The SEISLOG Data Acquisition System

PC version

Guide to installation, maintenance and daily operation of the system.

by

Terje Utheim and Jens Havskov

Version 8.44
Last update May 2002

University of Bergen
Institute of Solid Earth Physics
Seismological Observatory

Allegaten 41, N-5020 Bergen Norway

Fax (047)55589669 Tel. (047)55583408

e-mail: terje.utheim@ifjf.uib.no
e-mail: jens.havskov@ifjf.uib.no

Home-page: http://www.ifjf.uib.no

Institutt for den faste jords fysikk
Institute of Solid Earth Physics

1. INTRODUCTION

1.1 Hardware and software requirements

SEISLOG is a general-purpose seismic data acquisition system designed for recording of earthquakes. The system can record 1-64 channels with 12-24 bits resolution and is based entirely on off the shelf hardware. Two different analogue to digital converter boards for installation inside the PC are supported (Metrabyte 1202, 16 channels and Metrabyte 1802, 64 channels, two Metrabyte 1202 boards may be installed to get 32 channels). Other ADC boards can easily be implemented. External digitizers from different companies are also supported, Nanometrics, Earth Data, GeoSys and TerraTechnology. A typical system consists of a standard PC with the real time operating system, QNX4, a 200-10 Gb disk, minimum 16-32Mb of memory and some means of communication with another computer for data analysis.

The PC based SEISLOG is implemented under the operating system QNX4 manufactured by Quantum Systems, Canada. It is a multi-user and multi-tasking real time operating system, which makes it ideal for this type of application. MS-DOS or Windows may co-exist with QNX on a separate disk partition.
Through the years, the SEISLOG software has been modified and upgraded many times, and the performance of the software has improved a lot since the early versions. New options have been added to give the user a better tool to record the wanted seismic events. Main modifications for version 8.50 see 1.6 Comments this release.
All programs are written in C . This makes it easier to transport the software to other hardware platforms in the future.

A typical SEISLOG PC based configuration, version 8.50, will include the following items, see figure 1.1:

- 486 or Pentium CPU with the multi-tasking, multi-user, real-time operating system QNX4.

- Serial lines for communication with timing-system, digitizers, remote terminals and/or modem.

- Hard disk storage for the operating system, SEISLOG software, data-buffers and recorded events.

- Digitizer as separate unit or integrated as a board in the computer system.

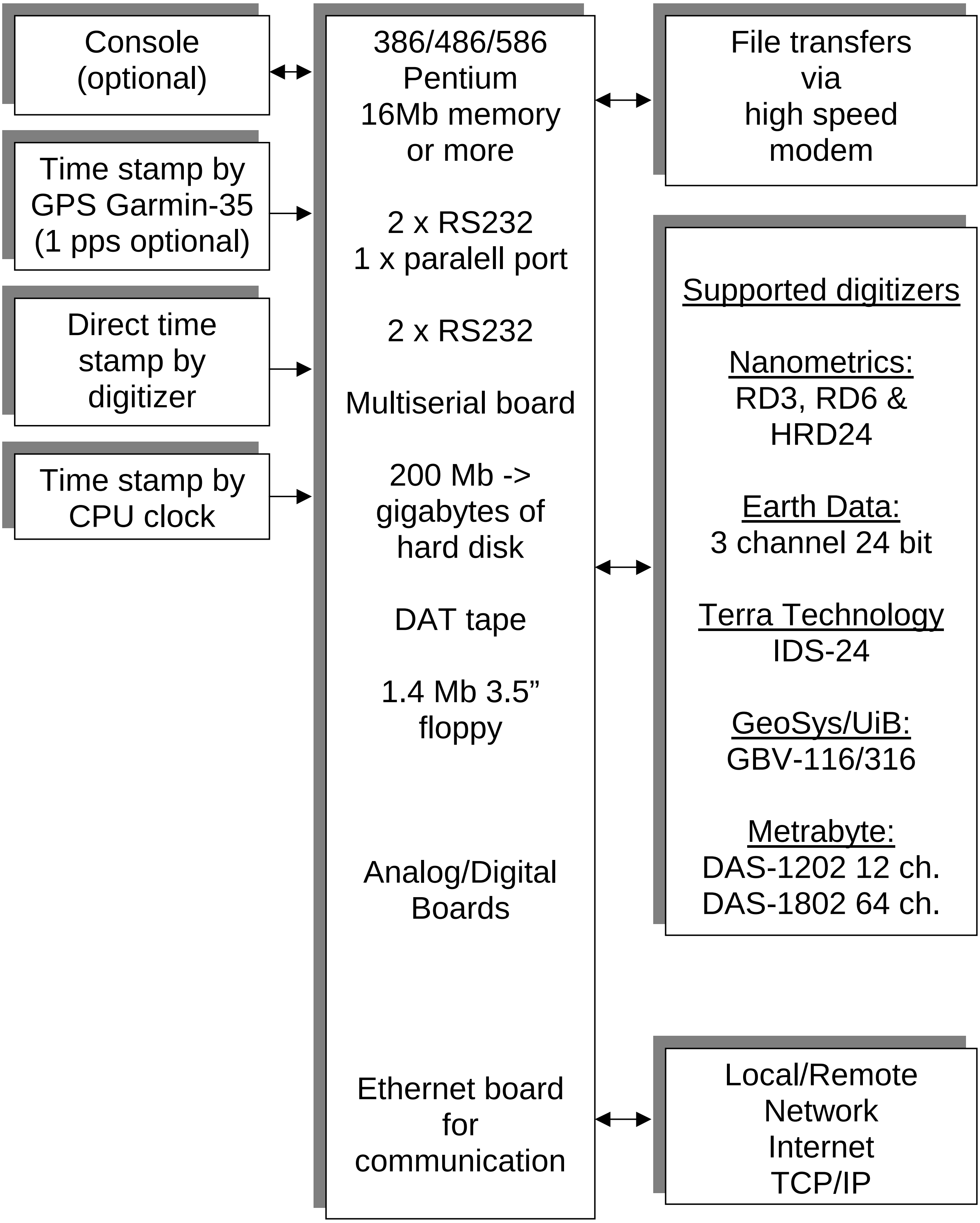
- High precision clock, normally a GPS.

- Communication to other computers by means of for example Ethernet, modem, ISDN, Internet

SEISLOG can use a variety of A/D converters. For conventional analog networks, a 12 or 16 bit, 16 or 64 channel A/D card can be used. For networks with continuous digital transmission by radio, the Earth Data or Terra Technology digitizer can be used. Currently Earth Data delivers digitizers with 16-24 bit and max sample rate of 75 for a 3 channel unit. For stations requiring 6 channels or less, an option is to use the Nanometrics digitizers, which are either 16 bit with gain ranging or straight 24 bit. Max sampling rate for a 6-channel system is 100 Hz. For stations requiring up to 8 channels or less, an option is to use the IDS-24 from TerraTechnology, which is a straight 24-bit digitizer. This unit also has the option of variable sampling rate. GeoSys delivers a range of different digitizers for different purposes, with different resolution and sample rates. Some of these units can now be delivered with an agreed communication format supported by this version of SEISLOG. The Nanometrics, Earth Data, GeoSys or TerraTechnology digitizers connect to the PC via a serial line or a multiserial board if several units are used.

Fig.1.1

Typical recording unit



Console (optional)

Time stamp by GPS Garmin-35 (1 pps optional)

Direct time stamp by digitizer

Time stamp by CPU clock

386/486/586 Pentium
16Mb memory or more

2 x RS232
1 x paralell port

2 x RS232

Multiserial board

200 Mb -> gigabytes of hard disk

DAT tape

1.4 Mb 3.5” floppy

Analog/Digital Boards

Ethernet board for communication

File transfers via high speed modem

Supported digitizers

Nanometrics:
RD3, RD6 & HRD24

Earth Data:
3 channel 24 bit

Terra Technology
IDS-24

GeoSys/UiB:
GBV-116/316

Metrabyte:
DAS-1202 12 ch.
DAS-1802 64 ch.

Local/Remote Network Internet TCP/IP

1.2 Definitions of some important expressions in the text.

- **Shared memory** Memory allocated by SEISLOG processes for easy and fast exchange of data between processes. In this shared memory region are constants, parameters, data arrays, variables for the detection algorithm etc. are defined. The different programs in SEISLOG links up to this memory area and may communicate through it.

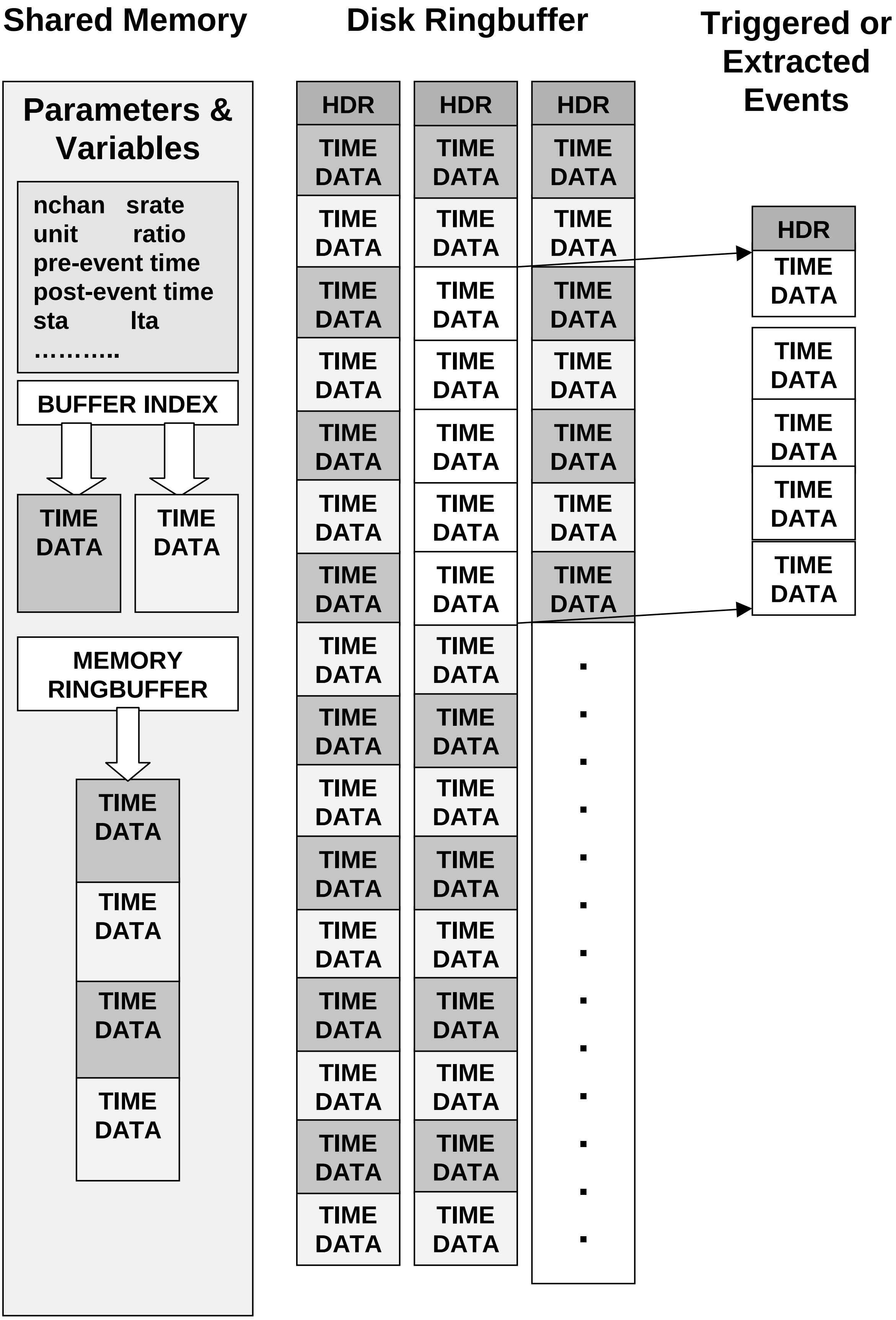
- **Memory ringbuffer** This is pre-allocated space in shared memory where time-tagged in-coming data is written sequentially in one second buffers. When the memory ringbuffer is full, SEISLOG starts over-writing data at the beginning of the memory ringbuffer. Depending on the parameters, this memory ringbuffer may contain from some seconds to hours of data. This ringbuffer works as a high-speed storage device where data can be buffered for some time before entering into the detection algorithm. It may work as the SEISLOG ringbuffer alone, but normally data is copied from this ringbuffer to the disk ringbuffer system.

. When seismic events are detected, data from this memory ringbuffer are copied on to a separate file. The memory ringbuffer must be large enough to hold the full length of the event. 1 Mb of memory is allocated by default for this ringbuffer. In the case of a configuration with a 3-component digitizer with a sampling rate of 50 Hz , the size in minutes is around 100 minutes. In the case of a 32 channel configuration with 50 Hz sampling rate, the size is around 10 minutes.

- **Disk ringbuffer** This is allocated space of the hard disk where time-tagged in-coming data (from the memory ringbuffer) is written sequentially in one-second buffers. The disk ringbuffer is built up of several identical files which together make up the ringbuffer design. The number of files and the size of the files are specified through the parameter program. One file at the time is filled up with data, and when all files have been filled with data, SEISLOG starts all over again from the first file. The disk ringbuffer can be specified to be very big depending on the size of the hard disk. This also implies that continuous recording is automatically part of the system. See Fig. 1.2.

Fig. 1.2

Basic data-flow



1.3 Principal of operation

The SEISLOG software consists of several separate but synchronized programs, which communicate through a shared memory. A device driver program reads 1-second buffers of data from a digitizer unit, time stamps (if not time stamped by the digitizer) the data and copies it on to the memory ringbuffer. A detection program reads the 1-second buffers from the memory ringbuffer, writes the buffer out to a ringbuffer on hard disk, and performs the detection check on the data. The memory ringbuffer works as a delay buffer in case of variable system load. If an event is detected, the specifics of the event are stored in an event record and a signal is sent to the cataloging program which copies the event out from the ring buffer and generates an event file. During this process, all or specific channels may be selected for storing. At the same time, an event list is maintained of the last 1000 triggers indicating trigger time, trigger type (could e.g. be a manual trigger), duration and file name of the event file.

The disk ringbuffer is defined through a parameter program and consists of a specified number of files each with a specified length. Depending on the size of the hard disk and the type of recording you want to do, you may specify a number of files of a specified size to make up the total ringbuffer scheme. Each file in the ringbuffer has the same format as an ordinary event and can then be treated as such except that is does not have a detection number.

This ring buffer scheme means that continuous recording is automatically supported. The amount of data stored in continuous form is always the amount of disk space allocated for the ring buffer.

