SEISLOG for Linux

Seismic Data Acquisition System

Version 3.0

Terje Utheim

Department of Earth Science
University of Bergen
Allegaten 41, 5007 Bergen
Norway

Ph. +47 55583408
Email: terje.utheim@geo.uib.no

January 2011

Copyrights.

1 INTRODUCTION

The original SEISLOG seismic data acquisition system for the QNX operating system was developed in 2001 (Utheim and Havskov, 1999, Utheim et al., 2001).
The QNX version of SEISLOG is no longer supported. A new version running under Linux has been developed. Most of the options from the QNX version has been implemented in the Linux version plus a number of new options.

This version of SEISLOG is designed to be able to work on standard desk top PCs , laptops and small embedded systems. This also means that it must be able to work without a console screen connected permanently.
For this reason there is no complete graphical user interface software for configuring this version of SEISLOG. All configuration files can be modified via the network with a standard text editor.
Some monitoring functions can be done locally or remotely through client-server utilities.
The source code for the SEISLOG Linux embedded version is identical to the PC desktop version. However, the distribution for the SEISLOG Linux embedded is included as binaries to make installation easy for the user. The SEISLOG Linux embedded runs on the TS-7260 from Technologic Systems (http://www.embeddedarm.com).
Installation and use is described in a separate section in this manual.

2 INSTALLATION

SEISLOG can be installed on a standard installation of Linux. It has been tested on different distributions of Linux.

Pre-installation:

The **GNU C** and the **ncurses** development library must be present before compiling.

STEP 1:

First create a user account for SEISLOG with a username and directory name less or equal to 10 characters. We will use **seismo** as an account throughout this manual.

Username: **seismo**
Password: selected by the user.
The account must be of shelltype: **tcsh**.

You may select another account name, but the default through SEISLOG is seismo and is recommended. This will create a home directory: /home/seismo.
Log into this account to start the installation.

STEP 2:

SEISLOG is distributed as a SEISLOGddmmmyy.tar.gz file, where dd is day, mmm is three first letters in month and yy is year.
Download the distribution file to your SEISLOG account home directory, and unpack the distribution:

tar –xzvf SEISLOGddmmmyy**.tar.gz**

If your home directory is /home/seismo, this will create the following directory structure:

/home/seismo/SLG
/home/seismo/SLG/COM
/home/seismo/SLG/EMB
/home/seismo/SLG/EVT
/home/seismo/SLG/LOG
/home/seismo/SLG/PAR
/home/seismo/SLG/PRO
/home/seismo/SLG/RNG

SEISLOG normally reads digitised data from a serial port. The serial ports on Linux systems are normally protected for reading and writing and must therefore be opened. On some versions of Linux this has to be done after each reboot of the system. To make this change automatically at each reboot, see: Automatic start of SEISLOG at reboot.

Linux is normally prepared for many serial ports. To get a list of the possible serial ports on your system type:

ls /dev/ttyS*

This will give you a list of a lot of ports. Normally your PC is equipped with one or two serial ports, /dev/ttyS0 and /dev/ttyS1 respectively. If you connect your digitiser to serial port one, you have to open /dev/ttyS0 for user access.

To open serial port ttyS0:
Login as su and type:

chmod 777 /dev/ttyS0

log out from su.

STEP 3:

Setup your SEISLOG account:

If your SEISLOG account is new:

Copy a predefined login file to your home directory:

cp /home/seismo/SLG/PRO/.cshrc /home/seismo/.cshrc

source .cshrc

to set up the new environment.

STEP 4:

type:

hom

cd SLG

Compile all programs

type:

./compilePC

In the case of a 32-bit architecture, run the command:

cp /home/seismo/SLG/COM/slg2mseed /home/seismo/SLG/PRO

If you have the cross-compiler for the ARM TS-7260 installed on your system, a compiled version can be prepared by typing:

./compileARM

SEISLOG should now be ready to use. It is recommended that the user do a test of the installation by running SEISLOG with a default parameter set that do no require a physical digitiser connected to a serial port. Synthetic data will be created and the functionality of SEISLOG can be tested. See section: **3 TEST SEISLOG.**
When the test is completed, the user will normally skip to section 4.

3 TEST SEISLOG

To start SEISLOG in test mode, with synthetic data, type:

srt –d DIG_SNT –c CHA_SNT -tag

SEISLOG will print out some information lines on the screen to inform that SEISLOG has started:

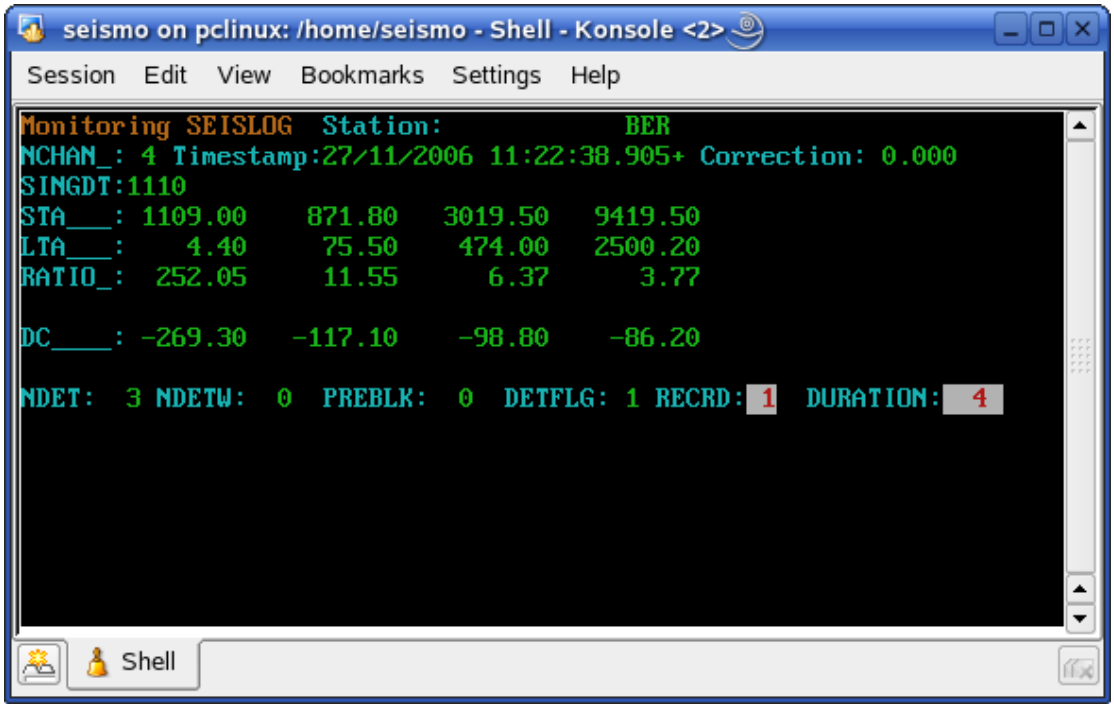
Stop current active SEISLOG before starting again.
Start all servers
SEISLOG started

Open another window on your screen and type:

mon

This will produce a screen like the one below. If you see the time and some of the other values changing every second, your installation is probably ok.
The meaning of the different values is explained in the section: Monitoring SEISLOG.

To stop the monitoring program, you press any key on the keyboard.



SEISLOG in test mode will record ringbuffers and generate events. To see where the different files are recorded, type:

paths

This command will give the following listing on your screen:

```
------------------------------------------------------
Current path setting
------------------------------------------------------
Events.............:/home/seismo/SLG/EVT
Events S-files.....:/home/seismo/SLG/EVT
Ringbuffers........:/home/seismo/SLG/RNG
Ringbuffers S-files:/home/seismo/SLG/RNG
------------------------------------------------------------------------
```

4 PARAMETER FILES

SEISLOG uses three different parameter files that have to be configured before starting data acquisition. The files are stored as text files and can be edited with any text editor. Two files, DIG_XXX and CHA_XXX, define the setup for the digitiser and the channels respectively. The 'XXX' is a three-letter code that indicates the digitiser to use. CHA_XXX also set up the size of ring buffers and the trigger parameters. Some example parameter files are stored in the SEISLOG_TOP/SLG/PAR directory as for example: DIG_SNT and CHA_SNT which can be used to test the functionality of SEISLOG with synthetic data.
Two default parameter files DIG and CHA are used if SEISLOG is started without specifying any parameter files. DIG and CHA should be set up to fit your configuration.
To test the functionality of the system you should start SEISLOG as described in the previous section with synthetic data.
Normally the digitiser is connected to one of the serial ports on the computer and the corresponding port and baud rate is specified in the parameter file DIG_XXX as explained below.
The user is responsible for removing recorded events from the hard disk or the USB pen in the case of an embedded installation.

