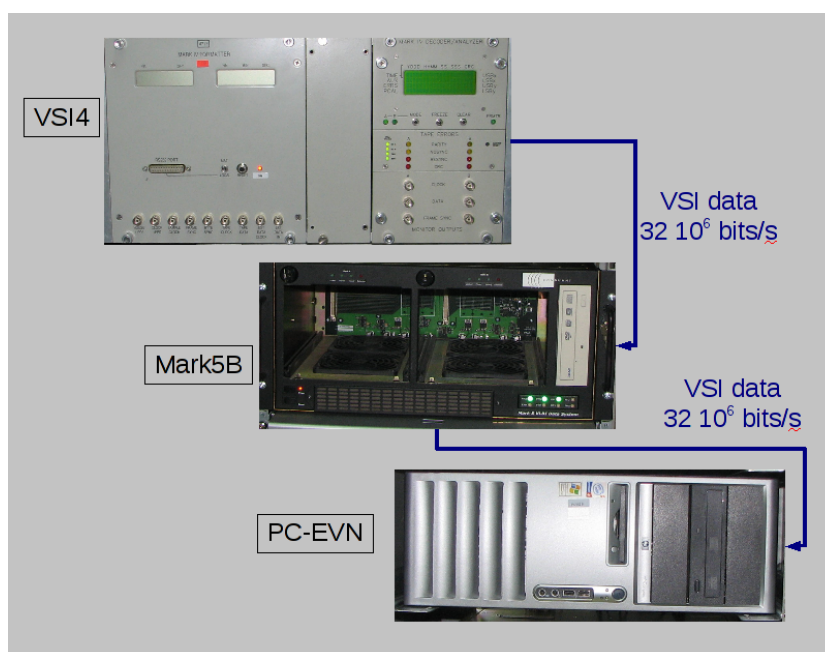

Figure 1: *Schematics of the connection between the VLBI terminal (VSI4 sampler), the Mark5B and the PC-EVN*

   Currently, the VSI4 module at the VLBI terminal has got two output VSI connectors. We only use one of the outputs, that one connected to the Mark5B. The latter also has an output connector that replicates the flow from the input connector. By using a VSI cable from the

Mark5B VSI connector to the PC-EVN and setting the Mark5B in cascade mode the data from the VLBI terminal can be recorded simultaneously at the Mark5B and at the PC-EVN. Data is sent at a rate of $32\,10^6$ bits/s. Two modes can be selected in the VSI4: "geo" and "vlba". The mapping is summarized below (Smythe 2004). Geodetic data only use 1 bit (sign) and the upper side band of 14 BBCs. Data in vlba mode use two bits (sign and magnitude) and both side bands for the first 8 BBCs.

The VSIB board captures the VSI flux and manages it in real time. However the PCI Bus is too slow for working with a sustained $32\,10^6$ sampling rate and it usually works with half the rate by discarding part of the samples or some channels. This behaviour is controlled from the VSIB utilities.

3   Installing the essentials: the linux kernel

Debian Lenny was installed in the PC following the standard procedure; that is, using a minimal installation from CD and retrieving all required software from the network. Since the PC has two ethernet cards we modified `/etc/network/interfaces` to provide the PC two IP addresses: one from the public network and one from the private one:

```
allow-hotplug eth0
iface eth0 inet static
    address 193.146.252.xx
    netmask 255.255.255.128
    network 193.146.252.0
    broadcast 193.146.252.127
    gateway 193.146.252.1
iface eth1 inet static
    address 192.168.0.yy
    netmask 255.255.255.0
    network 192.168.0.0
```

In the previous example we have taken out the IP address for security reasons.

In order to make the VSI-B card work in Debian, a Linux kernel driver prepared for the driver has to be installed in the system. Below we enumerate the steps followed to install such a kernel in the PC.

- Install packages: build-essential, libfftw3-dev, fftw-dev, zlib1g-dev, kernel-package, time, pgplot5, libpng3-dev, libx11-dev and cl-plplot.