The event detection is based on a conventional STA/LTA algorithm. User specified channels are used for the detection. For each channel used for detection, the signal may be filtered, either band-pass or using a simple difference filter and STA and LTA are calculated. When the STA/LTA ratio is above a given limit or the STA is above a certain level (level trigger), a detection is declared for this channel. An event start is declared when a given number of channels detect within a specified time window. The end of the event is when no more channels are above the de-trigger STA/LTA level. Data is then saved with a given pre-event and post-event interval of data limited by a set maximum recording time. The limits on both maximum recording time and pre- and post-event memory are dependent on the size of the memory ring buffer, so in practice there is no strict limitation. All trigger parameters can be set individually for each channel.

The user can at any time turn the recording on. This means that from that moment, all channels will be saved in an event file until the recording is turned off. This can be used for recording at given times or during tests.

The disk ringbuffer may contain from minutes to days of continuous data from all digitized channels. Any time interval can be extracted and the resulting event file will enter the normal event directory and be recorded in the event list.

In order to handle different types of data, the system can be set up with up to 5 different parameter sets, each corresponding to a given set of channels. The system will trigger if the trigger criteria is fulfilled for any of the trigger set and a separate event file will be created for each trigger set and likewise identified in the event detection list. For each trigger set, the resulting detection file will only contain the channels specified at sample rates equal to or lower than the original sample rate. The use of this is best described through some examples.

Example 1:
A small analog network records centrally using SEISLOG with a 16 channel A/D board sampling at 50Hz. To the same board is connected a 3-component LP seismometer. The network has it own trigger set and the LP seismograph another set making it possible to let the network trigger on local events and the LP seismograph trigger on distant events and generating event files with different sample rate and duration.
In this case it would also be possible to have continuous recording instead of or in addition to the triggered mode on the LP channel.

Example 2:
A telemetered network has some stations dedicated to vulcanological observations. One or two of these stations sometimes record hundreds of events a day which are not observed on the rest of the network, but never the less should be taken care of. In this case one parameter set is focusing only on the vulcanological stations making it possible to separate out channels with volcanic events from a few stations without saturating the system.

The SEISLOG system will start storing an event as soon as the trigger flag is lowered. However, it cannot finish the storing of the event before the post event memory time has expired. This means that in case of large events there can be a delay of several minutes before the event become available to the user. Using a parameter set to trigger only on strong events and cut out recording after a short time, e.g. 60 seconds can solve this problem.

1.4 Timing

Timing is done by synchronizing an internal software clock with an external GPS. The data buffers are time tagged by requesting the time from the software clock. The clock is updated every second by the external clock. However, the synchronization can take place at other intervals, which could happen if the external clock temporarily stops sending time information. Between updates the clock operates with the accuracy of the CPU's internal timer which is usually not very good.
This version of SEISLOG supports the Garmin 35 with a very accurate 1 pulse per second. The 1 PPS is fed into an interface for signal conditioning (described in this manual). The pulse is connected to the parallel port and synchronizes the software clock to within 1 millisecond.
When the 1 pps is not present, the time from the GPS is used as time source without synchronization from the 1 pps. In this case the time tag in each buffer is marked NSY, for Not Synchronized. However, the time tag will be accurate for a reasonable time ahead.
Newer digitizers supplies timing in the data stream as an option. This version of SEISLOG supports Garmin 35 and RSC8000 from Radiocode.

1.5 Data transfer to a host computer

A data acquisition system is critically dependent on the capability of transferring the data to a host computer. In SEISLOG, this can be done in several ways. Most systems use either the serial line or FTP via Ethernet. Using FTP is simply a case of copying the file to the host computer, while using the serial line most applications use a particular SEISLOG program listing the event file on the screen in compressed ASCII (MNP6, GSE format) form, and the receiving end use a communication program to capture the file. The two other methods used are streamer tape or floppy, depending on amount of data. Whichever way the files are transferred, a single program is used on PC or Sun to read and demultiplex the event files.
Many SEISLOG's can be connected together using modems or Internet with the SEISNET software, see 4.3.

1.6 Comments this release

Version 8.50 of the SEISLOG software is not a mayor update of the software. Some bugs have been fixed. New drivers have been implemented for the new Earth Data digitizer and for the digitizer designed by Ramon Ortiz. Some options in the parameter program PAR have been removed. This is basically to reflect what digitizer units that are supported. For the user, most options from the previous releases are also included in this version. Some new programs have been added as well as some new options.

1.7 Work in progress

- Improve stability

1.8 Licenses

The SEISLOG software is free of charge, but you need a written approval from the Institute of Solid Earth Physics - Seismological Observatory. The complete address is on the front page of this manual.
The SEISLOG software may NOT be used for any commercial purposes unless agreed upon with the authors.
Before installing the SEISLOG software, the user has the responsibility of getting a legal license of the products mentioned in this manual.

All system software (QNX, Watcom C, TCP/IP, MS-DOS etc) are licensed products, and the users are responsible of getting such a license.

2. INSTALLATION, UPDATE and BACKUP

2.1 Installation, Update and Backup

This chapter will describe the basic installation of the QNX operating system including the C compiler. It will also go through the installation of the SEISLOG software. It is also described how the startup file may be or should be modified to fit the actual configuration.

2.2 PC requirements to run SEISLOG

As a general rule QNX and SEISLOG can be installed on any standard PC. To install MS/DOS and SEISLOG you need a 486 or Pentium with 16-32 Mb of RAM, hard disk capacity of 200 Mb or more, and a 1.4 Mb 3.5” floppy or a CDROM. Depending on the timing system and the type of digitizer you want to use, 1 or 2 serial lines are also needed. To use the PPS synchronization, the parallel port is utilized.

2.3 Partitioning of hard disk

A SEISLOG system can have two partitions on the hard disk. One partition can be used to configure the PC as a standard Win95, Win98 system or Win2000. The QNX operating system can be installed in the other partition. The PC can be set up to boot the QNX operating system by default, and SEISLOG may start data acquisition automatically at power up. Win95, Win98 or Win2000 can be loaded by pressing the key indicating the partition that holds this operating system. The PC will then work as a standard PC.

It is important to notice that the user may access files in the Windows partition directly from QNX, but not the other way around.

2.4 Step by step installation of QNX operating system

The QNX4 distribution contains several floppies including the complete operating system and the C-compiler. The distribution is also available on CD. The distribution is a licensed product from QNX Software Systems.

The QNX4 operating system can be installed on standard PCs from the 486 and upward. The hardware configuration for the SEISLOG application must include a hard disk, a 3.5” floppy drive and it is recommended to install at least 32 Mb of RAM.

If you want to have a separate partition on the hard disk running Windows, this should be installed **before** you install QNX. If Windows already exists on the disk and occupies the whole disk as one partition, the disk should be re-partitioned and Windows re-installed.

When the above criteria are met you start the installation of QNX.

2.5 Step by step installation from floppy

STEP 00 insert the QNX boot floppy into the floppy drive and reset the computer.
The PC will now boot from the floppy and end up with the QNX logo on the screen. On the last line you will find the prompt #.

STEP 01 type **install** and press return.
The install-script will tell you step by step what to do.

After the installation has finished with the last line showing the # prompt, you reboot your system by pressing reset. Do **NOT** create any root account on this stage.
When the system has rebooted, the line:
login:
Will show up on your console.

STEP 02 type **root** and press return.
Now a line showing the date will show up on the screen followed by the prompt #.

STEP 03 Installation of the C-compiler. Not needed if you are using a pre-compiled version of SEISLOG.
insert the first floppy of the C-distribution into the floppy drive and
type **vol -r /dev/fd0 | pax -rv** and press return.
The installation procedure will ask for the next floppy until all three are copied onto the hard disk.

Type: **/etc/install -u /updates/wcc95/Released/wcc952.tar.F** and press return

The C-compiler will now be installed.

STEP 04 Install the license floppy
type: **license** and press return
Answer **N** and you are finished

STEP 05 Install the first TCP/IP floppy
type: **/etc/install** and press return
The installation procedure will ask for the next floppy. When finished, the TCP/IP is installed, and you have to update the sysinit.1 file and also the hosts file.

When the installation has finished you have a fully operating QNX system running on your PC. Now you can install the SEISLOG application software or start making your own programs, but first you have to create some user accounts, see section 2.5. This is also described in detail in the QNX manuals

2.6 *Step by step installation from CD*

The installation from CD is self-explanatory. Change the BIOS on the PC to boot from CD, insert the CD and reboot.

2.7 Step by step installation of SEISLOG

The installation of the SEISLOG software assumes that a QNX operating system has been installed on the computer. The hardware configuration must include a 3.5" floppy drive for installation. The distribution of SEISLOG is stored on 1, 3.5" high-density floppy in compressed tar format.
The floppy includes all files necessary to do a complete installation of SEISLOG on your PC. The installation procedure described here includes some extra steps, which normally are necessary only for the first installation after installation of the operating system.
The installation procedure should follow immediately after the completion of the installation of the QNX operating system.
For later updates or upgrades of the application software, follow the steps in section 2.10, or a separate Update procedure supplied with the distribution.

The distribution floppy contains all the necessary programs to run SEISLOG.
The distribution includes all source code, installation procedures and drivers for the different digitizers and clocks supported.

Below is a step by step installation of the system with a short explanation of the different steps.

<u>STEP 00</u> Reset the PC to reboot the machine

<u>STEP 01</u> Setup user accounts.
You should only go through this step if SEISLOG is not installed on the PC or if you know that the accounts does not exist.

type: **passwd seislog** and press return
The machine answers:
User id # (number) press return
The machine answers:
Group id # (number) press return
The machine answers:
Real name () type **seislog** and press return
The machine answers:
Home directory (/home/seislog) press return
The machine answers:
Login shell (/bin/sh) press return
The machine answers:
New password: type **seislog** and press return (the password is not echoed)

The machine answers:
Retype new password: type **seislog** and press return (password not echoed)
The machine answers with the prompt #, and the account is created.

type: **passwd events** and press return
The machine answers:
User id # (number) press return
The machine answers:
Group id # (number) press return
The machine answers:
Real name () type **events** and press return
The machine answers:
Home directory (/home/events) press return
The machine answers:
Login shell (/bin/sh) press return
The machine answers:
New password: type **events** and press return (the password is not echoed)
The machine answers:
Retype new password: type **events** and press return (password not echoed)
The machine answers with the prompt #, and the account is created.

type: **passwd seismo** and press return
The machine answers:
User id # (number) press return
The machine answers:
Group id # (number) press return
The machine answers:
Real name () type **seismo** and press return
The machine answers:
Home directory (/home/seismo) type **/home/work** and press return
The machine answers:
Login shell (/bin/sh) press return
The machine answers:
New password: type **seismo** and press return (the password is not echoed)
The machine answers:
Retype new password: type **seismo** and press return (password not echoed)
The machine answers with the prompt #, and the account is created.

.

type: **logout** and press return

STEP 02 login as user seislog:
login:**seislog**
Password:**seislog**
type: **su** and press return
type: **cd /home** and press return

STEP 03 Create some directories:

mkdir resamp
mkdir rngbuf

Insert the floppy containing the SEISLOG distribution into the floppy drive

STEP 04 type: **vol -r /dev/fd0 | melt | tar -x** and press return

STEP 05 type: **cd seislog** and press return
type: **cc c.c -o c** and press return
type: **c** and press return
type: **chmod a+s *** and press return
type: **cd /home** and press return
type: **chmod 777 *** and press return
type: **cd /home/seislog** and press return

At this point the application software is installed and compiled and ready to use. The first thing to do now is to set up a temporary parameter set for your seismic network configuration.

2.8 Adjusting the sysinit.1 file for your configuration

In QNX there is a scriptfile that is executed each time the computer is booted. The file is called *sysinit.1* and is stored in /etc/config.
In chapter 10 is an example of a typical *sysinit.1* file. Some of the lines in the file has a '#' in the first column. This means that the line is commented out and that the action on the line is not performed during startup. By editing this file the startup procedure for your system can be set to fit your requirements.
Be aware that some of these commands may perform differently on different computers. All the commands can, and should be tested manually from the console before they are activated in the *sysinit.1* file.

2.9 Setting up the disk ringbuffer system

Before you start SEISLOG the first time or when you modify certain parameters you should re-allocate the disk ringbuffer. Parameters that influence the disk ringbuffer are number of channels, sample rate, digitizing unit, minutes per ringbuffer file and the number of files in the ringbuffer.
If there is already an old SEISLOG version installed, you should turn the PC off and on again at this point.

All parameters are set with the **PAR** program. It is important to decide a reasonable size of the ring buffers and also the number of files you want to allocate.
To allocate the ringbuffers and run the RE_INITI program you MUST be superuser.
After all parameters are set you have to run the <u>RE_INITI</u> program to allocate the ringbuffers. The program will also reset the detection number to 0. First event will have detection number 1. Old ring buffers are removed with this program. Old events are <u> NOT</u> removed but will no longer have a reference to a detection number.

2.10 Update SEISLOG

Normally updates of SEISLOG are distributed as complete distributions. This also means that an update will follow the same installation procedure as for full distributions with a few exceptions. If SEISLOG already is installed, there is no need to establish the SEISLOG user accounts.
Go through the following steps to update your SEISLOG system:

login as user seislog:
login:**seislog**
Password:**seislog**
type: **su** and press return
Password: answer with password if implemented
type: **cd /home** and press return

Insert the floppy containing the SEISLOG distribution into the floppy drive

type: **vol -r /dev/fd0 | melt | tar -x** and press return
type: **cd seislog** and press return
type: **cc c.c -o c** and press return
type: **c** and press return
type: **chmod a+s *** and press return

Turn PC off and on at this point.

2.11 Backup SEISLOG to floppy, source distribution

To make a backup of the SEISLOG software you may use the procedure for making a distribution. This backup can later on be used as a distribution for an installation.
Note that the procedures ONLY make a backup of the application software. No recorded events or user written programs will be saved.
The program bck_stru is used to make backups of the application software. The bck_stru with no arguments, make a backup of all source and needed file to generate an executable SEISLOG. The C-compiler is needed to do this.

The bck_stru with the argument 1 (bck_stru 1), makes a complete backup of SEISLOG, including executables. To make a distribution to floppy, only the source version can be used.

<u>STEP 01</u> Login as:

login: **seislog**
Password: **seislog**

<u>STEP 02</u> insert high density floppy (1.4Mb)

<u>STEP 03</u> type **bck_stru** and press return

This command will make up a backup directory structure if it does not exist and copy all necessary files into this structure to form a complete distribution of SEISLOG.

<u>STEP 03</u> type **qnx_dist** and press return

The procedure will copy the whole SEISLOG directory structure to the floppy. The files on the floppy are **compressed**, and may **NOT** be listed or copied directly from floppy.

2.12 Backup SEISLOG to floppy, update only

To make a backup of the SEISLOG software you may use the procedure for making an update distribution. This backup can later on be used as a distribution for an update installation. This backup will not save the current parameters. An installation of this backup will not change the parameters in your system.

Note that the procedures ONLY make a backup of the application software. No recorded events or user written programs will be saved.