Parameter files are located in SEISLOG_TOP/SLG/PAR:

- DIG_XXX Digitiser info
- CHA_XXX Channel info
- syspar SEISLOG paths

Below is a complete parameter setup example for the SADC24 3 channel 24 bits digitizer from SARA:

This file defines the setup of the digitiser and where it is connected.

A DIG_XXX example file is shown below:

DIG_SA2

DIG_NO	UNIT	SERIAL_PORT	BAUDRATE	IP_ADDRESS	SOCKETPORT	SAMPLERATE	TIMING	TIME-CORR
1	SA2_S	COM01	38400	-	-	0.020	2	0.000

DIG_NO Each digitizer connected must be assigned a unique number. The numbers must be assigned in an increasing order with no gaps in between.

UNIT Each type of digitizer has a predefined unit code. SA2_S is the 24 bit 3 channel digitizer from SARA. The "_S" indicates that its connected to a serial port. Supported digitisers are listed below.

EDA_S 10 EDAS-3 Geodevice

RD3_S 13	**Nanometrics RD3**
SA3_S 15	**SADC30 16 channel 16 bit**
SA2_S 16	**SADC24 3 channel 24 bit**
SA1_S 14	**SADC10 4 channel 16 bit**
GBV_S 18	**GeoSig GBV**
K2K_S 21	**Kinemetrics Altus**
ED3_S 23	**Earth Data 3 channel**
ED6_S 29	**Earth Data 6 channel**
GUR_S 30	**Guralp**
SNT_S 99	**Synthetic data for testing**

BAUDRATE	RS232 baud rate.
IP_ADDRESS	If the digitizer is connected to the Internet, an IP number must be assigned. (no support this version)
SOCKETPORT	If the digitizer is connected to the Internet, a port number must be assigned. (no support this version)
SAMPLERATE	The sample interval is given in seconds between samples.
TIMING	This parameter indicate the timing system to be used in this setup. 0 – CPU time, 2 – digitizer timing
TIME-CORR	Floating point value in seconds for time correction (+/-)

A CHA_XXX example file is shown below:

This file contains information and parameters for the network trigger and for all the channels supplied by the specified digitiser.

CHA_SA2

```
AGENCY NETWORK
BER         NS
TRIGGER PARAMETERS:
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
       20         20                20            3           200
CHANNEL PARAMETERS:
SEQ DIG CH MBUF FSIZ DAY  STAT CHA LOC F-LOW F-HIGH   STA   LTA TRIG DTRIG DSK REC
  1   1  1    5   20   1   BER BHZ  00   2.0   10.0   4.0 400.0  2.5   1.5   0   1
  2   1  2    5   20   1   BER BHN  00   2.0   10.0   0.0 400.0  2.5   1.5   0   1
  3   1  3    5   20   1   BER BHE  00   2.0   10.0   4.0 400.0  2.5   1.5   1   1
*******************************************************************************
```

1. line: AGENCY NETWORK (commentary line)
2. line: BER, agency name (max 5 char). NS network name (2 char)
3. line: TRIGGER PARAMETERS: (commentary line)
4. line: Title line for network trigger parameters. (commentary line)
5. line: Network parameter line contains the following parameters:

PRE_EVENT:	Number of seconds pre-event memory.
POST_EVENT:	Number of seconds post-event memory.
ARRAY-PROPAGATION:	Number of seconds array-propagation window.
MIN_TRIGGERS:	Number of triggers to initiate event recording.
MAX_RECORDING:	Maximum number of seconds to record.

6. line: CHANNEL PARAMETERS: (commentary line)
7. line: Title line for channel parameters. (commentary line)
8.-n. line: Channel parameter line contain the following parameters:

SEQ: Sequence number for all channels connected.
From 1 – number of channels connected.
DIG: Digitizer number. Must be one for all channels.
First digitiser no.1, second digitiser no.2, etc.
CH: Channel sequence number this digitizer.
Individual channels of each digitiser are numbered 1 to number of channels of this digitiser.
MBUF: Length of memory ringbuffer this channel in minutes.
FSIZ: Length of each disk ringbuffer file for this channel in minutes.
DAY: Number of days in ringbuffer system.
STAT: Station name, max 5 characters.
CHA: Component, always 3 characters, BHZ, BHN etc.
LOC: Location, always 2 characters, 00,01,10
F-LOW: Bandpass filter, low pass frequency for trigger.
F-HIGH: Bandpass filter, high pass frequency for trigger.
STA: Short term average in seconds.
LTA: Long term average in seconds.
TRIG: Trigger ratio (STA/LTA)
DTRIG: De-Trigger ratio (STA/LTA).
DSK: Disk ringbuffer, 1-record, 0-no recording.
REC: Event recording, 1-record, 0-no recording.

syspar

The syspar file defines the directories used by SEISLOG to store recorded data.
The paths can be set to any valid directory name. The directory will be created if it does not exist.

```
Events.............:/home/seismo/SLG/EVT
Events S-files.....:/home/seismo/SLG/EVT
Ringbuffers........:/home/seismo/SLG/RNG
Ringbuffers S-files:/home/seismo/SLG/REA
```

Event S-files and Ringbuffer S-files are stored in a year-month-day structure. Data can be recorded directly into a SEISAN data base structure. This will give direct access to the data from for example the ‘eev’ in SEISAN. It is important to remember that the name of the SEISAN database always is 5 characters. So a typical syspar for SEISAN setup could be:

```
Events.............:/home/seismo/seismo/WAV
Events S-files.....:/home/seismo/seismo/REA/BERB_
Ringbuffers........:/home/seismo/SLG/RNG
Ringbuffers S-files:/home/seismo/seismo/REA
```

Where BERB_ is the name of the SEISAN database, actually a directory.
See definition in SEISAN manual.

Description of the directories specified in the syspar file:

Directory for event waveform files: /home/seismo/SLG/EVT
Directory where the triggered waveform files are recorded

Directory for event s-files: /home/seismo/SLG/EVT
Directory for s-files. An s-file is created for each triggered event and contains trigger time for each channel as well as the duration of each trigger for each channel. The s-files can be used for SEISAN processing like earthquake location and plotting. The s-files are stored in a year-month-day directory structure.

Directory for ringbuffer files: /home/seismo/SLG/RNG
Directory for location of ringbuffers. In this directory, new files will be created – one for each ringbuffer.

Directory for ringbuffer s-files: /home/seismo/SLG/RNG
Directory where the ringbuffer s-files are located. An s-file is created for each ringbuffer file. The s-files can be used for SEISAN processing like plotting of continuous data. The s-files are stored in a year-month-day directory structure, and base names are.

5 START SEISLOG

SEISLOG can be started from the command line with different arguments or it be started automatically when the computer is started or rebooted. Automatic start at reboot is described in separate section below.
The SEISLOG server used for local and remote monitoring will be started by default when starting SEISLOG.

In the PAR directory there are example files for some of the supported digitisers. They are named DIG_SNT and CHA_SNT for the synthetic digitiser and DIG_SA3 and CHA_SA3 for the SADC30 16 channel digitiser from SARA. These are valid parameter files and can be used directly to test SEISLOG with the corresponding digitiser. The user must of course adjust the parameters to the actual configuration. (number of ringbuffers, trigger-levels etc.)

The general command to start SEISLOG is **srt** or **srt &**

srt runs SEISLOG in the foreground, which in practical terms means that it will occupy your console window session. You will not be able to for example type in new commands in the same window.
srt & runs SEISLOG in the background, which means that SEISLOG is started, but your console window is released so that the user can use it.
It should be noted that when SEISLOG is started with the arguments that gives output to the screen, it can be confusing to write new commands in the same window.

The command takes different arguments that is explained below.

The following command will start data-acquisition with the parameter files explained in the example under the section: PARAMETER FILES, assuming that the digitizer SAD24 from SARA is connected to the serial port 1.

srt –d DIG_SA2 –c CHA_SA2

As SEISLOG by default uses digitiser file DIG and channel file CHA, it is possible to make the start command even shorter by copying DIG_SA2 to DIG and CHA_SA2 to CHA in the parameter directory.
The start command will there after be:

srt

This assumes that SEISLOG has been installed under the default account seismo.

The command above can be used to test the SEISLOG right after installation, assuming default top directory, /home/seismo.
Be aware the starting SEISLOG with synthetic data, will generate triggers and record events and ringbuffers.