- The standard way should be to install the following packages from Metsähovi:

```
wget http://www.metsahovi.fi/en/vlbi/vsib-tools/packages/kernel-lenny-pcevn/linux-h
wget http://www.metsahovi.fi/en/vlbi/vsib-tools/packages/kernel-lenny-pcevn/linux-i
wget http://www.metsahovi.fi/en/vlbi/vsib-tools/packages/kernel-lenny-pcevn/linux-s
wget http://www.metsahovi.fi/en/vlbi/vsib-tools/packages/kernel-lenny-pcevn/vsib.c
wget http://www.metsahovi.fi/en/vlbi/vsib-tools/packages/kernel-lenny-pcevn/vsib.ko
```

VSI output #1	Sampler output VLBA	Geo
0	1US	1US
1	1UM	1UM
2	2US	2US
3	2UM	2UM
4	3US	3US
5	3UM	3UM
6	4US	4US
7	4UM	4UM
8	5US	5US
9	5UM	5UM
10	6US	6US
11	6UM	6UM
12	7US	7US
13	7UM	7UM
14	8US	8US
15	8UM	8UM
16	1LS	1LS
17	1LM	1LM
18	2LS	8LS
19	2LM	8LM
20	3LS	9US
21	3LM	9UM
22	4LS	10US
23	4LM	10UM
24	5LS	11US
25	5LM	11UM
26	6LS	12US
27	6LM	12UM
28	7LS	13US
29	7LM	13UM
30	8LS	14US
31	8LM	14UM

Table 1: *VSI4 sampler to VSI bit stream mapping. Geo mode only uses 1 bit (sign)*

and install them using `dpkg`:

```
dpkg --install linux-source-2.6.26-pcevn_2.6.26-pcevn-10.00.Custom_all.deb
dpkg --install linux-headers-2.6.26-pcevn_2.6.26-pcevn-10.00.Custom_i386.deb
dpkg --install linux-image-2.6.26-pcevn_2.6.26-pcevn-10.00.Custom_i386.deb
```

but the PC has got a Broadcom Corporation NetXtreme BCM5751 card which uses module *tg3* which is not compiled in the kernel supplied by Metsähovi. So we retrieved the kernel sources and recompiled it adding support for the Broadcom card.

- Download package linux-source-2.6.26

```
apt-get install linux-source-2.6.26
```

- Apply the "bigphysarea" patch to the kernel. This patch provided by Metsähovi, is a driver which allocates a big piece of memory during boottime and returns the entire piece or parts of it to a requesting driver:

```
wget http://www.metsahovi.fi/en/vlbi/vsib-tools/bigphysarea/bigphysarea-2.6.26.diff
mkdir /usr/src/bigphysarea
cp /home/oper/downloads/bigphysarea-2.6.26.diff /usr/src/bigphysarea/
cd /usr/src/linux
patch -p1 < /usr/src/bigphysarea/bigphysarea-2.6.26.diff
cp /boot/config-2.6.26-2-686 .config
make oldconfig
make menuconfig
make-kpkg clean
make-kpkg kernel-image --initrd --append-to-version=vsib
```

In the previous lines the patch plus the standard configuration for Debian kernels is applied. The latter is achieved by copying file `.config`.

- The patch should be loaded during boot time and hence file `/boot/grub/menu.lst` requires to be modified accordingly:

```
default = 2
/boot/vmlinuz-2.6.26vsib root=/dev/sda1 ro bigphysarea=36000
```

- Restart the PC with the new kernel

4   Installing the VSIB driver and related software

Once the PC is running the new kernel with the bigphysarea patch, we compile the VSIB driver provided by Metsähovi. The required instructions should be:

```
cd /home/oper/downloads
wget http://www.metsahovi.fi/en/vlbi/vsib-tools/packages/vsib_drivers/vsib_src_12may08.t
mkdir /usr/src/vsib
cd /usr/src
tar xzvf vsib_src_12may08.tar.gz
```

but that version uses an old vsib.c, that does not work with kernels more modern that 2.6.24, so a new one has to be downloaded:

```
cd /usr/src/vsib
wget http://www.metsahovi.fi/en/vlbi/vsib-tools/packages/kernel-lenny-pcevn/vsib.c
```

Since Debian Lenny does not supply g77, the compiler should be replaced by gfortran. Modify file `/usr/src/vsib/Makefile-tools` so that where we get

```
FL = g77
```

is replaced by:

```
FL = gfortran
```

Finally execute:

```
cd /usr/src/vsib
./yank26
```

to get all software compiled.