<u>STEP 01</u> Login as:

login: **seislog**
Password: **seislog**

<u>STEP 02</u> insert high density floppy (1.4Mb)

<u>STEP 03</u> type **bck_src** and press return

This command will make up a backup directory structure if it does not exist and copy all necessary files into this structure to form a complete distribution of SEISLOG.

<u>STEP 03</u> type **src_dst** and press return

The procedure will copy the whole SEISLOG directory structure to the floppy. The files on the floppy are **compressed**, and may **NOT** be listed or copied directly from floppy.

2.13 Backup SEISLOG to file

To make a backup of the SEISLOG software you may use the procedure for making a distribution. This backup can later on be used as a distribution for an installation. Note that the procedures ONLY make a backup of the application software. No recorded events or user written programs will be saved.

STEP 01 Login as:

login: **seislog**
Password: **seislog**

STEP 02 insert high density floppy (1.4Mb)

STEP 03 type **bck_stru** (source version) or **bck_stru 1** (executable version) and press return

This command will make up a backup directory structure if it does not exist and copy all necessary files into this structure to form a complete distribution of SEISLOG.

STEP 03 type **cd /home/backup** and press return

STEP 04 type **tar -cvf /home/seislog.tar - .** and press return

The procedure will copy the whole SEISLOG directory structure to the file /home/seislog.tar. This file can then be copied to another machine with ftp, and can then be used as a distribution file, or installation file.

If you copied the distribution file seislog.tar to /home,

type **cd /home** and press return
type **tar –xvf seislog.tar**

and SEISLOG is installed under /home/seislog directory.

2.14 Download SEISLOG and documentation from Internet

A complete distribution of SEISLOG and documentation can be downloaded from ftp-server ftp.ifjf.uib.no . The SEISLOG distribution is stored as a tar file seislogxx.tar, where xx indicate the version number. The documentation is stored as a Word file as seislog.doc

The SEISLOG distribution must be downloaded to the /home directory on your QNX system. The QNX system has to be prepared according to the installation procedure described under section 2.4 in the manual. The part describing the installation from floppy diskette should be skipped. The next step in the installation should be as described in section 2.7, STEP 05.

Be aware that to run the SEISLOG system, you will need the licenses described in section 1.8

2.15 Prepare tar distribution file to disk

STEP 01 **cd /home/seislog** and press enter

STEP 02 **bck_stru** (source version) or **bck_stru 1** (executble version) and press enter

STEP 03 **cd /home/backup** and press enter

STEP 04 **tar -cvf /home/seislog850.tar - .** and press enter

A complete distribution file has now been stored in /home as seislog80.tar. '850' in the filename indicate the version.

2.16 Installation of SEISLOG from a tar file.

If you already have installed SEISLOG on your computer you can proceed directly to STEP 02

If this is the first installation of SEISLOG, you have to follow the steps below.

STEP 01 Follow the steps in section 2.7 Step by step installation of SEISLOG STEP 00 through STEP 03, and skip STEP 04

STEP 02 **cd /home** and press enter

STEP 03 **tar –xvf /home/seislog850.tar** and press enter. Then you proceed with STEP 05 in section 2.7

2.17 Use of tape for backup of events and ringbuffer system.

If you have a DAT tape installed on your system, you may take backups of all your events and the ringbuffer system with the program *tape_sav*

The tape drive is normally installed as a SCSI device. See chapter 10 sysinit.1 on how to install the tapedrive.

3. USING SEISLOG

This chapter will discuss topics like parameter settings, user SEISLOG accounts, and the different outputs from SEISLOG.

3.1 HELP

To get some information on a routine, you may type the command **help**. This will give you a summary of programs and utilities in SEISLOG. To get information on a specific routine, you type **use 'name'**. Not all routines have help-files.

3.2 Setting the parameters

SEISLOG supports the option to let the trigger algorithm run through more than one set of parameters with different trigger criteria i.e. the data entering the system will be checked for events 'n' times if we activate 'n' parameter sets. You are allowed to specify up to 5 different parameter sets.
The advantage of using multiple parameter sets is the possibility to record data based on different criteria.
How to use this option very much depend on the type signals (LP, SP) entering the system, the design of the seismic network and how much data you want to record.

The different parameter sets let the trigger algorithm:

- Look at the same channels or different channels or overlapping channels.
- Use different trigger parameters on the same channels.
- Use different pre-event, post-event and array-propagation windows
- Recording of different channels

Before starting SEISLOG, all the parameters must be set to reasonable values in the parameter files. When starting the data acquisition for the first time it is a good idea to use only one parameter set. When this works normally you may introduce more complex parameter settings step by step.
The program PAR updates the parameter files. Each parameter is described in more detail in section 3.3.
It is important to be aware that this option may have a dramatic influence of the performance of the system unless it is used with care. For example to specify 25 channels for the detection algorithm, all with a bandpass filter in 5 parameter sets may cause problems due to limited CPU resources. The idea is to use the different options in the parameter sets to optimize the detection rate without overloading the system.
At the moment there are only a few restrictions on the parameters to give the user the maximum flexibility. On the other hand, this leaves some of the responsibility on the user when setting up the parameters.

In this version the parameters go through a check for obvious errors. This check will not catch all conflicting parameter settings, but for example inform the user of possible wrong serial ports selected, wrong baudrate for a particular digitizer, wrong number of channels for the selected digitizer etc.

3.3 Using the PAR program

The PAR program let you maintain your parameter sets. **This program updates parameter set #1 by default.**

To start the utility you simply type:

PAR and press **return** on your console.

To update another parameter set you give an argument to indicate the parameter set number:

PAR 3 for parameter set #3
PAR 4 for parameter set #4
etc.

NB! When modifying parameter sets other than number 1, be aware that most of the parameters on page one can NOT have other values than zero. This is valid from parameter 08 Number of channels and upward.
If you try to enter a value other than zero, you get a flashing message at the lower part of the screen, and you can not move the cursor to other parameters.

The program operates in full screen mode and supports the use of the arrows.
The arrows will move around in the parameter file and highlight the current parameter. You may enter a carriage return to blank the field before entering a new value or just write in the new value immediately. Press return and type in the correct value. On the bottom line there are some information on the current parameter. If you type in a value that is out of range or illegal, a blinking warning message will be written above the info line. You will not be allowed to go on to another parameter before you have corrected the current parameter to a legal value.
Five pages on each parameter set are available through the PAR program. You may switch from one page to the next by pressing the **TAB key**. The first page contains parameters that are not channel dependent, like sample-rate, type of clock, pre-event memory etc.
Page 2,3,4 and 5 describes channels 1-16, 17-32, 33-48 and 49-64 respectively.
To quit the program you press the **ESC key**. You will then be asked to enter two comments describing what changes you have made. Press RETURN to exit the program. Your parameter set is now updated and you may modify another one.
On the next page is a typical parameter set:

page #1

```
02 Network Name______________:   WNN 17 Number of Parameter-sets_:       1
03 Minim. number of Triggers:     2 18 Disk space level_________:       0
04 Pre-Event_Memory_________:    10 19 Duration minute pulse____:    1000
05 Array Propagation Window_:    20 20 Event duration low limit_:       5
06 Post-Event-Memory________:    10 21 Minutes per ring-buffer__:       3
07 Maximum Recording________:   100 22 Number of ring-buffers___:      10
08 Number of Channels_______:     3 23 SPARE.......................:       0
09 Sample Interval__________: 0.020 24 SPARE.......................:       0
10 Resampling (3-comp only).:     0 25 SPARE____________________:       0
11 Type of Clock____________:     2 26 Auto Processing__________:       0
12 Digitizing Unit__________:    31 27 Send Mail:_______________:       0
13 Automatic reboot on error:     0 28 Base Address 1 A/D_______:     200
14 Time Correction__________: -2.00 29 Base Address 2 A/D_______:     300
15 SPARE.......................:     0 30 Port/Baudrate clock______:  2,9600
16 A/D board Interrupt______:     5 31 Port/Baudrate digitizer__: 1,19200
```

NETWORK name: 3 letter code

page #2

```
Cha Name Comp Gain Filt_Low Filt_Hig  STA    LTA Trig-Lev De-Trg-Lev TT RC
__________________________________________________________________________
  1  SUL S  Z    1      0.1      5.0  3.5  400.0      3.0        2.0  1  1
  2  BER S  Z    1      0.5     10.0  3.5  400.0      3.0        2.0  1  1
  3  HYA S  Z    1      5.0     10.0  3.5  400.0      3.0        2.0  1  1
  4  ESG S  Z    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
  5  ASK S  Z    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
  6  ASK S  N    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
  7  ASK S  E    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
  8  COL S  Z    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
  9  COL S  N    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
 10  COL S  E    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
 11  SNT S  Z    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
 12  SNT S  N    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
 13  SNT S  E    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
 14  KON A  Z    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
 15  KON A  N    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
 16  KON A  E    1      0.0      0.0  0.0    0.0      0.0        0.0  0  0
```

Page #3 to # 5 is identical to page #2 but define channels 17-64.

02 This parameter gives the name of the network. Here you can enter a three-character network name. This name will later be used as the file extension in the event files. The field is 4 characters long, but you MUST enter only 3 characters.

03 The minimum number of triggers needed within the array propagation window to declare an event. If a configuration has a total of for example 16 channels and 7 are marked to run the detection algorithm, this number is the minimum of the 7 that needs to be in trigger mode.

04 Pre-event-memory is the number of seconds to record BEFORE the system trigger.

05 Array propagation window is the number of seconds within which the trigger criteria has to take place.

06 Post event memory is the number of seconds to record AFTER the system trigger flag is turned off.

07 Maximum recording is the maximum number of seconds to record in case of events with a long duration. This is to put a limit on how big each file can be. The system counts seconds of duration from the detection flag is raised and permit the system to count up until this number. The recording however, will start from the start of the pre event and record up until the specified number in this parameter.

08 Number of channels is the total number of channels to digitize.

09 Sample interval is given in seconds and should always be present in the parameter file even though it is indirectly given by for example the setup of the Nanometrics or by the frequency supplied from an external source.

10 Resampling. When this parameter is set to 1, you will be taken into the editor when you exit the par program. Here you can modify the resampling parameter file according to what is described in section 3.4.

11 Type of clock is a number indicating which type of clock that is used in the configuration:

0	CPU clock
1	Garmin-35, No PPS support
2	Garmin-35, PPS support
4	Radio Code Clock RSC8000
8	Direct time tagging by digitizer

Clock type 2 needs the signal conditioning interface described in section 6.9 and figure after 6.10.

12 Digitizing unit is a number to indicate the type of unit that is used for digitizing in the configuration:

10	GBV	GeoSys/UiB	16 bit	3	channels
30	DAS1202	A/D board	12 bit	16-32	channels
31	IDS-24	Terra Tech.	24 bit	8	channels
32	GeoSys	Boreholesys	18 bit	12	channels
36	DAS1802	A/D board	12 bit	64	channels
38	RO	R.Ortiz	16-24	1-3	channels
40	HRD24	Nanometrics	24 bit	03	channels
53	RD3	Nanometrics	16 bit gain ranging	03	channels
56	RD6	Nanometrics	16 bit gain ranging	06	channels

63	ED24	Earth Data 24 bit	03	channels
64	ED2400	Earth Data 24 bit, GPS integrated Type of clock must be set to 0=CPU. If GPS not synchronized at startup, the current CPU time will be used. When GPS synchronizes, the GPS time will be used.	03	channels
70	OR+CA	24 bit, 9600 baud	03	channels
96	Synthetic	Synthetic data for testing	n	channels
99	ILI	Earth Data 24 bit		

13 Automatic reboot on error. If set to zero the reboot/restart function is not active. If set to one, the system will be rebooted whenever an error is discovered. A program will monitor if all vital programs are running. If for example a digitizer is not receiving data from the serial port, this is considered an alarm situation and the system is rebooted. In some situations the serial ports enter a “hang” status. In some cases this can be resolved by a hard reset of the systems. In other situations the serial port on the digitizer or the computer is damaged, and the error situation will continue until the port is repaired.

14 Time correction is the number of seconds to add/subtract to the time received from the clock to compensate for delays in for example the internal buffering of data from the Nanometrics units. Filters in the Nanometrics units may also cause some delays in the data. These delays can only be found by digitizing a signal giving a correct minute mark and compare the actual time recorded at that moment. The difference in time may be used in this parameter. Both positive and negative numbers are accepted. In the table below are some measured delays for different configurations of digitizer and clock:

Nanometrics RD3	Garmin-35	-2.150
Earth Data	Garmin-35	
IDS-24	Garmin-35	-0.120
Metrabyte 1202	Garmin-35	-0.95
RO	Garmin-35	-1.0

15 Not available.

16 A/D board Interrupt. Only active when using DAS1202 or DAS1802. Interrupt number used by A/D board. Can not conflict with other devices. Normally 5 will work.

17 This parameter is used to inform the detection algorithm about how many parameter sets to go through. It is only a valid parameter in parameter set #1.
NEVER use a higher number here, unless you are sure you want to use more than one parameter set. Parameter sets 2 and higher **MUST** be defined properly for SEISLOG to work.

18 Not availlable

19 Length of minute pulse output through parallel port in milliseconds.

20 Event duration limit. Events with duration less than the number of seconds specified are not recorded. Only the detection record is recorded, but with no filename.

21 Minutes per ring-buffer. Here you specify the length of each file in the disk ringbuffer system in minutes. Whenever this number is modified, the ringbuffer system must be reinitialized by the utility **re_initi**.

22 Number of ringbuffers. Here you specify the number of files to make up the disk ringbuffer system. Whenever this number is modified, the disk ringbuffer system must be reinitialized by the utility **re_initi**.

23 Not available

24 Not available

25 Not available

26 Autoprocessing. This parameter can be set to four different options:

0 - no actions
2 - execution of the script auto_ch
3 - output of event in SAC format
23 - execution of auto_uch and output of event in SAC format

The auto_uch script is a special option used by University of Chile. It sends the file name of the last event to an e-mail address. Other users may edit this script and modify the e-mail address if found useful.

Output of the event in SAC format can be useful for users using the SAC processing routines. Be aware that if this option is set, the event will be recorded both in SEISLOG format and in SAC format. This will eventually fill up the disk unless cleaned up.

27 Send mail. The parameter has two options: 0 – do not send mail, 1 – send mail. When you specify a 1 for this parameter and later exit the par program, you will enter the editor with the list of e-mail addresses. Here you specify the addresses where you want to send the e-mail. Be careful not to specify addresses to persons that have no interest of this information. When set to 1, the event record (same as each line output from program LGS) is sent to the e-mail address specified in the file described above.
Note: For the e-mail option to work, you have to make the directory **queue** under **/usr/spool**

28 Hexadecimal base address of first A/D board, channel 1-16. Has to be set according to DIP switches on board. Typically 200

29 Hexadecimal base address for second A/D board, channel 17-32. Has to be set according to DIP switches on board. Typically 300

30 Number of the serial line where GPS is connected, and baud rate of serial line. Normally 2,9600 respectively. 1 is COM1 and 2 is COM2

31 Number of the serial line where digitizer is connected, and baud rate of serial line. Dependant on digitizer type. 1 is COM1 and 2 is COM2.