Type **srt –h** to see the different options that can be used when starting SEISLOG:

```
srt –h
Usage: seislog [options]
-h                  Print this information
-tag                Print time tags,
-n timeserver       Run ntpdate at 10 seconds interval.
-m timeserver       Run ntpclient at 10 seconds interval.
-s                  Do NOT start server,
-u processor        0-standard PC, 2-TS-7260 embedded
-t  SEISLOG_TOP     Enter path to SEISLOG top directory,
-d  digfile         Name digitiser file, must be specified, if not: DIG is used.
-c  chafile         Name channel file, must be specified, if not: CHA is used.
-f                  Do not send info.
```

SEISLOG options

-tag — This option will print out each time tag on the screen from the digitiser data format, or the time tag written by the driver. Default: no print.

-n timeserver — This option will start the **ntpdate** utility that updates the cpu time every 10 seconds with the timeserver entered. The timeserver entered must be valid. The default is: no cpu update. This option can only be used when your computer have a permanent Internet connection. Only available for PC desktop version. Default: no cpu update.

-m timeserver — This option will start the **ntpclient** utility that updates the cpu time every 10 seconds with the timeserver entered. The timeserver entered must be valid. The default is: no cpu update. This option can only be used when your computer have a permanent Internet connection. Only available for embedded version. Default: no cpu update.

-s — This option will turn off the server function. By default it will always start. With the server running, the monitoring functions can be used locally or remotely. Default: start server.

-t — This option is used to give an alternative top directory for SEISLOG. Default: SEISLOG_TOP

-d digfile — With this option the user may use alternative digitiser files than the default which is DIG. This can be useful for

example to test the functionality of SEISLOG without connecting a real digitiser, by specifying the DIG_SNT file. This “digitiser” will supply synthetic data to SEISLOG and also generate synthetic events. Default: DIG

-c chafile — With this option the user may use alternative channel files other than the default which is CHA. This can be useful for test purposes as described above. The alternative channel file must however, fit the digitiser specified in the option above. The CHA_SNT can be specified together with the DIG_SNT to run SEISLOG with synthetic data. Default: CHA.

-f — By default, SEISLOG sends a “state-of-health” message to a server in Bergen to monitor the functionality of the system. Systems that are connected to Internet through lines where the user must pay per call or per byte transferred, should turn this option off by adding a –f in the start line. For systems with normal free traffic, just leave it on.

Examples:

srt
Start SEISLOG with default parameter files.
Top directory: /home/seismo
No printing of time tags
The cpu time is not updated
The server is running
SEISLOG will use the current DIG and CHA parameter files

srt –tag –n ntp.uio.no –t /home/mydir
Time tags will be printed
cpu time will be updated by connecting to the timeserver ntp.uio.no, the computer must have access to Internet.
SEISLOG top directory is /home/mydir

srt –tag –n ntp.uio.no –d DIG_SNT –c CHA_SNT
Time tags will be printed
cpu time will be updated by connecting to the timeserver ntp.uio.no, computer must have access to Internet
SEISLOG will use parameter files DIG_SNT and CHA_SNT to test functionality with synthetic data.

Stop SEISLOG

The command to stop SEISLOG and the SEISLOG server is: **stp**
Type command: **stp**

6 TIMING

SEISLOG can use two different sources to time stamp the data.

- Data time stamped by digitiser with integrated GPS. Time from data stream is always used and is assumed to be synchronized.

- CPU time.

    1. Free running

    2. Synchronized by ntp server. This option depends on a permanent Internet connection. The network time protocol deamon (NTPD) must be installed if not installed during installation of Linux, and the xntpd must be configured to start at boot time. This is done by activating this service. In SuSE Linux this can be done from YAST/SystemServices/Enable/Disable. The ntp server must be configured to connect to 2-3 time servers to improve the time setting. The ntp-deamon switches between the time servers and selects one as a "favourite" when reception is good. The time servers are added in the /etc/ntp.conf file with a text editor. Different time servers closer to your site can be found at:
    http://ntp.isc.org/bin/view/Servers/StratumOneTimeServers
    For example:
    server swisstime.ethz.ch
    server ntp1.nl.net
    server ntp1.mmo.netnod.se

    3. For systems where ntpd is not available, the program **ntpdate** may be used if present in the distribution.
    ntpdate sets the date and time by polling the Network Time Protocol (NTP) server given as the server argument to determine the correct time. ntpdate is normally part of the Linux distribution.

The three SEISLOG utilities **snc**, **adj** and **log** are used to set the CPU time and to update the shared variable **correction** which is the current offset from correct time.
The program **snc** is started from SEISLOG if the option –n is set. The program takes one argument, which is the name of a valid time server, and calls the **adj** program. The adjust program runs the **ntpdate** program every 10 seconds with the given time server as an argument. The output from adjust is piped into the program **log** which in turn updates the shared variable **correction** to indicate the current time offset.
If enabled, the xntpd service must be disabled to run the ntpdate program.
The ntpdate is a privileged utility. To be able to run this utility from normal user account, change the attributes of the ntpdate utility like this as superuser:

chmod u+s /usr/sbin/ntpdate
chmod g+s /usr/sbin/ntpdate

7 AUTOMATIC START OF SEISLOG AT BOOT TIME

Before you configure SEISLOG to start automatically at startup or reboot of the computer, you should check that you have

- valid parameter files
- a network connection that is working if needed
- tested the **srt** start command with the options you plan to use
- the –t topdirectory MUST be included when starting SEISLOG automatically at boot.

Below are outlined three different ways to start SEISLOG automatically after a reboot. They all work under Suse Linux, but different versions of Linux may not support all three, but one of the three should normally work.
The first method described is by far the easiest to implement.

Simplest way to start SEISLOG automatically at boot time.

Edit the /etc/rc.d/boot.local file. You must be superuser to do this.
(For Linux Ubuntu the file is: /etc/rc.local)
Add the complete start command for your configuration, for example:

/home/seismo/SLG/PRO/srt –d DIG_XXX –c CHA_XXX –t /home/seismo &

The ampersand '**&**' at the end is important. The ampersand puts SEISLOG in the background and continues the boot procedure.

The description below works for SuSE Linux, but should be the same or similar for other distributions.

8 SEISLOG SERVER AND SOCKETS

SEISLOG is designed to work without a graphics user interface. However, some monitoring functions are available through a server program that can communicate with other programs through sockets. This communication can take place on the machine where SEISLOG is installed or through the Internet. This of course, assumes that both the SEISLOG machine and the user machine are both connected to the Internet.
The server program is part of the distribution. SEISLOG can work without the server, but the user will then not have access to the monitoring functions.
Most of the server functions are multi-client, i.e. the functions can be started by several clients at the same time.
By default, the server is started when SEISLOG is started.

The SEISLOG server responds to different requests from clients. The most important server function is to transmit packets of data in a well defined format to other programs or clients. The SEISLOG server emulates the LISS server (USGS) and transmits data packets in MiniSeed format to clients that request data. This format is recognized by Earthworm and SeisComp, and data from SEISLOG can therefore be input to these systems. It is important to know that the protocol is one way and re-transmission is not possible. However, on stable Internet connections, the data loss is minimal. The clients may also run on the same system as the server, which means very stable communication.
Data from the server may also be requested for plotting by the utility LISSPLOT that is described in section 9. LISSPLOT can connect locally or remotely to any station in the GSN network or to any SEISLOG Linux system.

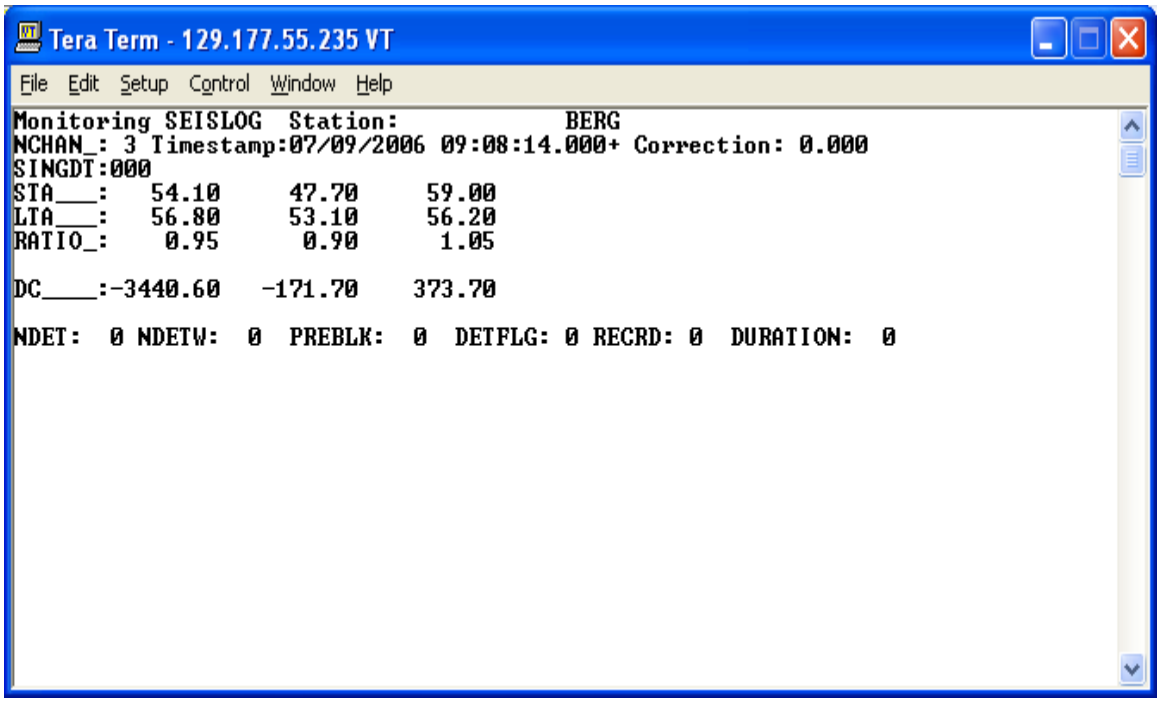
9 MONITORING SEISLOG

Checking for events.