Since the VSIB module should be ownerhipped by the "oper" account and group "vsib", permissions should be set by modifying `/etc/udev/permissions.rules`, so that the device gets the correct permissions after each reboot:

```
# vsib board
KERNEL="vsib", MODE="0664", GROUP="oper"
```

And the module should be loaded automatically when booting the computer:

```
# automatically load MRO VSIB board driver, 144MB bigphysarea
vsib bigbufsize=144000000
```

While testing, we changed read/write permissions,

```
chmod a+rw /dev/vsib
```

Data from the VSIB board will be written on directory `/data`. This directory is the mount point for a 500 Gb SATA disk, although an optimum solution show be to have a RAID system composed of several large disks. Below is the output from command `df`:

```
/dev/sdb1           480719056 141832632 314467224   32% /data
```

Permissions for directory `/data` and subdirectory `/data/t` (reserved for tests) were changed as below:

```
mkdir /data/t
chmod a+w /data
chmod a+w /data/t
```

and file `/usr/src/vsib/vsib-test` should contain the following line:

```
export VSIRAIDMOUNTPOINT=/data
```

Some binaries could be copied to an alternate directory where they can be used from any account provided that the PATH contains `/usr/local/bin`:

```
cp wr /usr/local/bin
cp rtspectrum /usr/local/bin
```

5 Software spectrometers

One of the advantages of sampling data at a rate of $32\,10^6$ bits/s is that it is possible to make an FFT on a short time interval and obtain spectra of several bandwidths. Two different schemes have been implemented in the PC-EVN (see Mujunen 2006, and Wagner 2009). One of the methods is offline and requires recording data on a file. The file is processed later and the spectrum obtained. The second method is to obtain a real time spectrum while the data are being acquired.

Applications `swspectrometer` (offline), `rtspectromter` (real time) and `vsi4spec` (real time) require the INTEL compiler to perform the FFT. `rtspectrometer` is bundled with the VSIB module software.

First we install the Intel Fortran Compiler. The tar file is available from the Intel webpage once a user is registered. Let us suppose we already downladed the tar file.

```
l_ipp_ia32_p_5.3.4.080.tgz
mv l_ipp_ia32_p_5.3.4.080.tgz /home/oper/downloads/
cd /home/oper/downloads/
tar xvf l_ipp_ia32_p_5.3.4.080.tgz
cd l_ipp_ia32_p_5.3.4.080
./install.sh
NLJG-P6FZH6TF  (serial number)
```

To make it available for the oper account we modify `.bashrc`:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:"/opt/intel/ipp/5.3.4.080/ia32/sharedlib/"
export IPP_PATH="/opt/intel/ipp/5.3.4.080/ia32"
```

`swspectrometer` is downloadable from
http://www.metsahovi.fi/en/vlbi/spec/swspectrometer .
Compile as follows:

```
cd cellspe-tasklib/spectrometer
apt-get install autoconf
make
cp src/intel_swspectrometer /usr/local/bin
cp src/swspectrometer /usr/local/bin
```

`vsi4spec` software is downloadable from:
http://www.metsahovi.fi/en/vlbi/vsib-tools/vsi4spec_sources/ .
Compile as follows:

```
cp -r vsi4spec_v10 /home/oper/
cd /home/oper/vsi4spec_v10
make
cp vsi4spec /usr/local/bin
```

This application has been patched by the author to integrate several samples and write amplitude for each channel on an ASCII file.

6   Transfering geo observations via Internet

The geodetic EURO experiments can be currently run in a sort of eVLBI mode. The data are recorded in disk paks and simultaneously stored in the PC-EVN disk, and when the experiment has finished, they are sent via Internet to the Bonn correlator. This operation requires several tools whose installation will be described here.

The philosophy of the recording is as follows:

- The SNAP schedule is transferred from the FS PC to the PC-EVN.

- The schedule is read by an application that creates a bash script file.

- The script triggers some commands at the correct time to create one file per scan with the correct sample rate. Each file has a standard Mark5B header that will allow the correlator to read it.

- Once the experiment has finished, all files are retrieved from Bonn correlator, via Internet using tsunami, an optimized fast UDP protocol.

Tsunami is a file transfer protocol, running in the user space, that uses TCP control and UDP data to transfer over very high speed long distance networks (> 1 Gbps) . It is designed to provide more throughput than TCP over the same network. It was originally developed by the Indiana University but lately it is maintained and improved by Aalto University Metsähovi Radio Observatory.

The version installed in Yebes was obtained directly by G. Molera who provided an updated tar file since the official site in sourceforge seems to be outdated. It was installed in a standard way:

```
cd /home/oper/tsunami-udp
make
su
make install
cd util
tar xzvf rectools.tar
```

`makeRecexpt.sh` and `makeRecexpt-tsunami.sh` are two scripts provided with the tsunami code, that create bash scripts to record data captured from the VSIB card. The first one creates a script file that will record the data on the local disk. The second one transfers the data to another host. The script files created have the same name of the experiment (for example `recexpt_EUR105_Ys.sh` and contain a list of all scans inside. This script creates one file per scan following a name convention in which the experiment name and the start date and time form the name of the data file. This filename is passed as argument to a procedure called `recpass`.

`recpass` calls `wr`, a C program that reads the VSIB card, as requested by the user, and writes the data on a file. The user may select the mode, the channels, the flow rate and the number of blocks to read and write. It is possible to only read some channels and skip samples. However this program does not produce valid Mark5B headers. In order to produce Mark5B compatible files, J. Wagner wrote `wr_mk5b`. This C program was obtained privately and installed in the PC-EVN:

```
cd /usr/src/vsib
gcc wr_mk5b.c -lm -o wr_mk5b
cp wr_mk5b /usr/local/bin
```

Therefore it was necessary to patch `recpass` so that it uses the `new wr_mk5b`. Below we show the applied modification:

```
echo "/usr/local/bin/wr_mk5b --rate=128 ${raiddir}/${filenm}.evn ${SECS}"
/usr/local/bin/wr_mk5b --rate=128 $raiddir/$filenm.evn $SECS
#echo "/usr/bin/wr 80000 ${TOTALBLKS} 0 0 0 0 1 ${raiddir}/${filenm}.evn ${TOTALBLKS} <
#/usr/local/bin/wr_mk5b 80000 $TOTALBLKS 0 0 0 0 1 $raiddir/$filenm.evn $TOTALBLKS < /de
```

The original lines are the last two commented and the new ones are the first two.

`wr_mk5b` is triggered at the beginning of each scan. In order to be on time, the PC-EVN needs to be previously synchronized with NTP. `recpass` runs a command that delays the recording in the required ammount by reading the name of the file. The file name follows a given convention which contains the start date and time. For example:

```
euro105_139-1800_2010y139d18h00m00s
```

`dtstart2` is the small application that delays commands. It is customized so that the time to sleep is obtained from the file name that is passed as argument. The new version was provided by G. Molera. It was compiled and installed in a standard way:

```
cd /usr/src/vsib
gcc dstart2.c -lm -o dstart2
cp dstart2 /usr/local/bin/
```

7 An useful tool: fusemark5

`fuseMark5` is a read-only filesystem for the Conduant StreamStor disk recorders. It allows to mount the recorded disk packs as mounted disk and access all scans as files. It is extremely useful since it allows to copy individual scans from the disk pack to anywhere else. For example all scans could be transferred to any correlator using tsunami or regular scp.

`fuseMark5` needs to be installed in the Mark5B. The last available version is 1.0.4 which works correctly with Mark5B and that can be downloaded from:

```
http://sourceforge.net/projects/fusemk5a/files/fuseMk5A/fuseMk5-v1.0.4/fuseMk5A-v1.0.4.t
```

In order to install it, some packages should be installed first: fuse-utils, libfuse-dev and libfuse2.