Page #2 and #3 give information about channel specific parameters. For this version 9 different fields are defined:

Name Four-letter code for station name. The name may be typed in lower case or upper case letters. After pressing RETURN the name will be right justified. There is room for more than 4 letters, but 4 are the standard in SEISLOG.

Comp This is a four-letter code to identify the type of component. The letters may be typed in lower case or upper case letters. After pressing RETURN the component code will be right justified. There are room for more than 4 letters, but 4 is the standard in SEISLOG/SEISAN. If the data is going to be processed in SEISAN, the following conventions must be followed.

character 1	defines the instrument type:	
	S	short period
	L	long period
	B	broadband
	A	accelerometer
character 2,3	any information like ‘L ‘ for low gain	
character 4	defines the component.	
	Z	vertical
	N	north-south
	E	east-west

Gain This is an integer number to set the gain for the actual channel. Different digitizing units allow different gains:

DAS1802 unit 36: valid

Filt_low Low cut frequency.
This and next parameter define the band pass filter for the data entering the detection algorithm. Note that original data is passing

through the system without filtering. Data stored in ringbuffers or as events are original data. The filter used has 4 poles. Do not use low cut frequencies lower than 0.005. This may cause numerical problems. Low cut frequency

Filt_Hig High cut frequency

STA The length of the Short Term Average in number of seconds (floating point number).

LTA The length of the Long Term Average in number of seconds (floating point number).

Trig-Lev This parameter should be defined together with the parameter TrgTyp that is the algorithm type. For the moment there are three types of trigger algorithms: STA/LTA ratio, signal levels and continuous recording. When the STA/LTA is used, this parameter describes the minimum trigger ratio STA/LTA. The parameter is typed in as a floating-point number. When the continuous recording mode has been specified, this parameter has no effect.

De-Trg-Lev This parameter is defined the same way as the previous, but is the criteria to de-trigger the event (floating point number).

TT This parameter decides if the component is to be processed by the detection algorithm. 0 – not processed, 1 – processed.

RC This parameter is used to indicate the channels you want to record when an event is detected. A zero means no recording, a one means recording.
A normal situation may be that most channels are marked for recording in parameter set one while some specific channels are set up for recording in parameter set two if there are some special detection criteria for these channels.

3.4 Resampling of data

The first version of resampling is included in this version of SEISLOG. Resampling works on 3 component stations. To setup resampling parameters a program *par_smp* is supplied. The program starts the QNX text editor, and loads the resampling parameter file */home/seislog/pr/par_smp*:

```
ACT/PAS   ORDER    FL     FH   RATE HZ   TYPE
0         2        0.1    0.1  1         LP
L__Z
L__N
L__E
00-07     1 1 1 0 0 0 0 0
```

```
08-15     0 0 0 0 0 0 0 0
16-23     0 0 0 0 0 0 0 0
24-31     0 0 0 0 0 0 0 0
32-39     0 0 0 0 0 0 0 0
40-47     0 0 0 0 0 0 0 0
48-55     0 0 0 0 0 0 0 0
56-63     0 0 0 0 0 0 0 0
```

This version of resampling works only with this channel configuration. Only the first line can be modified:

ACT/PAS 0: indicates no resampling, 1: indicates resampling

ORDER Filter order, 1-5

FL,FH Filter applied to the data. If TYPE=LP, the FL is ignored. If TYPE=HP, the FH is ignored.

RATE HZ Sample interval, in the example 1 sample per second of data. The program can only do resampling at 1 Hz, so this parameter should always be set to 1.

TYPE LP, lowpass filter. HP, highpass filter

With resampling active, a continuous 24 hour file is stored in */home/resamp*.

3.5 SEISLOG user accounts and directory structure

After the installation of SEISLOG there are 4 accounts defined in the password file. All the accounts have the same rights and restrictions. We use 4 different accounts for practical reasons. For example, if you log in to the account EVENTS, your default directory is */home/events* where you find all recorded seismic events. The following accounts are established:

- **seismo** def. dir.: ***/home/work*** def. exec. directory: ***/home/seislog***
  This is the normal account for the operator of the system. From here you may do start, stop, monitoring, change parameters etc.
  .

- **events** def. dir.: ***/home/events*** def. exec. directory: ***/home/seislog***
  If you log into this account you still have the possibility to execute all programs and utilities. The difference is that you are in the directory where all the events are stored.
  The directory is not backed up by the **bck_stru** and **qnx_distr** procedure.
  This directory contains the complete data set for recorded events. Each seismic event has a unique filename based on time, station name and the number of channels. It is of great importance to keep track of the disk space these files allocate. Normally these files should be processed and/or transferred to a central computer and then deleted. In the parameter file you may specify that old events should be deleted when the free space on the hard disk drops below a certain level. Not supported this version. If you do not specify this option, there is NO warning to the user when the disk is full, and you will loose data and even confuse the system by filling up the disk. You may check the disk space with the command FRE.

- **seislog** def. dir.: ***/home/seislog*** def. exec. directory: ***/home/seislog***
  This account is used during the installation phase of SEISLOG and when advised in this manual. The user directory contains the complete source, and should normally not be accessed by the user.
  /home/seislog/pr contains parameter files and current detection number

- **/home/buffer** Ring buffer files
  This directory contains the ring buffer files for continuous recording of data buffers from the SEISLOG, */home/buffer/filename*. Another file in this directory is the small ring buffer for recording detection records, */home/buffer/event*. The files are allocated during installation, but may be altered

anytime as long as there is enough contiguous space on the disk to hold the files.

- **/home/backup** Backup directory for the complete SEISLOG application described under this section. Ringbuffer files and events are not saved here.

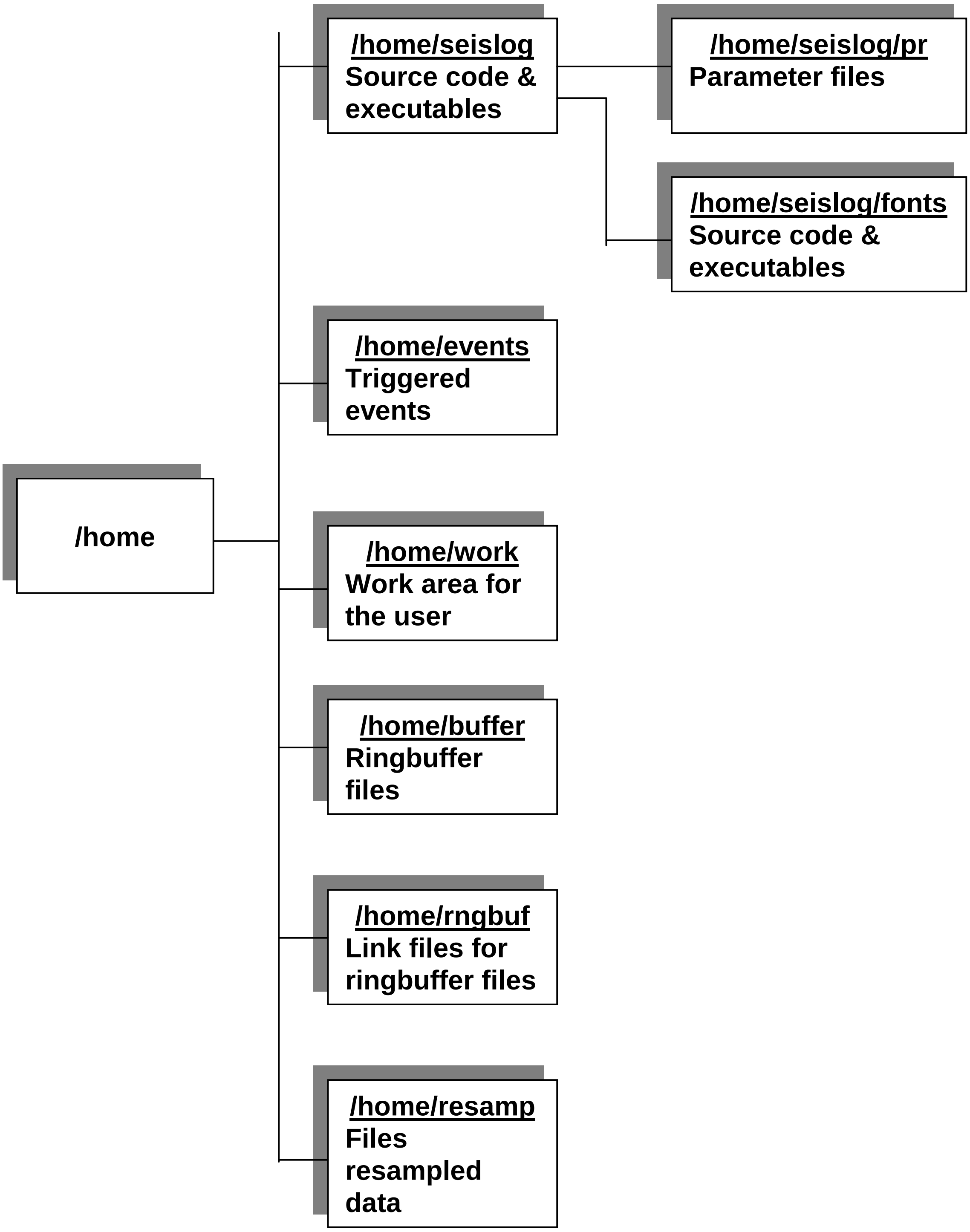
- **/home/resamp** Directory to keep resampled data.

New accounts may be added by executing the passwd utility. See QNX manuals for this operation.
New user accounts will not be backed up by the SEISLOG **bck_stru** and **qnx_distr** procedures.

The SEISLOG software is organized in a directory tree starting at ***/home*** with different subdirectories containing source-files, a/d-drivers, include files etc., see figure 3.1next page. This directory structure can be saved to make a complete new distribution, see section 2.11.
For each user account there is also a separate directory.
One directory ***/home/buffer*** contains the disk ringbuffer system and the detection file. Be aware that only files in the directory tree ***/home/seislog*** and ***/home/seislog/pr*** are saved during a backup procedure.

Fig.3.1

SEISLOG Directory Structure



3.6 Start and stop of SEISLOG data acquisition

SRT: This is the program that will start execution of several other programs to start data-acquisition. SEISLOG may also be started automatically at boot of the system from the QNX startup file: */etc/config/sysinit.1* by including */home/seislog/srt* near the end of the file or simply by typing “srt” from your console.
The following programs are started by *srt*:

RD_PARAM: This program is started by SRT and reads in all the parameter sets from the */home/seislog/pr* directory into shared memory.

The next step for SRT is to decide what timing program that should be started. This version of SEISLOG supports 5 different options, 0,1,2,4,8, where 0 indicate use of CPU clock, in which case no program is loaded, and data buffers are time-stamped by an un-synchronized CPU clock:

CLK_GPP: This timing program is started when option 1 is selected in the parameter file.
A synchronized pulse per second from the clock is fed into the parallel port, if available. This pulse synchronizes the internal clock to an accuracy of 1 millisecond. Timestrings corresponding to the 1 pps is received on a serial port.

CLK_GNP: This timing program is started when option 2 is selected in the parameter file. Timestrings are received on a serial port, and the internal clock is adjusted to the GPS.

CLK_MS1: This timing program is started when option 4 is selected in the parameter file. Work same way as CLK_GPP.

When option 8 is selected in the parameter file, each 1-second data buffer is timetagged by the digitizer. This timestring is extracted and copied into the SEISLOG format.

BUF_PIP: When this program is started, it will check the parameter UNIT and NCHAN to decide what digitizing driver should be started. According to the table above, the correct driver is started.
The driver fills up data buffers and BUF_PIP receives a signal when the buffer is ready. BUF_PIP time-stamps the buffer, writes it to the pipe and waits for the next buffer ready.

DET_EVT: This program reads the time stamped data buffer from the pipe. The data buffer is written to ringbuffer and is also processed by the selected detection algorithm. In the case of an event, a signal is sent to the CAT_EVT program.

CAT_EVT: This program is normally "sleeping", waiting for the detection algorithm to wake it up. When it receives a particular signal, it checks a queue to process any detected events. A detection record is written to disk, and then the actual event is cataloged as a separate file with a filename based on the time of the detection.

SEI_ACT: This program keeps track of when SEISLOG is active. A record is written to disk every 10 seconds telling the last startup-time and the current time.

The SEISLOG system is now active and the computer may be used for other things such as program development, monitoring of SEISLOG, communication etc.

STP: Stop data acquisition if already running.

3.7 Output from SEISLOG

When set up correctly, SEISLOG produces several types of output.

- **Triggered events** these are files recorded according to the current parameter setting. The files are stored in directory */home/events* with filenames based on date and time of detection, network name etc. Below is an explanation of how the name of the triggered events is constructed:

  File name: **1994_08_13_1319_36T.WNS_16_2**

Character	**meaning**
01-04	specifies the year
05	deliminator
06-07	specifies the number of the month
08	deliminator
09-10	specifies the number of the day in the month
11	deliminator
12-13	specifies the hour of the day
14-15	specifies the minutes of the hour
16	deliminator
17-18	specifies the seconds of the minute
19	the ‘T’ indicates that it is a temporary file.
20	deliminator
21-23	name of the network. Three first characters of parameter 02 in the parameter file.
24	deliminator
25-26	number of channels recorded in this event
27	deliminator
28	trigger source:

numeric from 1-5 indicates the parameter set detected the event

‘M’ indicates that the event was forced by the user turning the recording flag on and off using the utilities ONN and OFF, all channels.

‘X’ indicates that the recording was extracted from the circle buffer with the utility EXT, all channels.

‘C’ Ringbuffer files

‘R’ indicates resampled data

- **Manually triggered events** these are files recorded as a result of turning on and off the trigger flag manually (onn and off utility). The files are stored in directory */home/events* with filenames based on date and time of detection, network name etc. as for normally triggered events, but with the last character in the filename set to ‘M’ to indicate a manual trigger. Manually trigger does not include pre-event or post-event memory.

- **Continuous recording** is an integrated part of SEISLOG. By definition the continuously stored data is the current total disk ringbuffer. If this is defined as 20 files each 10 minutes long you have 200 minutes of continuous data at any time. If defined as 500 files each 20 minutes long, you have 10.000 minutes, or close to a week, with continuous data. The files making up the disk ringbuffer are pre-allocated at startup in directory */home/buffer*. Each file has the same format as if it was an ordinary triggered events.

- **Continuous resampled recording** is activated by running the par_smp program, setting the parameter ACT/PAS to 1. From startup of data acquisition, 24 hour files of resampled data are stored in directory /home/resamp. If the data acquisition is stopped in the middle of a 24-hour file, the header in each file contains information of how much data is recorded up until the acquisition was stopped. When acquisition is restarted, a new file is opened.

- **Event list** For each triggered event or manually triggered event a detection record is added to an event list file stored in */home/buffer*. This file can be inspected by the utility lgs to give an overview of the last seismic activity. Each recorded event will have its own detection number assigned to it, and can be referred to by this number by other utilities in SEISLOG.