Checking for recorded events can be done by looking for event files in the events directory, or by using the SEISAN eev program (SEISAN must be installed).

Console commands

mon - Monitoring SEISLOG activity (requires that server is running).



The utility program mon can be used to monitor the current status of SEISLOG. Values shown are:

Station:	Name of this station
NCHAN:	Number of channels
Timestamp:	Timestamp current second buffer. The '+' or '-' sign right after the timestamp, indicates weather the timestamp is synchronized to GPS or not.
Correction:	Time correction added. Set in the DIG_XXX parameter file.
SINGDT:	Single channel trigger flags
STA:	Short term average each channel
LTA:	Long term average each channel
RATIO:	Current STA/LTA ratio each channel
DC:	DC level each channel (counts)
NDET:	Current number of channels triggering
NDETW:	Current number of channels triggering within array propagation window.
PREBLK:	Counting buffers from first trigger

DETFLG: Network trigger flag. 0->no trigger, 1->trigger mode
RECRD: Recording flag 0->no recording, 1->recording mode
DURATION: Current duration in number of buffers (seconds) of an active trigger.

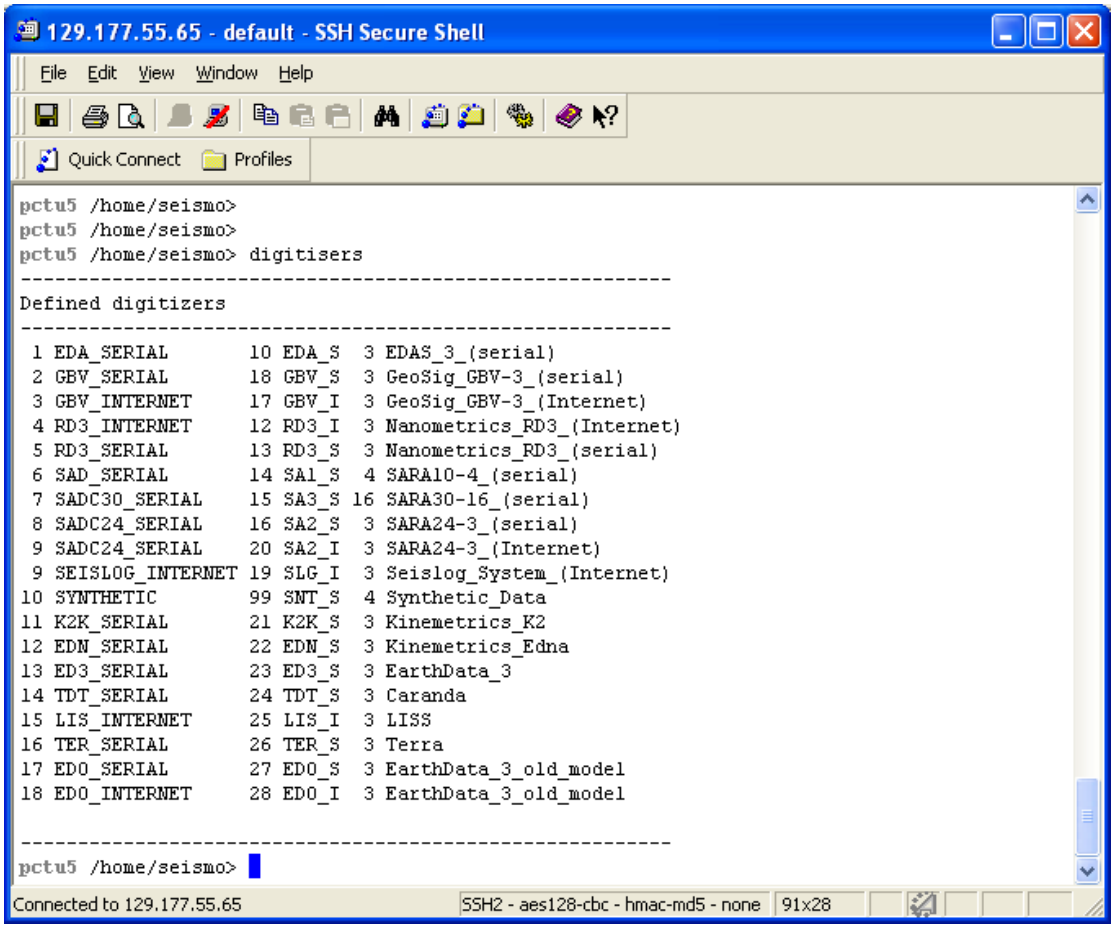
mon can also be started from a remote machine by giving the ip number of the SEISLOG computer as an argument: mon ip-number.

mon is stopped by pressing any character on the keyboard.

Mon requires that requests on TCP/IP port 10222 is allowed. See section System Setup.

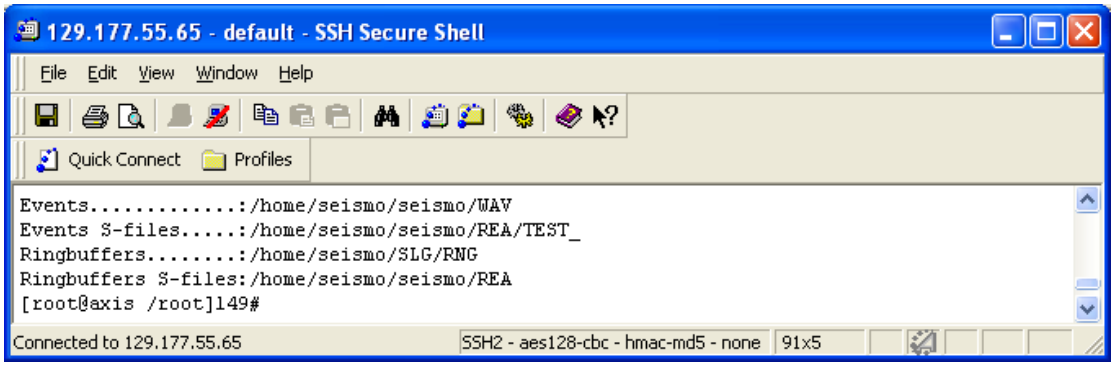
digitisers - List digitisers defined for SEISLOG.

The utillity digitisers lists all digitisers that are defined for SEISLOG. Some are under development. The list of supported digitisers will be updated for future releases.



paths - List current paths defined for SEISLOG.

The utility paths lists the current syspar.txt file.



onn – Turn on trigger flag manually.

The utility onn turns on the trigger flag manually

off – Turn off trigger flag manually.

The utility off turns off the trigger flag manually.

lissplot - Plot data in real time (requires that server is running).

lissplot is a java program that reads miniseed formatted data from a socket and plot them on the screen in a helicorder fashion.
Data may be in compressed form (Steim1 or Steim2) or in standard 4-byte integer form. lissplot can read and plot data from all the Global Seismograph Network stations (GSN) and from SEISLOG stations where the LISS compatible server is implemented. See documentation below.
The SEISLOG server supplies the lissplot program with data in miniseed format. lissplot can be started in different windows to show different components, filters or scales.
lissplot requires that requests on TCP/IP port 4000 is allowed. See:System setup.