```
apt-get install fuse-utils libfuse-dev libfuse2
```

To install it just untar and make:

```
mkdir /home/oper/fuseMark5
cd /home/oper/fuseMark5
wget http://sourceforge.net/projects/fusemk5a/files/fuseMk5A/fuseMk5-v1.0.4/fuseMk5A-v1.
tar xvf fuseMk5A-v1.0.4.tar
cd fuseMk5A-v1.0.4
make
rm -f fuseMk5A.o maxspeed.o fuseMk5A maxspeed fuseMk5A-v1.0.4.tar
g++ -O3 -Wall -D_LARGEFILE64_SOURCE=1 -D_FILE_OFFSET_BITS=64 -DFUSE_USE_VERSION=22 -D_RF
stor/linux//driver/lib/ -L/usr/local/src/streamstor/linux//lib/ -L/usr/local/src/streams
g++ -O3 -Wall -D_LARGEFILE64_SOURCE=1 -D_FILE_OFFSET_BITS=64 -DFUSE_USE_VERSION=22 -D_RF
stor/linux//driver/lib/ -L/usr/local/src/streamstor/linux//lib/ -L/usr/local/src/streams
```

8   Usage cases. Examples

Below we provide some examples of usage of the tools described in previous sections.

8.1   Using the real time spectrometer

The real time spectrometer displays the FFT from a selected channel in real time on a graphical window. The data get accumulated for a given time while the graph gets displayed. This means that while average takes place the noise decreases. Once the number of stacked FFTs has been reached the old ones are thrown away while the new ones come into average.

The application is started from the shell. It is possible to select the FFT size, number of averaged FFTs, mode, and number of samples to skip from the command line. These values are provided as parameters:

```
/rtspectrum samplerate fftsize fftstacklen channel mode skip giga [custom mode string cm
```

where:

- samplerate is the rate at which data always flow. In the VSIB card this is 32 MHz.

- fftsize is the number of points of the FFT. Half of them will be thrown away.

- fftstacklen is the number of FFTs accumulated to get the power spectrum. This value is related to the integration period as follows:

$$l = \frac{2 \, b \, 10^6 t}{s} \tag{1}$$

where t is the integration time in seconds, b the bandwidth in MHz and s the fft size.

- channel is the channel that will be displayed.

- mode is the mode to be used. There are several modes described by Mujunen 2006, which are programmed in the VSIB card firmware. We list here only the first modes:

  - 0: all32bits. This is all the VSI data and the MK5 32-bit mode
  - 1: even16bits. This is the 16-bit mode or all sign bits from a VLBA sampler
  - 2: low16bits. This is either 2-bit USB from 8 BBCs or 2-bit USB+LSB from 4 BBCs.
  - 3: low8bits. This is either 2-bit USB from 4 BBCs or 2-bit USB+LSB from 2 BBCs.
  - 4: loweven8bits. This is either MK5 8-bit mode or 8 sign bits from VLBA sampler

- skip is the number of samples to skip. The used sample rate is corrected taking into account this value. This value has to be chosen depending on the used BBC bandwidth. If the bandwidth is 8 MHz, skip should be 1, if bandwidth is 4 MHz, skip should be 3, if bandwidth is 2 MHz, skip should be 7.

- giga is to enable/disable gigabit mode

- cms is a customized bit string format.

Below we show a typical example usage:

```
rtspectrum 32 4096 15625 0 2 1 0 w16b1,0
```

which shows the power spectrum from BBC 1 at a bandwidth of 8 MHz and for an integration time of 4 seconds (see Fig. 2.
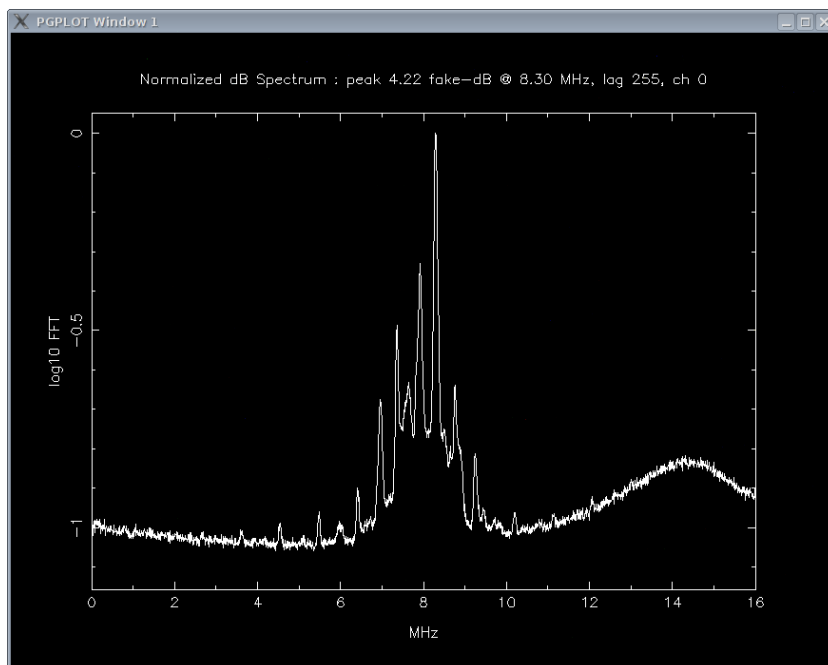

Figure 2: *Spectrum of the water line at 22 GHZ towards Orion*

We have tested this application and we believe there are two small bugs at the time of this report when using it with a mark5B equipment:

- Labels in the X axis are wrong and all units should be divided by 2.

- Some artificial features appear in the spectrum.

8.2   Using vsi4spec

`vsi4spec` is a real-time spectrometer software written by J. Wagner. It captures 4, 8 or 16-channel 2-bit data directly from the VSIB card. This application integrates data for an arbitrary period and computes the FFT of that data. Data are written in binary files. The author of this report patched `vsi4spec` to write the power spectrum of the FFT on an ASCII file. Usage of the patched application is simple:

```
vsi4spec fftsize bandwidth integTime channels totalIntTime
```

where parameters are:

- fftsize. Size of the FFT. Half of the channels will be discarded. It should be a power of 2.

- bandwidth. Bandwidth of the channel in MHz. It should match the BBC bandwidth.

- integTime. Integration time used to do one FFT [seconds]

- channels. Number of channels. Only 4, 8 and 16 are allowed.

- totalIntTime. The total integration time in seconds. FFTs are averaged for this time interval.

For example, the command below produces the result in Fig. 3:

```
./vsi4spec 65536.0 8e6 1 8 16
Integrating 244 of 65536-point FFTs which corresponds to 0.999424 seconds, Skipping 1 of
VSIB: mode 2, skip 1
```

After the total integration time has ellapsed file `vsi4spec.dat` is created with several columns. Each column corresponds to a channel.

In order to test the application we injected an artificial signal into the first BBC which differs 4 MHz from BBC01 LO. The line should appear at the center of the channel since BBC01 is set with a bandwidth of 8 MHz. The line appears in the FFT in column 4 as a Dirac delta with a resolution of 3.9 KHz. We have also observed that when the phase cal is activated the pulses are seen along the band (see Fig. 3)

8.3 Usage of tsunami for EURO geodetic experiments

Previous to starting the experiment it whould be checked that directory `/data` is empty. If it not were the case, remove all files there.

Once the EURO schedule is created with `drudg` it should be transferred to the PC-EVN and the recording script be created. Let us assume we are in the PC-EVN:

```
cd /home/oper/tsunami-udp/util
scp oper@fspc:/usr2/sched/eur105ys.snp .
./makeRecexpt.sh /home/oper/scheds/eur105ys.snp EUR105 Ys
```

File `recexpt_EUR105_Ys.sh` will be created. This file may be executed at any moment. Once it is executed it will sleep until the time has come to start. For example:

```
Next scan in list: EUR105_Ys_139-1800_2010-05-19T18:00:00

DEBUG: interpret_as_utc(2010, 0 days, 5, 19, 18h, 0m, 0s) : fixed_mktime()=1274292000.00
Detected time format: ISO basic extended
33191.17 seconds until 'EUR105_Ys_139-1800_2010-05-19T18:00:00.evn', sleeping 16595.59 s
```

Once the experiment is over the Bonn correlator should be informed to remotely start tsunami served and the download of the experiment files.
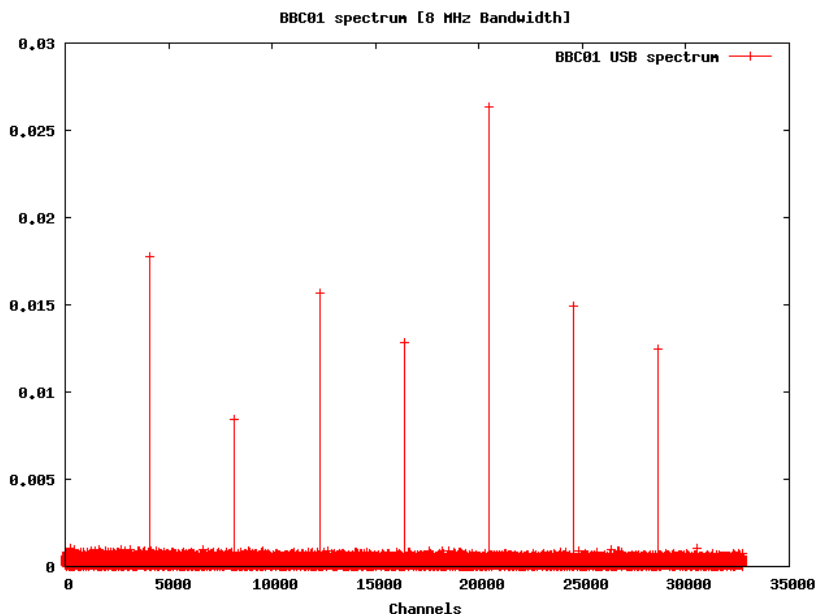
Figure 3: *Spectrum of noise from BBC01 at C band with 32768 channels. The pulses from the phase cal are separated 1 MHz. Y axis is in arbitrary units. Each channel is 244 Hz wide*

8.4   Usage of fuseMark5 for fringe tests

`fuseMark5` has been used to transfer files from the first mm fringe test in which the 40 m radiotelescope took part.

To use it it is important to stop `dimino` or `jive5a` previously. Then as root one should create a mount point and run `fuseMark5` as we show below.

```
su
mkdir /mnt/diskpack/
fuseMk5A --verbose /mnt/diskpack
fuseMk5A version 1.0.4
Bank selected  : 0
Prefetch cache : 15990784
Checking that port 2620 is closed, 'Mark5A' not running...
Opening diskpack...
Selected bank 0. It has label 'USN-0006/960/1024Recorded'
Diskpack: set of 16 disks, contains 271 scans, 0 duplicate scan names were renamed.
Registering fuseMk5A to FUSE with: fuse_main(5, fuseMk5A -f -odirect_io -oallow_other /m
```

Files in the mount point `/mnt/diskpack` can be copied using tsunami or transferred to another host.

To unmount the disk packs, it is necessary to type from a different shell:

```
fusermount -u /mnt/diskpack
```

or abort `fuseMark5` by typing Ctrl-C in the first shell.

References

[1] D. Smythe, "VSI Interface Board for the Mark4 Formatter", Mark5 Memo 16, 2004.

[2] A Mujunen, "Using VSIB PCEVN software tools", 2006.

[3] J. Wagner, "Brief Summary Of Preconfigured Metsähovi System Disk For A VSIB PC", 2006

[4] J. Wagner, "Quick Install Guide And Cheatsheet For Setting Up A VSIB PC", 2006

[5] J. Wagner, "Channelizer monitoring with VSI4SPEC software", 2009