3.8 Retrieving data from ringbuffer system

The ringbuffer files are defined by the parameter program *par.* The ringbuffer system contains a number of files, where each file has the same length, defined as a number of minutes. Each file in the ringbuffer scheme can be treated as an event file, with the exception that is has not a detection number. The ringbuffer files are generated in the */home/buffer* directory as files with names R0000-Rnnnn, where n is the total number of files. The ringbuffer files can be retrieved by copying files directly from this directory; however, the name convention here is just the sequence number R0000 etc. All physical ringbuffer files has a corresponding link file in directory /home/rngbuf. The link filename has the format R0023_1999_08_13_1319_36T.WNN_16_C. To extract a specified time interval from the ringbuffer system, the user can use the program *ext*. The program will go through the ringbuffer system and inform the user on the status of the ringbuffer system. If the acquisition has been stopped and restarted recently, a message will inform the user. In the case the ringbuffer system is not continuous, there is a chance that you can not retrieve the data window you specify.

3.9 Basic utilities

The software includes several utilities to monitor the current status of the data acquisition. The utilities can be split into two main groups:

Monitoring of real time data acquisition

MON: This utility is the most useful for monitoring how SEISLOG is performing. It gives a continuously updated picture of the different variables in SEISLOG. The screen is divided into two, where the lower half is showing SEISLOG constants from the current parameter file. The upper half shows how the trigger turns on and off flags on different channels, how the STA and LTA for different channels vary, current address in the ring buffer etc. You may also see and hear (if the loud speaker on the PC is connected) when an event is detected. The bell will sound as long as the detection flag is set. When you have an event, the duration counter will increment and will reset to zero when the event is finished. When an event is detected, CAT_EVT will be started immediately and you can watch the ‘To CATLOG:’ value to see how many records are written to disk. This value will decrement to zero unless the event is longer than the MaxBufRec value. In this case the duration, found by running LGS, will appear longer than the actual recorded data.

To monitor parameter set number 2, you type MON 2.

Example:

MON

```
Network:WNN   GPS Garmin                         GPS     :  98%  PPS:        123
Last update:12:24:08.728 01/08/1999 213
Buffer time:12:24:09.171 01/08/1999 213 SET      Rec time:   349  rng file:    4
detflg: 0  ipoint: 0  ndet  :  0  preblk :  0 To CATLOG  :     0  Last DPX:   17
recrd : 0  queue : 0  ndetw :  0  postbk :  0 iblk       :   324  duration:    0

sta:  4635.7        4436.9       11589.2       1528.7
```

```
lta:  4531.9        1928.8        8510.8       1408.6

chaflg:1111000000000000 0000000000000000 0000000000000000 0000000000000000

Samples:
     -4080    13333    185264    14623    6130    298304

ParSet: 1 Ext/Int:     1 ArPrWnd: 60 Min_rng : 131 Timcor:   0.000
Chan  : 6 MaxRec :   500 PreEvt :  5 SmpBufCh:  50 S-Rate:   0.020
ClkTyp: 6 BufSiz :   300 PstEvt : 10 CurPset :   1 CPUcor:   0.000
Unit  :56 Buf_rng:  7911 MinTrg :  2 Min/Rng :   5 Minpip:     811
```

Station : Network name
GPS Garmin : type of clock
GPS : Percent reception of GPS
PPS : Counter of PPS. Indicator of PPS
Last update : Last time CPU clock was updated
Buffer time : Time put into last buffer. At the end of the string:
SET: indicator to inform that internal clock is updated
Rec time : correct time strings received since startup.
rng_file : current ringbuffer number
detflg : network trigger flag. 0->no trigger 1->trigger mode
recrd : recording flag 0->no recording 1->recording mode
ipoint : pointer in event queue
queue : number of events in queue. 0-> no events pending
ndet : current number of channels triggering
ndetw : number of channels with trigger within array propagation window
preblk : counting buffers from first trigger
postbk : counting buffers from end of trigger
To CATLOG : number of buffers left to write for an event
Last DPX : last detection number
iblk : current buffer number in disk ringbuffer
errno : current error number. Not yet supported
duration : current duration in number of buffers (seconds) of an active trigger
sta : current STA of channels used in detector (counts)
lta : current LTA of channels used in detector (counts)

chaflg : channels used by the detection program. Active channels are marked with a ‘1’ and non-active with a ‘0’. When a channel is triggering it is marked in inverse video on the screen.
Samples : first 32 samples in a buffer, multiplexed data.
ParSet : number of active parameter sets
Chan : number of channels to be digitized
Clktyp : type of clock
Unit : digitizing unit
Ext/Int : external/internal sampling frequency
MaxRec : maximum number of seconds to record
BufSiz : number of samples per buffer
Buf_rng : total number of buffers per file in ringbuffer
ArPrWnd : number of seconds in array propagation window

PreEvt : number of seconds in pre-event memory
PstEvt : number of seconds in post-event memory
MinTrg : minimum number of triggers among flagged channels to initiate a recording
Min_rng : total number of minutes of data per file in ringbuffer
SmpBufCh : number of samples per buffer per channel
CurPset : current parameter set
TimCor : for debugging
S-Rate : sample interval in seconds
CPUcor : for debugging

DPL: This program plots a specified seismic channel in "real time" as new buffers are filled. There are several options for scaling both time and amplitude.

FPL: This program plots a specified seismic channel in "real time" as new buffers are filled. The program display the original channel data and the filtered channel data.

RTP: This program plots any selection of the channels that are flagged for detection. **(Only among channels flagged for detection)** The data are plotted in real time. There is a limitation on how many channels can be plotted, depending on the data throughput (number of channels, sample rate etc.). In cases where a lot of channels are be plotted, some data will be lost, but the picture will give a reasonable impression of the data. The reason for this is that the data-acquisition and event recording always will have higher priority.

LGS: The detection information in */home/buffer/event* may be formatted and written out on your terminal using LGS. The user is prompted to input a number as follows:
Negative: number of n previous detections
Positive: number of n hours back in time
Below is a typical output from LGS:

Example:
LGS

```
Station name: WNSN
Station code: WNS  Computer node name: WNS
Last detection number was: 5

How many detections counted from the last (negative no)
or number of hours from now (positive no)...-5
Current time: 13:33:56.354 13/03/91 072

 dpx  y doy  date     time dur   sta    lta   blk                     filename
 1 91 67  3 8  15:25:32.19   9    13      7   131 1991_03_08_1524_32T.WNS_16_1
 2 91 70  3 11 21:11: 3.95  70    30      8   717 1991_03_11_2110_03T.WNS_16_1
 3 91 71  3 12  6:39:10.87  60    -1     -1   504 1991_03_12_0638_10T.WNS_16_X
 4 91 71  3 12 11:34:32.15  57    10      9   112 1991_03_12_1133_32T.WNS_16_M
 5 91 72  3 13 13:20:36.87  16    85      9   468 1991_03_13_1319_36T.WNS_16_2
```

Explanation of abbreviations for LGS:

dpx	:	detection number
y	:	year
doy	:	day of year
date	:	month and day
time	:	hh.mm.sec.msec
dur	:	duration of event in seconds
sta	:	average maximum STA of all channels used by detection program
lta	:	average LTA of all channels used by detection program at detection time
blk	:	disk record at detection time
filename	:	filename of catalogued event

PLT:

The program can plot up to 32 channels. The main purpose of the program is to plot data and it does not have the same picking facilities as the MULPLT program in SEISAN. The program is started up by typing *plt*, below is a sample session.

```
plt
 Filename, DPX #, or -nn last events, plot ringbuffer press ‘r’: -5
 Current DPX:53
 Dpx=    49                                  Duration of detection(secs)=    2
 Dpx=    50                                  Duration of detection(secs)=    2
 Dpx=    51                                  Duration of detection(secs)=    2
 Dpx=    52    1992_04_04_2055_42T.JMI_08_X   Duration of detection(secs)=    8
 Dpx=    53    1992_04_04_2303_36T.JMI_08_2   Duration of detection(secs)=   14
```

Lines that contain no filenames, indicate that there have been a detection, but that the duration was too short to record it as an event according to the parameter setting for the trigger algorithm.

```
 Filename, DPX # or -nn last events, plot ringbuffer press ‘r’: 53
 File: /home/events/1992_04_04_2303_36T.JMI_08 FOUND !!!
 Total duration of recording (seconds):     76.0
 Defaults, press RETURN. Options press O
 Available channels
   1  JMI S  Z     2  JMI   S  N      3  JMI S  E    4  JNW S  Z    5  JNE S  Z
   6  JMI SL Z    7  TIME PRE    8  TIME MAI

                               Enter channels to plot:
Enter channel numbers on one line with space in berween, finish with RETURN
To use default values, just press RETURN.
1 4 5
Auto-scaling, press RETURN. Options press O
PRESS RETURN !!!
Start/Stop time, press RETURN, Options press O
PRESS RETURN
```

The plot of the selected channels will now appear on the screen.

On the top of the screen is a menu of options for some simple processing of the event:

```
Time and date   Options: R Replot       F,V Filter    N/B Next/back event
                         Q Quit         K   Spectrum  L   Locate
Event number: 53         O Other event  MLB, Sp Zoom  P,S,C Phases
```

R	replot current event
Q	quit plt program
O	plot other event
F,V	F: select a filter and plot event. V: Select a filter and plot the filtered data on top of unfiltered data
K	generate specter of data. Works only when only one channel is selected.
MLB, Sp	Mouse Left Button or ‘space’ can be used to select area for zoom.
	Position the cursor at a the leftmost point of your window, press left-button on mouse (or ‘space’). Then move the cursor to the rightmost point of your window, and press the left-button on the mouse (or ‘space’). Two vertical lines will indicate you window, zoom it and plot it. To get back to the original scale, select rightmost first, and then leftmost.
N/B	N: plot next event. B: plot previous event.
L	Locate event. First you have to pick the phases.
P,S,C	Pick phases. Position the cursor on the p-phase and press the ‘p’ on the keyboard to select the time for the p-phase. Use the same procedure for the s-phase and the coda for each channel.
t	This option do not appear in the menu, but can be used to plot the next ringbuffer when you have specified to plot a ringbuffer.

To get back to normal text mode on the screen, type ‘q’ for quit.

Event input can be either the file name, or dpx number. By typing e.g. -10, the last 10 detections are shown. If a return for all defaults is used, the questions following do not appear and the program will used default channels and the whole time window. The default channels are specified in file /home/seislog/pr/pic.def. If e.g. channel 3,5 and 7 were chosen the file would look like

03
05
07

If no default file is available, all channels will be plotted. If manual selection is chosen, each channel number must be given separately as shown above

The distance between traces is regulated automatically. If 3 or less traces are selected, the scaling is so that each trace will occupy 1/3 of the screen in order to not blow the signal up too much.

The selection of time limits will be adjusted to coincide with buffer limits. So if e.g. the buffer size is 2 seconds, the start time will be adjusted to within 2 seconds of what has be specified by start time and end time. This also means that the smallest window which can be shown is one buffer.

On top of the plot one line of text is shown. The left-hand time string gives the time of the first sample on the plot, the following text shows the options, all activated by pressing one key. If a correct key is pressed, a short beep sounds, if key is not valid, a long beep sounds. All can be lower or upper case.

When one event has been plotted, another event can be selected or the program can be terminated with return.

VER Print out version number of SEISLOG

EPI: Epicenter location, very primitive

The program uses a simple halfspace with a constant velocity. The program iterates for location and origin time but not for depth. The depth can optionally be found by letting the program test a range of depths and selecting the one with the lowest rms.

How to run program:

The program uses an input file called /home/seislog/pr/*station*. Below is an example

```
7.0  3.5
5.0  30
2.0 0.003 -2.0
2.0 35.0 53.0 68.0
ODD1 59.912 6.628
BLS5 59.423 6.456
ASK  60.483 5.195
EGD  60.271 5.226
OSG  60.479 2.876
```

The first line gives the P and S-velocity respectively. The P velocity will in most cases be between 6 and 7 km/sec considering that this is an average velocity. The S-velocity should then be the P-velocity divided by 1.73. However, if Lg is a dominating secondary phase, the S- velocity should be set to 3.5 irrespective of the P-velocity.

The second line gives the initial depth and number of depths to test. If only one depth is tested that is also the depth being used, this means a fixed depth. This might often be advisable. However if good station coverage is available, test for a range of depths. If e.g. 20 depths are tested for, the program will start with the first depth and try 19 other depths increasing the depth 10 km each time and the last depth tested will be 195 km. The depth giving the smallest rms will be used.

The third line gives the coda magnitude parameters. Magnitude is calculated as

```
    mag = c1*log10(coda) + c2*distance + c3
```

coda is the coda length (seconds) from the ORIGIN time and not P as is the most usual. This means that compared to other coda magnitude scales, the distance correction term c2 should be reduced a bit.

The forth line give the maximum and minimum longitude and latitude for mapping.

The remaining lines give the station code and latitude and longitude, positive N and E. There should be no blank lines in the file and no carriage return after the last line. The first station is the list is used for coordinate system reference and should be near the seismic network center. It can be a dummy station.

Input of phases comes from a file redirected to standard input, the following is an example:

```
ODD1 40.9 1
ODD1 68.9 2
BLS5 38.9 1
BLS5 69.1 2
ASK  25.5 1
ASK  43.1 2
EGD  25.5 1
EGD  42.0 2
OSG  11.0 1
OSG  21.0 2
EGD  160 3
```

Each line contains the station code, the phase arrival time and an index giving the phase type. 1 is P, 2 is S and 3 is coda. Program PLT makes such a file when picking phases. If the name is phase, the program is run by typing **epi<phase**.

Output on the screen should be self-explanatory. In addition to the screen output, the epicenter and magnitude is written out to a file called epicenter. The MAP program to plot the epicenter uses this file.

MAP: Draw maps of selected areas defined in file /home/seislog/pr/*station.* At present only a limited number of maps are available.

4. COMMUNICATION

4.1 Data transfer to a host computer

A data acquisition system is critically dependent on the ease of transferring the data to a host computer. In SEISLOG, this can be done in several ways. Most systems use either the serial line or FTP via Ethernet. Using FTP is simply a case of copying the file to the host computer, while using the serial line most applications use a particular SEISLOG program listing the event file on the screen in compressed ASCII (CMP6) format (defined in the international GSE format), and the receiving end use a communication program to capture the file (see below for description on dial up network). KERMIT can also be used on the serial line to directly copy the binary file, however, this is more time consuming than the ASCI transfer. The two other methods used are streamer tape (e.g. DAT) or floppy, depending on amount of data. Whichever way the files are transferred, the same Fortran program is used on PC or Sun to read and demultiplex the event files.