10 LISSPLOT

The program can be used to display seismic signals from seismic stations that are included in the Global Seismograph Network (GSN) and supported by the Live Internet Seismic Server (LISS), USGS, Albuquerque Seismological Laboratory.
Data from other seismic stations that conforms to the same data exchange specifications can also be displayed.
LISSPLOT is written in Java and can be installed under Linux, SUN or Windows operating system.
The program is distributed as a tar file: lissplotddmmyy.tar where ddmmyy is the date of the distribution. To install lissplot, untar the distribution file in your preferred directory:

For installation on Linux and SUN:

Download the lissplotddmmyy.tar to your home directory and unpack the file with the command:

tar –xvf lissplotddmmyy.tar

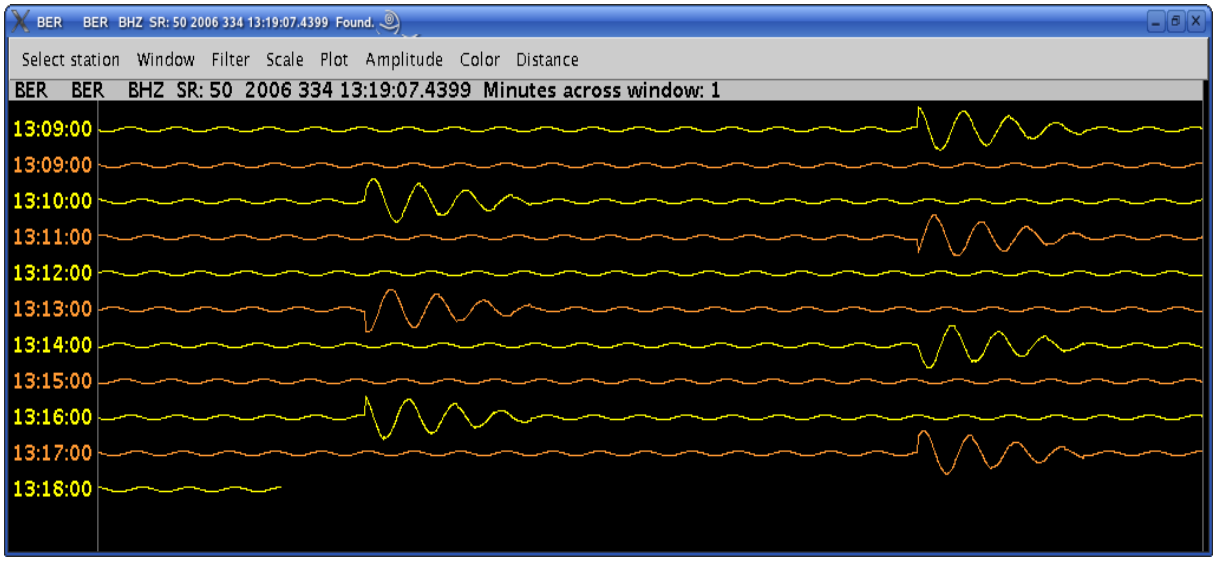
For installation on Windows:

Download the lissplotddmmyy.tar to c: and use WinZip or similar to unpack the distribution.

Test functionality of LISSPLOT

For this test to work, Java must be installed and also defined in your path. An easy way to test this is to write **java** in your window. If you get the message *Command don't exist* or similar, java is at least not in your path, and must be installed. If you get some kind of explanation on the java command, you can go on with the test.

First start SEISLOG with default parameter files (synthetic data) as described in section 3.
Then write the command **lissplot_test** on your screen and you should get the following picture on the screen.



For both installations, the program will be placed under the main catalog LISSPLOT and create the complete project structure for JBuilder including source and classes. Under the LISSPLOT directory you will find the SETUP directory that contains the following files:

coefficients0 program to make filter coefficients (Linux)
coefficients1 program to make filter coefficients (Windows)
coefficients2 program to make filter coefficients (SunOS)

mystations.def — Text file containing the IP address and station name for the component you want to plot. Lissplot can plot data from the SEISLOG running on the same PC, or from a SEISLOG that is running on a remote PC. You may add or remove components with a standard text editor. The user will always find the line: **127.0.0.1 BER** in the file. If the user is testing SEISLOG with the default parameter files, he should select this station which corresponds to the default parameters.

Example mystations.def file:

127.0.0.1	BER
127.0.0.1	CNI
127.0.0.1	BRU
129.177.55.xx	BER
200.108.177.xx	PTP
129.177.55.xx	BGO

liss.def — Text file containing the Internet address and a the name of official stations in the Global Seismograph Network (GSN). You may add and remove stations with a standard text editor.

How to use LISSPLOT:

LISSPLOT can be run interactively or from a script:

Interactivly

Before starting the program it is recommended to make an alias or .bat file depending on your operating system, to make the start of the program more easy.

For Linux and SUN:

The predefined aliases are set up in the /home/seismo/SLG/COM/.SEISLOG file. The user should modify this file to fit the installation if not default. The default alias for LISSPLOT is:

alias lissplot 'java –classpath "/home/seismo/LISSPLOT/classes" lissplot.lissplot /home/seismo'

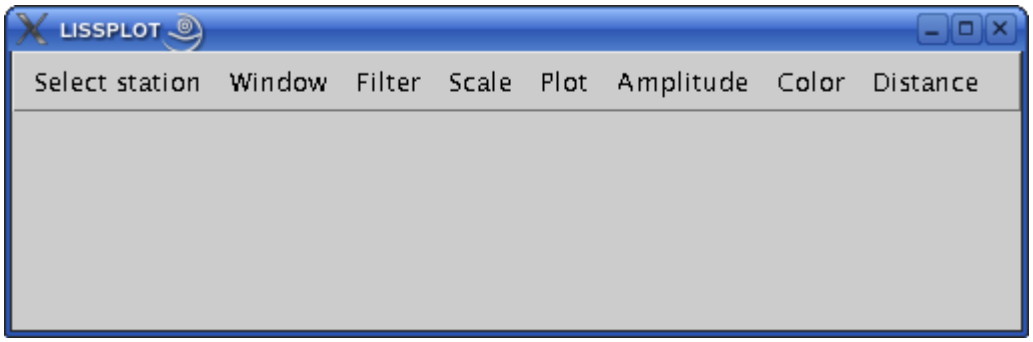
An alias like the one above will work as long as 'java' is in the path.

alias lissplot '/home/seismo/Borland/JBuilder2005/jdk1.4/bin/java –classpath "/home/seismo/LISSPLOT/classes" lissplot.lissplot /home/seismo'

The alias above assumes that the java development system is installed. This will normally not be the case for the user.

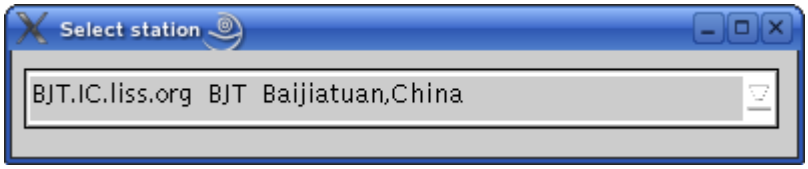
If you now write **lissplot** at the prompt the program will start and the user has to select parameters from the menus on top of the window.

Main menu:

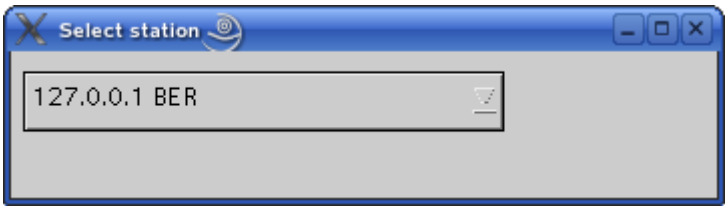


Select station (mandatory) Pressing this item gives you three options:

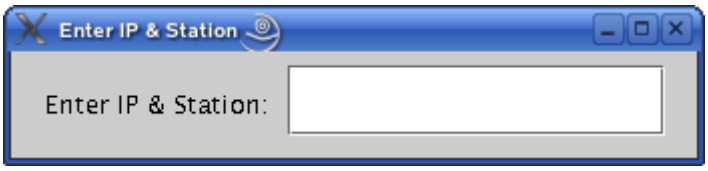
Choose from LISS Select among registered official LISS stations in the GSN global network. Stations defined in liss.def.



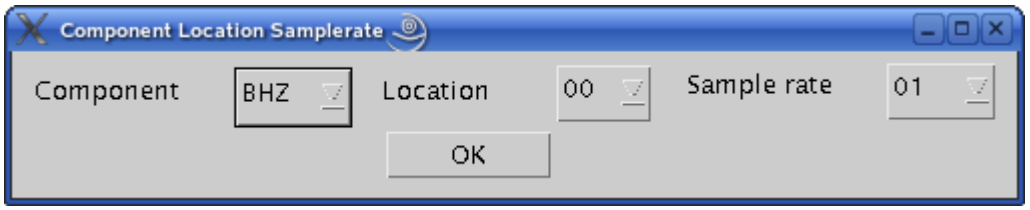
Choose from Network Select among SEISLOG stations defined in mystations.def.



Enter IP and station Enter the ip number of a station and the station name with a space in between and press 'enter'. Station name is the name under 'STAT' in the CHA_XXX file.



After selecting one of the options, a new menu will appear:



Component Location Samplerate

Component Location Samplerate OK (all mandatory) In this menu select 'component', 'location' and 'samplerate'. Press 'OK' when finished

Window (optional) Select size of window for plotting.

Full screen Full screen (1/1 height x 1/1 width)

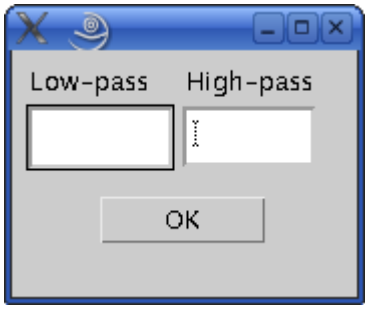
1/2 screen Half screen (1/2 height x 1/1 width)

1/3 screen Third screen (1/3 height x 1/1 width)

1/4 screen Quarter screen (1/2 height x 1/2 width)

Filter (optional) Enter the low-pass and high-pass frequency or no filter.

Filter Selecting this option will open a new menu where you can enter the low-pass and high pass frequency.



Low-pass High-Pass OK	Enter the frequencies end press 'ok'
No filter	No filtering.
Scale	(optional) Select the total time over the screen. Default is selected based on samplerate.
Plot	(mandatory) Start plotting or stop plotting
Amplitude	(optional) Multiplication factor amplitude.
Color	(optional) Color scheme. Default: yellow on black background.
Distance	(optional) Distance between traces in pixels.

NB ! Start of plotting must be done after all mandatory options have been selected. The only option that can be selected after plotting is started is Amplitude.

For Windows:

Create the file lissplot.bat in your path containing the line:

java –classpath "c:\LISSPLOT\classes" lissplot.lissplot c: %1

The classpath is the complete path of the program. The argument 'c': is the top directory or location where the lissplot distribution has been installed.
So for example if you have installed lissplot under c:\mydir\plotting the corresponding bat file would be:

java –classpath "c:\mydir\plotting\LISSPLOT\classes" lissplot.lissplot c:\mydir\plotting %1

You can then start the program by typing**: lissplot**

Parameters are selected as described above for Linux.

The user may start several copies of the program, for example from different DOS windows or from different Linux windows. By selecting for example the

windows size “Quarter screen”, the user may monitor 4 different stations on the same screen, or one station with 4 different filters etc. For PCs that support 2 screens, up to 8 stations can be monitored simultaneously.

Run lissplot with fixed parameters from a file.

Create a file containing one line with all parameters, for example:

KONO.iu.liss.org KONO BHZ 40 10 2.0 8.0 1 0 0

KONO.iu.liss.org	name of station (Internet address)
KONO	station name
BHZ	component
40	samplerate
10	location
2.0	filter lowpass 2.0 Hz
8.0	filter high pass 8.0 Hz
1	display size 1 – the whole screen 2 – half the screen 3 – one third of the screen 4 – quarter of the screen
0	x – position of display (pixels)
0	y – position of display (pixels)

Store it as KONO in the directory you prefer.

For both Linux and Windows, lissplot can then be started with the command:

lissplot [path]KONO where path is where you stored it above.

A practical way to use lissplot is to set up the alias for your installation and then create parameter files for the stations and components you normally want to plot. They can then be easily monitored one by one or several together.

To start the program at boot time (Windows):

Make new .bat file(s) and store them in the Windows startup folder.

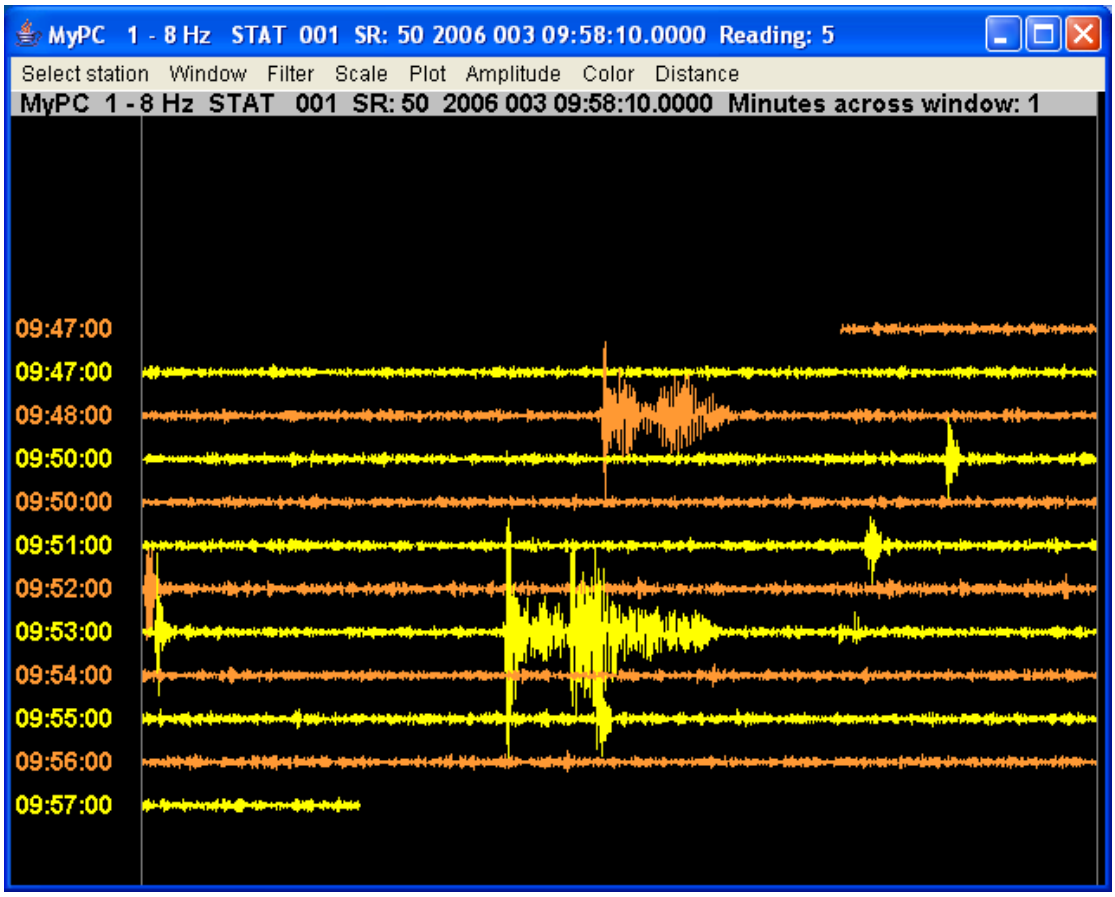
For example:

d1.bat

java –classpath “c:\LISSPLOT\classes” lissplot.lissplot c: LVC

d2.bat

java –classpath “c:\LISSPLOT\classes” lissplot.lissplot c: KONO



11 DATA RECORDING AND DATA INSPECTION

When SEISLOG has been set up and is running, data will be recorded in different directories depending on the setup. The ring-buffer files will continuously be over written, however the event files will gradually fill up the disk. It is the user's responsibility to make sure that there is space left on the disk for new recordings.

Continuous data.

For each disk ring-buffer defined in the parameter file, there will be a directory where all the ring-buffer files are stored. The directory will be created below SEISLOG_TOP/SLG/RNG and is based on the station name and component. If the station name is ABCD and the component is BHZ, the directory name will be ABCD_BHZ. Initially all files are named R00000 to R0xxxx where xxxx is the total number of files. Based on the parameters, the sizes of the files are identical. The files are written sequentially, 0,1,2 etc. When a file has been completed with data, the file is reformatted to MiniSEED format, compressed and renamed.