4.2 Manual download of data from SEISLOG

The idea is that an event file is listed on the screen in printable ASCII characters and the user captures the data on the terminal connected. Most PC terminal emulation programs have an option to open a download file.
The first step is to login to SEISLOG and then open the logging file on the PC. Then give the command:

ASCC Then you will be asked:

-ASCII (A) or Compression (C): ASCII is plain number while compression is CMP6 characters.
Next question:

File (F) or Screen (S): **F**ile means that the data goes to a local file on the SEISLOG system called /home/seislog/os9.log,
screen means that the data is listed to the screen.

-Enter detection number, -1 to end: The detection number obtained when using command LGS. The program will come back to this question again after listing the event, so many events can be listed.

-Number of buffers to write out: xx: xx is the actual number of buffers to list, return to start listing.

The first line listed will be **START OF EVENT** **********************, which is an indication to the decoding program that a new event is starting. Then the header and then the data on the screen either as plain integers (10 per line) or in compressed printable ASCII format (CMP6, compressed). However the header is always in ASCII and in the same format as in the original file. The PC or Sun conversion program can use either type of data, but it is clearly much more efficient to use compressed CMP6 format. Note that the ASCC program is also useful to inspect the content on a binary event file. ASCC is not intended to be used on a regular basis, but is rather a tool for the occasional user logging into a station. For automated data transfer, use the SEISNET program, see next section.

4.3 Network of SEISLOG systems using SEISNET

With the new versions of SEISAN and SEISNET, data communication between central and field stations can now be accomplished using the SEISNET package. SEISNET supports several options for file transfer, logging of network status and preliminary localization. Documentation on SEISNET can be found on ftp site:
ftp.ifjf.uib.no

Introduction to SEISNET

The SEISNET software is made to combine seismic stations of various types with communication capabilities into a network. The main routines carried out by SEISNET are transfer of parametric data, network event detection, transfer of waveform data and automatic determination of epicenter location and magnitude. The data is stored in a central database, which is using the SEISAN system for storage and processing. Alternatively, the automatic waveform data transfer can be based on given hypocenter and origin time. An important design goal has been to build software that can support various types of acquisition systems and communication methods. SEISNET is meant for networks with data acquisition on remote systems and is therefore not working in real-time.

Overview of SEISNET features

The following list gives an overview of the main functions that are automated in SEISNET:

- Retrieval of detection information from seismic stations
- Retrieval of epicentral information provided by seismic centres
- Retrieval of waveform data from seismic stations
- Retrieval of waveform data using AutoDRM (Kradolfer, 1996)
- Network event detection
- Automatic phase identification, hypocenter location and magnitude determination
- Transfer of waveform data from selected field stations based on a given hypocenter location and origin time

System requirements for SEISNET

SEISNET is available on the Unix platforms SunOS, Solaris and Linux. The software is based on freely available software only. SEISAN has to be installed, since the two packages are closely integrated. SEISNET requires the Expect, Tcl and Tk software, which are available for the three Unix platforms. In addition, the C-Kermit software is required. Information on location and installation of all the packages is given in the manual.

The data acquisition systems supported at present (1998) are Quanterra, as e.g. used on the IRIS stations, SEISLOG (Utheim and Havskov, 1997, Natvik et al, 1999) for QNX and Windows. SEISNET also supports any kind of data acquisition system that creates event files and is accessible through ftp. The only requirement is that date and start time are given in the event file names. In addition, the software supports automatic data retrieval through AutoDRM. More systems will be added as the need arises. Methods of communication supported are TCP/IP (Internet and VSAT) and modem dial-up.

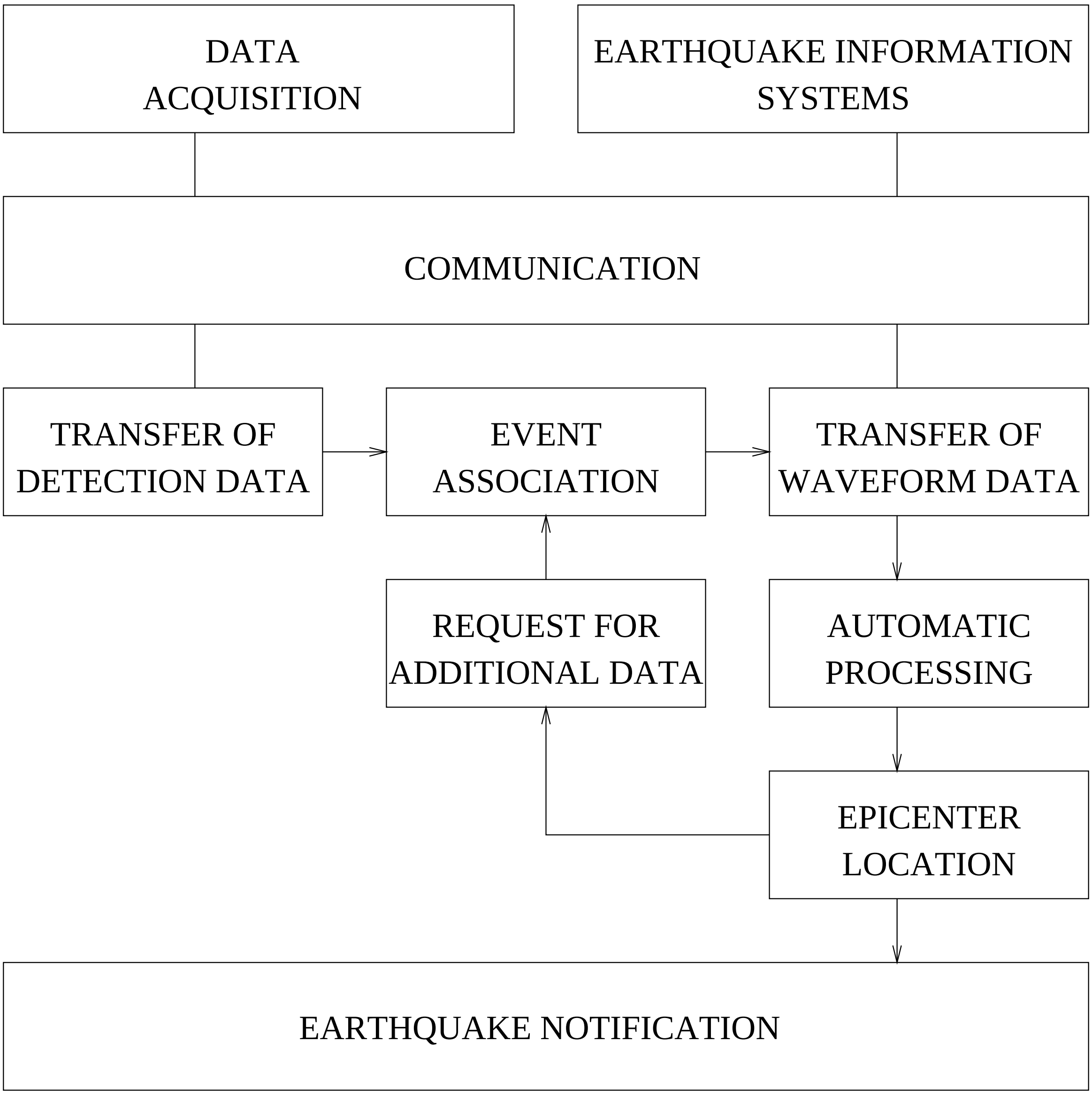
For parametric data retrieval SEISNET supports the 'finger quake@…' command (and format), which is commonly used to report earthquakes in near real-time, and SEISAN databases at other locations. In both cases TCP/IP is required.

Basic operation of SEISNET

In SEISNET, the data transfer is initiated and controlled by the central computer. This gives tight control on the operation and facilitates the support of various field systems.

The routine of downloading data can be divided into four stages. In a first stage, the system connects to the field stations sequentially and transfers parametric data on event detection. This information is stored on the central computer in individual databases. In the second stage, the databases are scanned for identification of network event detections, which will be put into a central database. At this point, based on the detection times from the stations, a first preliminary event location becomes available. The next stage is the transfer of waveform data for the network detections. In the case of sparse networks, the parameters for network detection can be set to transfer waveform data for all the detections in the network. In the last stage, automatic processing is carried out on the transferred waveform data for determination of event location and magnitude. Based on these parameters, waveform data from additional stations, which had not triggered, can be initiated.

Figure: Flow of data in SEISNET


DATA ACQUISITION
EARTHQUAKE INFORMATION SYSTEMS
COMMUNICATION
TRANSFER OF DETECTION DATA
EVENT ASSOCIATION
TRANSFER OF WAVEFORM DATA
REQUEST FOR ADDITIONAL DATA
AUTOMATIC PROCESSING
EPICENTER LOCATION
EARTHQUAKE NOTIFICATION

The SEISNET software can also be used to automatically collect waveform data for an event with given hypocenter and origin time. This works similar to the Incorporated Research for Seismology (IRIS) Spyder system (Malone, 1996).

The main philosophy in the design of SEISNET is to create flexible and modular software, which can support different acquisition systems, rather than a fixed solution. This has become essential due to the large variety of data acquisition and communication systems used. SEISNET has been developed through automation of existing software.

Integration of SEISAN and SEISNET

The SEISNET software makes use of the SEISAN database structure and the formats for parameter and waveform data. This means that the data, which are transferred by SEISNET, can interactively be accessed and processed using the SEISAN programs and no additional conversions or transfer to other computer platforms are required. The SEISAN trace-plotting program has built-in functionality to interactively check and group the incoming seismic traces. Based on event location and origin time, additional data from seismic stations can be requested, which will be automatically retrieved and added to the SEISAN database.

Setting up the software

The SEISNET software is mainly aimed at local and regional seismic networks with remote data acquisition systems in the field. The seismic network, within SEISNET, is defined in a single parameter file. There is practically no limitation on the number of stations, however, in the case of a large number of stations with slow communication, several SEISNET processes might have to run in parallel. A set of parameters has to be set for each station, for example the IP-address or telephone number, the type of station, the type of communication, login and password, etc. SEISNET is started at certain times using the Unix command cron and using short time intervals, the software can run continuously. For automatic retrieval of data through AutoDRM, a continuous process checks for incoming email.

In addition to the station parameters, a number of general parameters have to be set. These are for example path definitions, the name of the central database, parameters on logging, the time interval for data transfer and processing respectively and the number of retrials in case of errors. A few parameters define the network event detection, like number of minimum triggers in the network and length of the array propagation window. The automatic processing is specified through parameters as desired. A detailed description of all parameters is given in the software manual.

Examples of networks

The flexibility of the SEISNET software allows for many different configurations of seismic networks. In this section, a few examples will be given for illustration:

i. A national seismic network consists of 12 digital broadband systems with the Quanterra data acquisition and retrieval system. Half of the stations are connected to the central recording site through VSAT using TCP/IP protocol, while the other half is accessed through dial-up telephone lines. The SEISNET software is used to download detection information from all the stations with a given time interval. Network events are declared if two of the stations have triggered within the array propagation window. In the last stage of the automatic routine, waveform data is transferred for events detected in the network.

ii. A regional network of 20 IRIS/GSN stations in different countries is defined and accessed through SEISNET. The goal is regional event detection and waveform data transfer in near real-time. The central site can be anywhere within or outside the region. Communication is done through Internet and dial-up. A minimum of five triggers within the array propagation window is required for event detection. For selected events, based on location and origin time, additional waveform data is transferred using AutoDRM.

iii. A regional network consists of a few IRIS/GSN stations and several national networks, in which data is transferred through radio telemetry and recorded at a central site on the SEISLOG data acquisition system. The regional center is connected to the IRIS/GSN stations and national recording sites through Internet and dial-up connection. The goal of the center is to locate regional earthquakes in near real-time. SEISNET will transfer automatic phase picks from the national networks, then run a network event detection and based on this determine a preliminary epicenter location and magnitude. Then the waveform data transfer is initiated.

iv. In a regional center, data is transferred automatically from the regional IRIS/GSN stations for teleseismic events that are reported by the National Earthquake Information Center (NEIC) in near real-time.

In the four examples, the interactive processing using the SEISAN package can be started, as soon as the data is transferred. The goal in all cases is fast determination of epicenter and magnitude. Since the SEISNET software is meant for data acquisition using field stations without data transmission in real-time, there is a limitation on the response time of the system.

Adding new station types to SEISNET

The software is designed so that new station types and transfer protocols can be added. In general there is no limitation on the type of systems that can be integrated. New systems are integrated by including subroutines to handle the respective system. The SEISNET software will then automatically find the correct routines, based on the entry made for the new type of station in the parameter file. As mentioned under ‘System requirements’ any system that creates event files and is connected through ftp can be integrated without modifications.

References:

Kradolfer, U. (1996). AutoDRM-The First Five Years, Seismological Research Letters, 67,4.

Malone, S. (1996). Electronic Seismologist: SPYDERS on the INTERNET, Seismological Research Letters, 67, 5.

Ottemoeller, L. (1998). SEISNET user manual, Univeristy of Bergen.

4.4 Setup SEISLOG in local network

If you want to include your SEISLOG system as a node in your local Ethernet network, you have to install the TCP/IP distribution. You need a legal IP address and a legal node name. Contact your system manager.

After you have installed the TCP/IP distribution you have to edit some system files:

First log in as superuser

- Change directory **cd /etc**
- **vedit hosts** and add a line with the IP address and the node name like the line below:

  ```
  111.222.333.444    seislog1      seislog1
  ```
- exit editor
- Change directory **cd /etc/config**
- **vedit sysinit.1** and add the following lines:

  ```
  Net &
  Net.ether1000 -l 1 -p 240 -I 10 &
  ```

  Where Net.ether1000 is the driver for your specific ethernet board, -l 1 is logical network number (default=1), -p 240 the address space for your specific board and -i 10 the interrupt number for your board. The driver name, address space and interrupt number will vary from one board to another
- Add a line **Socket seislog1** where seislog1 is the same name you specified in /etc/hosts.
- Add a line **/etc/ifconfig en1 seislog1 up** where seislog1 is the name you specified in /etc/hosts
- Add a line **/etc/inetd &**
- Exit the editor and reboot the system

You should now be able to connect to the SEISLOG system from another machine on the network with telnet 111.222.333.444

4.5 Setup SEISLOG as an Internet node

If you want to include your SEISLOG system as a node in your local Ethernet network, you have to install the TCP/IP distribution. You need a legal IP address and a legal node name. You also need to know the name server domain and name server IP address. You need the netmask number and also the gateway and broadcast addresses. Contact your system manager.

After you have installed the TCP/IP distribution you have to edit some system files:

First log in as superuser

- Change directory **cd /etc**
- **vedit hosts** and add a line with the IP address and the node name like the line below:

  111.222.333.444   seislog1   seislog1

- exit editor
- vedit resolv.conf . The resolv.conf file should look like this

  domain abc.def.gh

  nameserver 111.222.333.1

  where abc.def.gh is the domain and 111.222.333.1 the IP address for the nameserver

- exit editor
- Change directory cd /etc/config
- **vedit sysinit.1** and add the following lines:

  Net &

  Net.ether1000 -l 1 -p 240 -I 10 &

  Where Net.ether1000 is the driver for your specific ethernet board, -l 1 is logical network number (default=1), -p 240 the address space for your specific board and -i 10 the interrupt number for your board. The driver name, address space and interrupt number will vary from one board to another

- Add a line **Socket seislog1** where seislog1 is the same name you specified in /etc/hosts.
- Add a line **/etc/ifconfig en1 seislog1 netmask 255.255.255.0 broadcast 111.222.333.456 up** where seislog1 is the name you specified in /etc/hosts. 255.255.255.0 is the netmask and 111.222.333.456 is the broadcast address
- Add a line **/etc/inetd &**

- Add a line **/etc/route add default 111.222.333.1** where 111.222.333.1 is the gateway address.