R00000-2005-09-14-0629-14-000300-0-ABCD_-BHZ
R00001-2005-09-14-0634-10-000300-0-ABCD_-BHZ

The fields from left to right are:
Ring buffer file number
Year
Month
Hour-Minute
Second
Duration of file in seconds: If the file is complete, this duration will match the setup for the ring buffer files.
Status of timing (not implemented, set to 0)
Station
Component

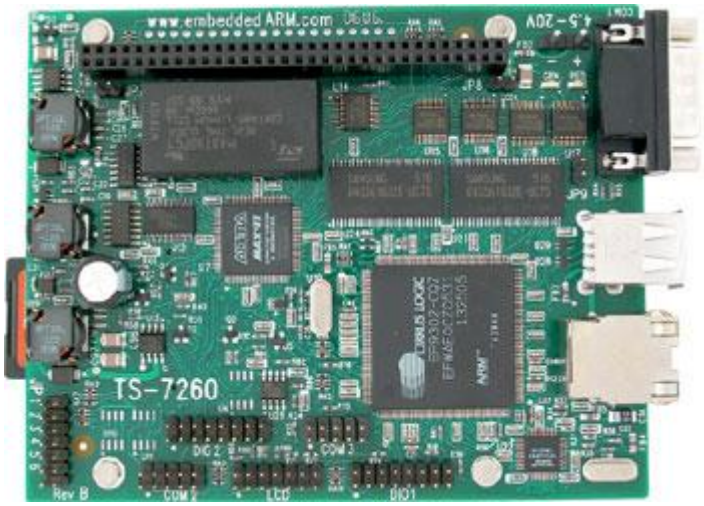
The intention with the informative file name is to have a complete log of the data available in the ring buffer system as well as the status of the ring buffer system. This information is not available any other place in SEISLOG. By just listing the file names, it is thus straightforward to find out if data in a given time interval is available. The corresponding ring buffer file can then be copied to an analysis program for further processing or it can be plotted directly using the SEISAN program mulplt. There is also a SEISAN program for extracting data in a given time interval, from some or all of the ring buffer files, if the ring buffer files are registered into a SEISAN data base.

12 SEISLOG on TS-7260

The TS-7260 is a single-board-computer (SBC) based on the ARM9 200 Mhz cpu.
The ARM architecture is different from the i86 standard:
“The ARM architecture (originally the Acorn RISC Machine) is a 32-bit RISC processor that is widely used in a number of embedded designs. Due to their power saving features, ARM CPUs are dominant in the mobile electronics market, where low power consumption is a critical design goal.”
A single-board-computer has normally the advantage of small size, low power consumption and low price. Disadvantages are normally no hard disk, less memory and less user friendliness.
The TS-7260 do have an embedded Linux operating system installed from the manufacturer if ordered.
The ARM architecture does not execute the binary code from the i86 platform and software must be re-compiled with a cross-compiler for this specific system.
A pre-compiled distribution of SEISLOG can be downloaded from our website.
As will be shown below, SEISLOG can also be re-compiled on the TS-6260 itself, but at the cost of speed.

Specifications for the TS-7260, Technologic Systems

- Voltage: 12VDC or 5VDC
- Power consumption: 300mA
- CPU: ARM9 200MHz
- Memory: 32Mb on-board NAND flash (Linux)
- Memory: 64Mb SDRAM
- Board size: 10 x 12 cm
- 2 USB 2.0 ports
- 10/100 Ethernet port
- 3 serial ports
- SD memory card



Cross-compiler for the TS-7260

The cross-compiler for the TS-7260 can be downloaded from the manufacturers web-site (www.embeddedarm.com).

The SEISLOG Linux distribution contains all the binaries that are needed to run SEISLOG on the TS-7260.

Preparation of the system

- Allow writing to internal flash memory

  Check the jumper setting on the jumpers located in the lower left corner to the left of the connector COM2 on the picture above.
  Only leave J2 & J3 connected

- Serial port 1 as cosole

  Connect to serial port 1 with terminal emulator from your laptop or desktop PC. See TS-7260 documentation for details

- Set up Ethernet configuration

  Modify the following files to set up your network configuration

  /etc/sysconfig/ifcfg-eth0
  /etc/sysconfig/network_cfg
  /etc/resolv.conf

- Configure serial port COM2 for connection of digitiser:

  cd /etc
  vi inittab

  comment out line:

  #lgn1:12345:respawn:/sbin/getty –L 115200 ttyAM1

  Save the file. Remember to reboot before connecting digitiser.

- Start secure shell:

  Generate host key:

  cd /etc/dropbear
  dropbearkey –t rsa –f /etc/dropbear/dropbear_rsa_host_key

  check that the key has been generated:

ls
cd /etc/rc.d
edit rcS.sysinit
uncomment line at end of file:
dropbear -i

Test secure shell from other computer to check that it works before removing telnet.

cd /etc/rc.d/rc3.d
rename S30telnetd to NOS30telnetd
now this script will not be executed at next reboot

- Check USB and SD scripts

  check if the two scripts sd and usb is present in the /root catalog.
  If not, add them with the editor and make them executable

  sd:
  mount –t ext2 /dev/sdcard0/disc0/part1 /mnt
  usb:
  loadUSBModules.sh
  sleep 3
  mount –t ext2 /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/cf

  if errors with the mount command try:
  mount –t ext2 /dev/scsi/host0/bus0/target0/lun0/disc /mnt/cf

  Make both scripts executable:

  type:
  chmod +x sd
  chmod +x usb

- Prepare startup script /etc/rcS.sysinit

  cd /etc/rc.d

  Edit rcS.sysinit and add the lines at the end that are marked in **bold** letters.
  Do not remove the comment mark # before your configuration has been tested properly.

```
#!/bin/sh
#
# /et/rc.d/rcS.sysinit
#       Technologic Systems         date:  8.03.04
# This file is used by inittab and init. Place here all the stuff you want
```

```
# to be done before any other system startup scripts are executed
# that will affect the system as a whole or are needed by the system

# Ingore CTRL+C
trap ":" INT QUIT TSTP

# Set our Path
PATH='/bin:/sbin:/usr/bin:/usr/sbin'
export PATH

echo
echo  "http://www.embeddedarm.com"
echo  "email:info@embeddedarm.com"
echo

#start devfsd
/sbin/devfsd /dev

echo "now mounting /proc"
/bin/mount -n -t proc proc /proc

echo "mounting local filesystems (in fstab)"
(insmod fat; insmod vfat; mount -a -rw -t nonfs,smb,smbfs) >/dev/null 2>&1 &


#attempt to load sdcard and TS-UART modules
insmod sdcard.o >/dev/null 2>&1
insmod tsuart0.o >/dev/null 2>&1
insmod tsuart7260.o >/dev/null 2>&1


#start system logger stuff
if  test -x /sbin/dmesg  ; then
        if  test -f /etc/consolelog.level  ; then
                . /etc/consolelog.level
        else
                LOGLEVEL=1
        fi
        /sbin/dmesg -n $LOGLEVEL
fi

##### syslog and klogd are started invoked by executing an rc directory

#set system clock here
if [ -f /proc/driver/rtc ]; then
        echo "setting system clock"
        /sbin/hwclock -s
        date
fi
```

```
if [ -f /www/apache/logs/httpd.pid ]; then
        rm -f /www/apache/logs/httpd.pid
fi
#dropbear -i
echo "Mount USB-pen or SD-card"
sleep 2
# mount usb
#/root/usb
# mount sd
#/root/sd
sleep 2
# start seislog from usb
#chroot /mnt /cfstart_seislog
# start seislog from sd
#chroot /mnt/ /start_seislog
```

A distribution of SEISLOG Linux for the ARM processor TS-7260 is available for download from our web-site:

ftp://ftp.geo.uib.no/pub/seismo/SOFTWARE/SEISLOG/LINUX/

This is a pre-compiled distribution that can run from either a USB-pen or a SD-card. The USB or SD must have a running Linux installed with a proper setup for running SEISLOG. Image files for USB and SD are available on request.

Image files for USB and SD:

arm.V.vv.USB and arm.V.vv.SD
where V.vv is version number

Install on USB or SD from scratch:

On standard Linux PC:

Log in as super user

Insert USB-pen or SD-card

Run dmesg to find device name of usb-pen or SD-card, f.ex. sdj.
Be aware the the name of the USB and SD device can vary each time it is inserted.

dmesg

Format the USB-pen or SD-card

/sbin/mkfs.ext2 /dev/sdj1

Copy image to USB-pen or SD-card

dd if=arm.3.10.USB of=/dev/sdj¨(USB pen)
dd if=arm.3.10.SD of=/dev/sdj (SD card)

(this will take some time)
Insert the USB-pen or SD-card in the TS7260
reboot
type:

./usb (for USB-pen)
./sd (for SD-card)

type:

chroot /mnt/cf (for usb-pen)
chroot /mnt (for sd-card)

mount –t proc none /proc

su – seismo

The image files have a SEISLOG version installed. If you want to install the latest version, remove the installed version before installing a new:

type:

cd /home/seismo
rm –rf SLG

The new distribution should be downloaded to the /home/seismo catalog and unpacked with the command:

tar –xzvf SEISLOGddmmmyy_ARM.tar.gz

to set up the SEISLOG environment:

cp /home/seismo/SLG/PRO/.cshrc /home/seismo

source .cshrc

If you do modifications in the SEISLOG source code, programs must be re-compiled:

type:

cd /home/seismo/SLG
./compilePC

Start and stop of SEISLOG

Manual start of SEISLOG

When you log in to the system, the current catalog /root will contain a script SD or USB that will mount the SD or USB.
The script can be executed manually after login or automatically by entering the following command lines at the end of the file /etc/rc.d/rcS.sysinit:

sleep 2
/root/sd or /root/usb

The user can now decide to start Seislog or Seislog+ Seiscomp by doing a:
chroot /mnt (for SD) or chroot /mnt/cf
./start_seislog or ./start_all

This can also be done automatically from the /etc/rc.d/rsS.sysinit by adding the following command:
chroot /mnt /start_seislog

13 System Setup

Several server functions are used in SEISLOG. These servers can communicate with clients that run on your local system as well as from remote systems over the Internet.

TCP traffic must be allowed on the following TCP ports:

4000, 4001, 4002, 4003, 10101, 10102, 10222

You must be superuser to do this.

14 PRINCIPLE OF OPERATION

SEISLOG for Linux has a different design than the QNX version. This version of SEISLOG is designed around the concepts of threads and sockets.

Definition of threads:

Threads can be thought of as *lightweight processes*, offering many of the advantages of processes without the communication requirements that separate processes require. Threads provide a means to divide the main flow of control into multiple, concurrently executing flows of control.

The threads used in SEISLOG complies with the POSIX standard.

Definition of sockets:

A method for communication between a client program and a server program in a network. A socket is defined as "the endpoint in a connection." Sockets are created and used with a set of programming requests or "function calls" sometimes called the sockets application programming interface (API). The most common sockets API is the Berkeley UNIX C language interface for sockets. Sockets can also be used for communication between processes within the same computer.

The application is split into several threads that are responsible for different tasks. The sockets are used for internal communication between threads within the application and for communication with external local or remote user utillities.

SEISLOG contains several server socket threads that are installed during start-up. The servers are connection points for different client utility programs that can be used locally within the same computer or remotely from another computer connected in a network. See details under Utilities.

Main data flow:

1. A digitizer driver-thread reads data from the digitiser, takes care of timing and writes one second buffers into a global 10 second memory ringbuffer. It sends out a ready signal after each second.

2. A memory-ringbuffer thread is started for each channel or component defined in the parameter file. These threads are waiting for the data-ready signal from the driver-thread. When it receives the signal, data from the corresponding channel are written into a channel memory ringbuffer, which size is defined in the parameter file. Each memory ringbuffer thread sends out a signal when one second of data has been written to memory.

3. A disk ringbuffer thread is started for each channel or component defined in the parameter file. These threads are waiting for the data-ready signal from the driver-thread. When it receives the signal, data from the corresponding channel are written to hard disk ringbuffer system.

4. A detection thread is started for each channel or component defined in the parameter file. These threads are waiting for the data-ready signal from the memory-ringbuffer threads. Each thread runs a filter specified in the parameter file and a detection algorithm based on STA/LTA. Detections are marked in a global detection-table.

5. A network detection thread is started. This thread examines the detection-table for triggers that together can be defined as a network trigger according to the parameters. The thread saves the start point in the memory-ringbuffer of the trigger and updates a queue of events to be written to disk.

6. A catalog event thread is started. This thread examines the event queue every second for new events to be written to disk. When there is a new event, the thread extracts the data from the memory ringbuffer and writes the complete event to hard disk.

7. Several utility socket threads are started and remains in a wait-state until there are requests from local or remote utility programs

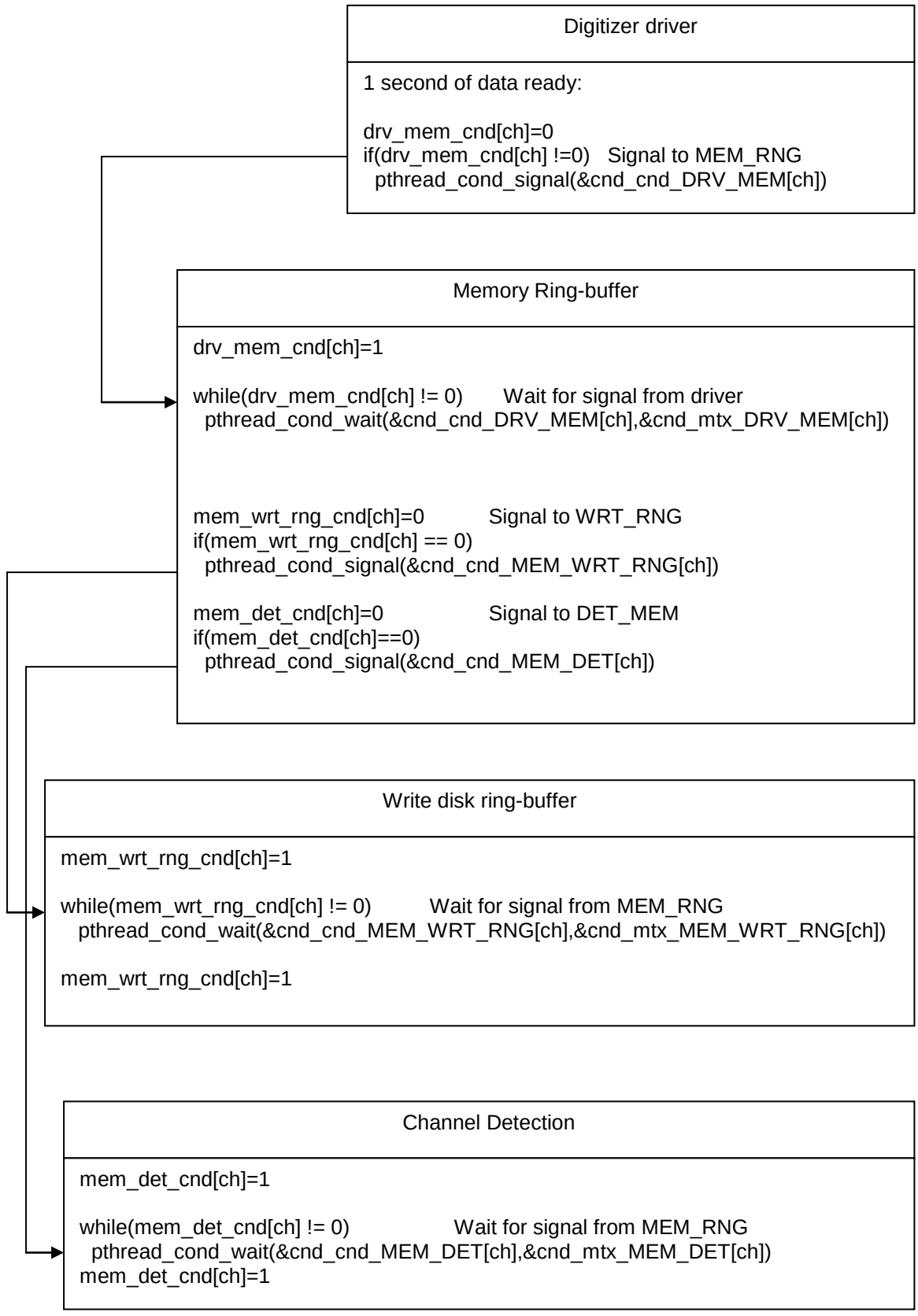
Event-files and ringbuffer-files are all accessible directly from the command prompt.

Data format.

Ringbuffer files and triggered events are recorded in MiniSEED format, and can be plotted with MULPLT in SEISAN directly.

The directory CAL is necessary only in cases where the digitizer transmits calibration data at requested or programmed intervals. The data containing calibration data will be extracted from the data stream and stored in files in this directory.

15 SYNCHRONIZATION



16 REFERENCES

Standard for the Exchange of Eartquake Data. Reference manual. SEED format version 2.3. February 1993.

Utheim, T. and J. Havskov, 1999. The Seislog data acquisition system, Version 8.44. Guide to installation, maintenance and daily operation of the system.

Havskov, J and L. Oettemøller, 2001. SEISAN: The Earthquake analysis software. Version 7.2.

Natvik, Ø. and J. Havskov, 2004. Seislog for Windows. A seismic data acquisition system for Windows9x/NT/2000/XP. Version 1.2.7.

Canabrava, R. IASTE Trainee. Mini-SEED files. 2004.

Trabant, C. IRIS Data Management Center. Libmseed library. (GNU Library General Public License).

Vasquez, C., U. 2006. Seislog – Linux TS-7260.

Mariotti, M. and T. Utheim, 2006. MiniSeed for LISS and data compression using Steim1 and Steim2.

Utheim, T. and U. Vasquez, 2006. SEISLOG – Linux, Embedded systems.

17 ACKNOWLEDGEMENTS

Department of geophysics, Seismological Service, University of Chile has paid for several stays in Chile to work on software development on SEISLOG Linux, both for desktop systems and for the embedded version.