- Exit the editor and reboot the system

You should now be able to connect to the SEISLOG system from another machine on the network with telnet 111.222.333.444

5. SOFTWARE TECHNICAL DESCRIPTION

In this chapter we will discuss how the different programs in SEISLOG interacts, how memory is organized as a shared resource between processes

5.1 Principles of operation

The real-time software contains several programs that communicate through a shared memory area. This module is divided into four main areas:

- Parameter area.
- Data variables.
- Data buffers.
- Memory ringbuffer to hold N sequential copies of data buffers.

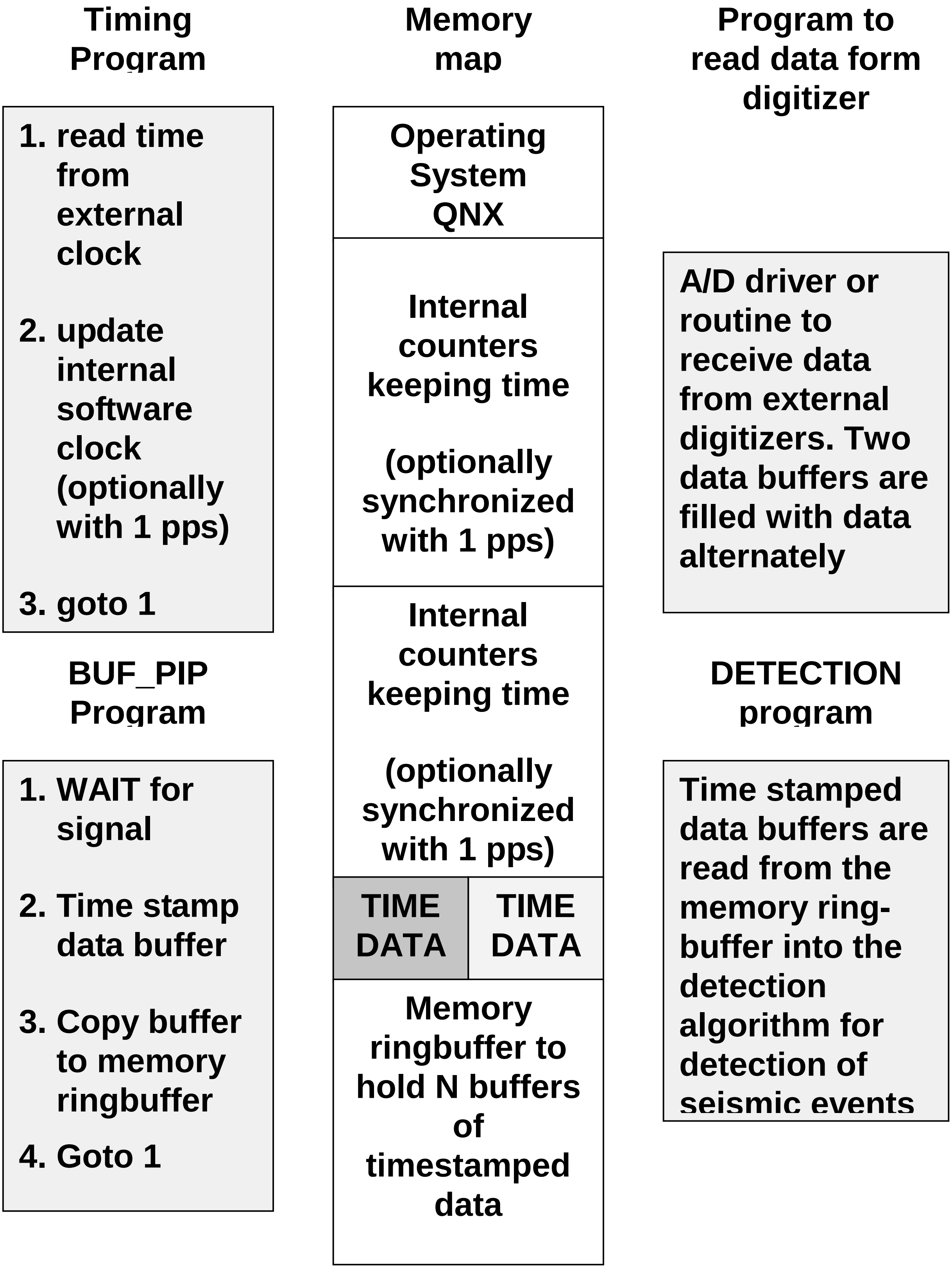
Figure 5.1 outlines a schematic of the memory map and how some basic programs control the timing system and data flow through memory.
During startup of SEISLOG the parameter files are read and the parameter area holds a copy of the parameters for fast access from active processes.
The program to synchronize the internal software clock, on figure 5.1 called TIMING, is the first program to be executed after the parameters have been read. This program waits for the time string to arrive on the RS232 line connected to the clock. When a valid time-string is received, the program will enable an interrupt on the parallel port and wait for the 1 pulse per second from the clock. The internal software clock is then synchronized and the interrupt disabled.
BUF1 and BUF2 are filled with one second of data from the A/D driver program or from the program supporting Nanometrics, Earth Data or Terra Tech.
Every time one buffer is full, the program BUF_PIP receives a signal from the A/D driver (or from the program supporting Nanometrics, Earth Data or Terra Tech digitizers) telling that a buffer is ready. The data from BUF1 and BUF2 are time stamped and copied onto the memory ring buffer.
The ring buffer concept used here can shortly be described as a chunk of memory where a program can fill in data on the top, and another program can read out data from the bottom. The advantage using our own pipe concept is that we may access data anywhere in this “pipe” using indexes.

The advantage of using this memory ringbuffer is that we may fill it at a higher speed than the speed we read data out. This scheme works fine as long as the memory ringbuffer has a size that is much bigger than the size of the buffers we are filling into it. The memory ringbuffer may buffer up time stamped buffers for a while when the system is busy doing other things (cataloging events or heavy communication). Later the system can catch up with the input rate and the amount of data buffers in the pipe will be reduced.

The next step in the process is that the detection program, DET_EVT, reads time stamped data from the memory pipe and starts processing the data for seismic events.

Fig.5.1

Principle of operation



Timing Program

1. read time from external clock
2. update internal software clock (optionally with 1 pps)
3. goto 1

BUF_PIP Program

1. WAIT for signal
2. Time stamp data buffer
3. Copy buffer to memory ringbuffer
4. Goto 1

Memory map

Operating System QNX

Internal counters keeping time

(optionally synchronized with 1 pps)

Internal counters keeping time

(optionally synchronized with 1 pps)

TIME DATA

TIME DATA

Memory ringbuffer to hold N buffers of timestamped data

Program to read data form digitizer

A/D driver or routine to receive data from external digitizers. Two data buffers are filled with data alternately

DETECTION program

Time stamped data buffers are read from the memory ring-buffer into the detection algorithm for detection of seismic events

5.2 Program Interaction

Figure 5.2, 5.3 and 5.4 illustrates the interactions between the main programs and the flow of data. Data from the a/d-driver or from the digitizer programs are stored in shared memory in BUF1 and BUF2 depending on the buffer index. First BUF1 is filled up with data and then BUF2. Immediately after one buffer is filled up, a signal is sent to the program BUF_PIP. From the buffer index this program knows which buffer to process. The buffer is time stamped and copied to the memory pipe. Some counters and switches are updated, and BUF_PIP waits for the next buffer to be ready (next signal).
The time is read from the software clock, which is synchronized by an external clock.

The memory ringbuffer system has the advantage that the only time-critical point in the system is to get data out of the pipe before it is overwritten. The memory ringbuffer has a flexible size up to a certain limit, and will cope with some delays in the system during a high load caused by detected events or communication.

The next step in the data flow is illustrated in figure 5.3.
The detection program DET_EVT checks the memory pipe for new buffers to process. If new buffers are available, one at a time is copied into buffer B1 or B2 depending on the buffer index. Data entering B1 or B2 are converted from integer to floating point numbers for filtering.
After filtering of a buffer has finished, a signal is sent to tell other utilities that they may look at the data. Original unfiltered data buffers are written to the ring buffer, and buffer B1 or B2 enter the detection algorithm. When a seismic event occurs, the event-queue is incremented, and a signal is sent to CAT_EVT to save the event.
Figure 5.4 shows how a seismic event is saved by CAT_EVT.
When CAT_EVT first is started, it checks the event-queue for events and enters a wait state if the queue is empty. If there are events in the queue, CAT_EVT will write a detection record in a small circle buffer (/home/buffer/event) for each event. This file will typically contain up to 1000 records before it is overwritten. The file may be processed by a utility to obtain information about seismic events over the last few days, or hours, depending on the seismic activity (LGS).
Next CAT_EVT checks if the queue is full. This is a situation that should normally not happen unless the seismicity reaches a very high level, and when the system has been set up with a very demanding parameter set.
Data in the ring buffer, that are declared as events, will then be overwritten before CAT_EVT manages to write the event to a separate file. In this case only the detection record is written to keep track of number of events and the time when they occur.

Normally the queue is not full and CAT_EVT will also record the complete event onto a separate file. These event files may then be processed by your own programs or by some of the supplied routines. Normally they will be transferred to a host computer for further processing and archiving.

Fig 5.2

Program Interaction

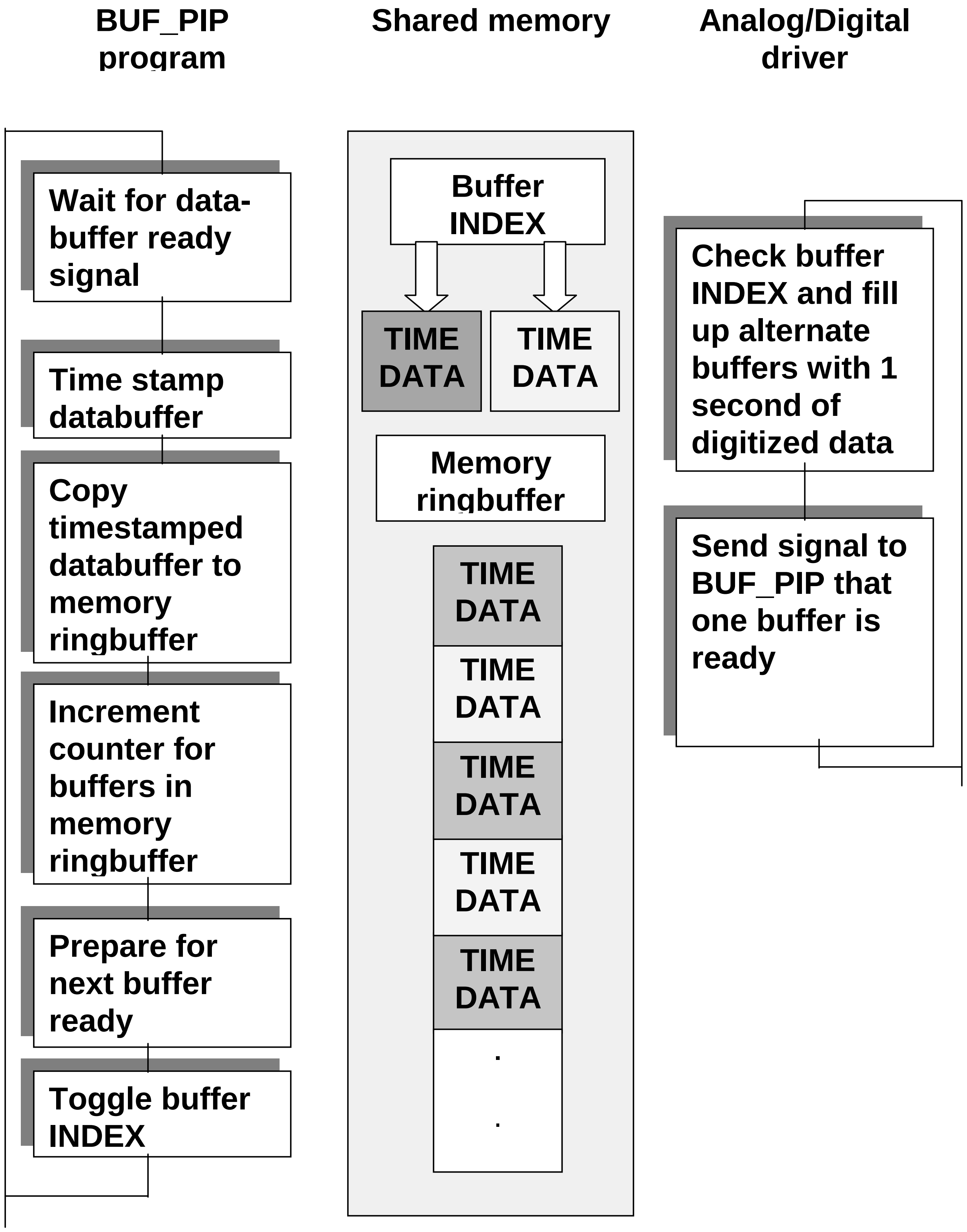

BUF_PIP program
Shared memory
Analog/Digital driver
Wait for data-buffer ready signal
Time stamp databuffer
Copy timestamped databuffer to memory ringbuffer
Increment counter for buffers in memory ringbuffer
Prepare for next buffer ready
Toggle buffer INDEX
Buffer INDEX
TIME DATA
TIME DATA
Memory ringbuffer
TIME DATA
TIME DATA
TIME DATA
TIME DATA
TIME DATA
.
.
Check buffer INDEX and fill up alternate buffers with 1 second of digitized data
Send signal to BUF_PIP that one buffer is ready

Fig 5.3

Program Interaction

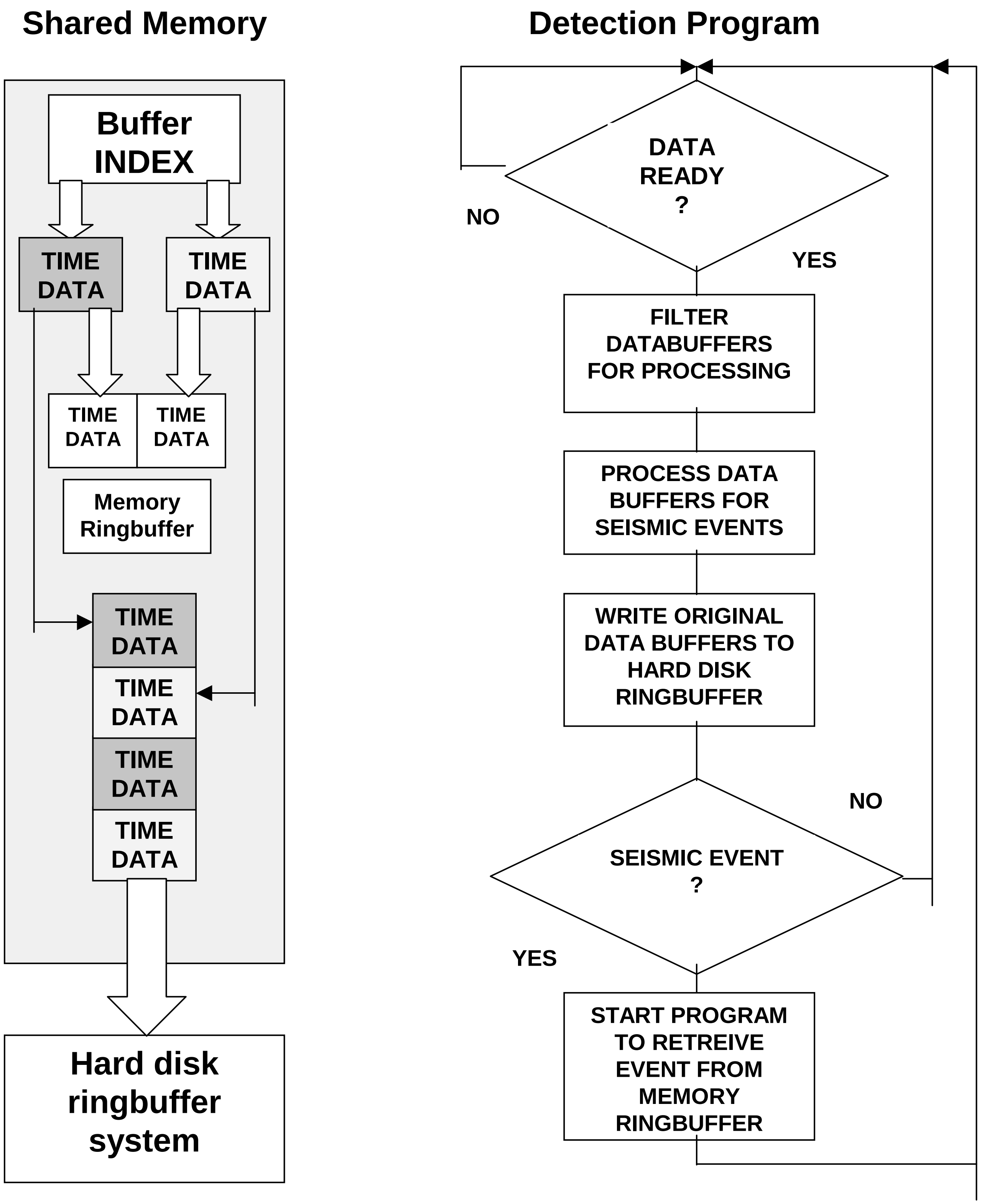
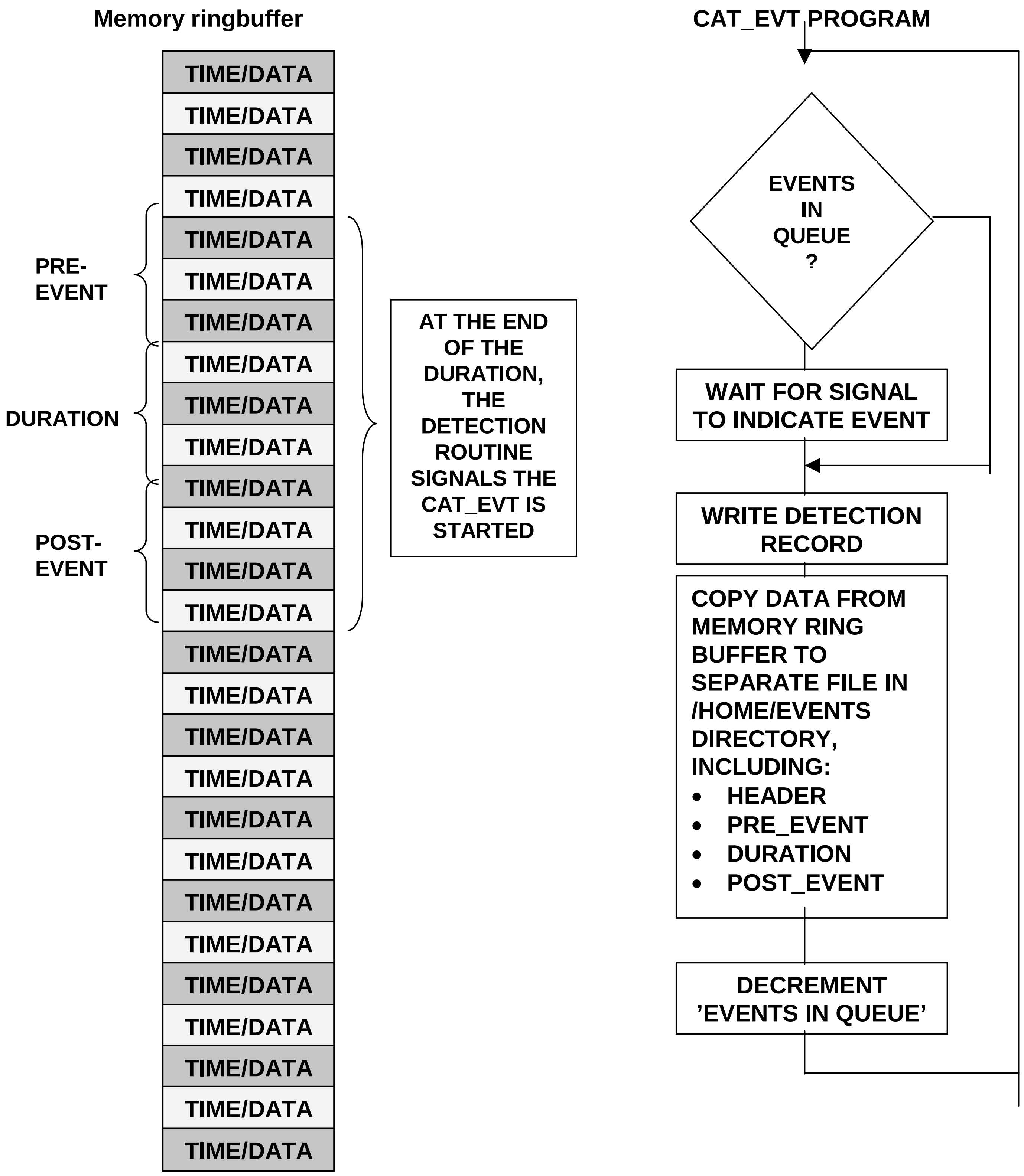

Shared Memory
Detection Program
Buffer INDEX
TIME DATA
TIME DATA
TIME DATA
TIME DATA
Memory Ringbuffer
TIME DATA
TIME DATA
TIME DATA
TIME DATA
Hard disk ringbuffer system
DATA READY ?
NO
YES
FILTER DATABUFFERS FOR PROCESSING
PROCESS DATA BUFFERS FOR SEISMIC EVENTS
WRITE ORIGINAL DATA BUFFERS TO HARD DISK RINGBUFFER
SEISMIC EVENT ?
NO
YES
START PROGRAM TO RETREIVE EVENT FROM MEMORY RINGBUFFER

Fig 5.4

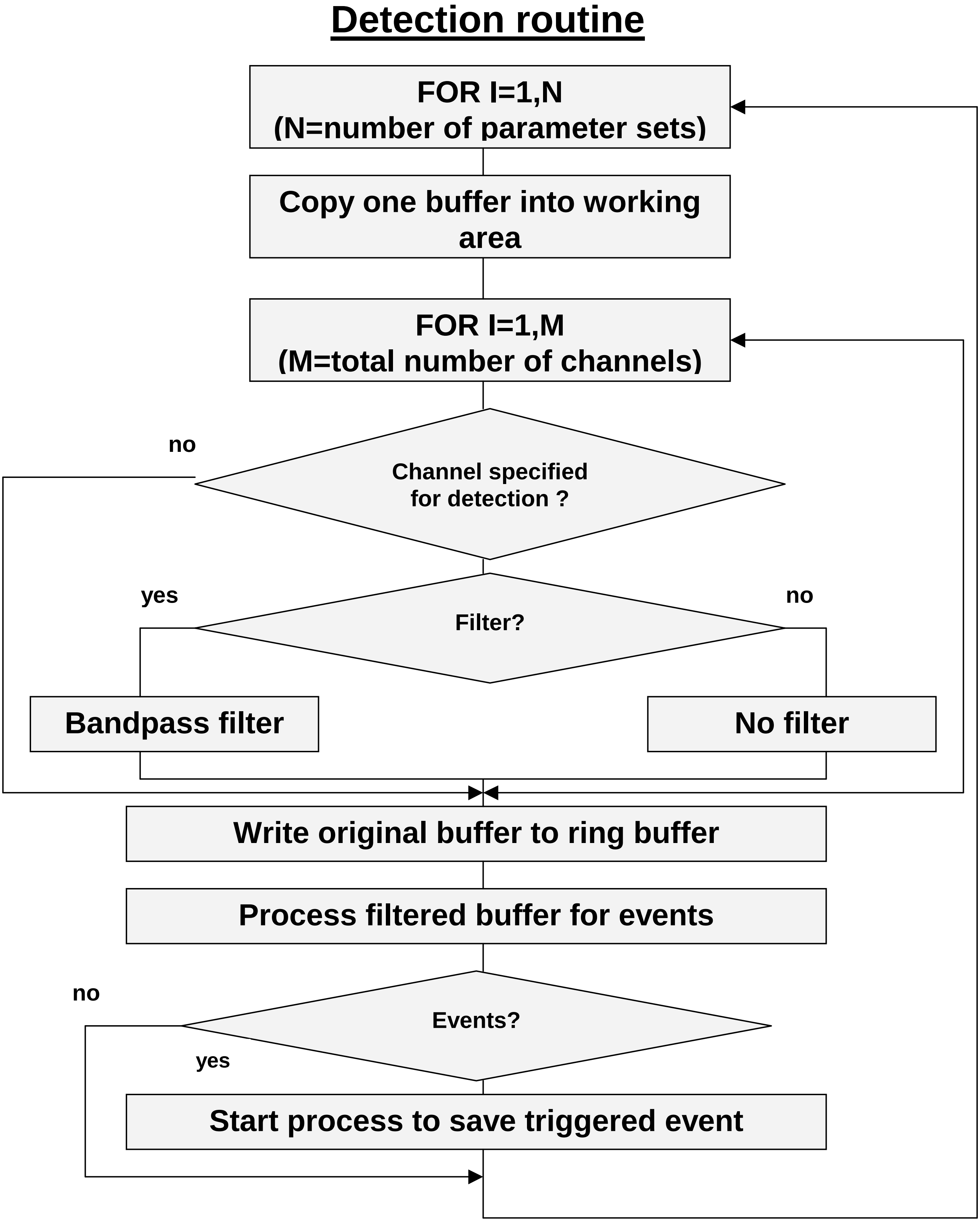
Program Interaction



5.3 The detection algorithm

The basic trigger criteria in the detection algorithm are based on the ratio between the short-term average (STA) and the long-term average (LTA) of the incoming data.
In SEISLOG you may specify multiple parameter sets. This way it is possible to use different trigger criteria on different selections of channels, for example between SP and LP data. The channels selected for detection processing in one parameter set may be filtered with different filters. Another parameter set may use other channels and other filters. The trigger and de-trigger levels may be different on different channels within one parameter set, but may also differ from one parameter set to another.
One parameter selects the minimum number of triggers among selected channels to start recording of an event. Post-event, pre-event and array propagation window (networks) may vary from one parameter set to another. Figure 5.5 gives an overview of the detection program.

Fig 5.5


Detection routine
FOR I=1,N
(N=number of parameter sets)
Copy one buffer into working area
FOR I=1,M
(M=total number of channels)
no
Channel specified for detection ?
yes
Filter?
no
Bandpass filter
No filter
Write original buffer to ring buffer
Process filtered buffer for events
no
Events?
yes
Start process to save triggered event

6. HARDWARE TECHNICAL DESCRIPTION

In this chapter we will describe in some more detail different hardware configurations, more details on how the timing system works.

6.1 Hardware configurations

The SEISLOG software supports a variety of options for external and internal digitizers and for different timing systems. In general any standard PC can be used to run SEISLOG. However, some criteria have to be followed, depending on the digitizer and timing system you want to use. If the system is going to support any kind of data communication, other criteria have to be considered. The following table gives an indicator of different possibilities.

	EXP. SLOTS	**COM1: /dev/ser1**	**COM2: /dev/ser2**	**COM3: /dev/ser3**	**LPT1 /dev/par1**
Internal multi-channel A/D	**1 or 2**				
External digitizer		•			
CPU clock only					
GPS 1 PPS			•		•
GPS No PPS			•		
Digitizer timing					
Radiocode MSC			•		•
24 bit ILI	•				
Modem comm.				•	
Ethernet	•				

6.2 Digitizers

This version of SEISLOG supports the following digitizers

2 types of analog/digital boards:

Unit **30**	**12** bit	**16+16**	channels	driver **dig_me2**
Unit **36**	**12** bit	**64**	channels	driver **dig_m18**

3 types of Nanometrics digitizers:

Unit **40**	**24** bit	**3**	channels	driver **dig_n24**
Unit **53**	auto gain	**3**	channels	driver **dig_nan**

Unit **56**	auto gain	**6**	channels	driver **dig_nan**

3 types of Earth Data digitizer

Unit **63**	**24** bit	**3**	channels	driver **dig_eda**
Unit **64**(new mod)	**24** bit	**3**	channels	driver **dig_edn**
Unit **99**	**24** bit	**n**	channels	driver **ili_seis**

1 type of Terra Tech digitizer

Unit **31**	**24** bit	**4(8)**	channels	driver **dig_ge1**

1 type of GeoSys digitizer,GBV

Unit **10**	**16** bit	**3**	channels	driver **dig_low**

2 types of R.O digitiser

Unit **38**	various res.	**1** or **3**	channels	driver **dig_ror**
Unit **70**	**24** bit	**3**	channels	driver **dig_324**

6.3 Setup Nanometrics digitizers RD3 &RD6

These digitizers use a built in sampling frequency and 1 second data buffers are transmitted to the SEISLOG via a RS232 serial line. The Nanometrics unit has a menu setup and the following parameters should be set to:

0. Baud rate	**: 9600**	
1. H/W sample rate (Hz)	**: 200**	Oversampling and
2. Decimation factor	**: 4**	decimation to 50 Hz
3. Channel scan rate (Hz)	**: 9600**	
4. XTAL frequency (Hz)	**: 19660800**	
5. XCLK frequency (Hz)	**: 0**	
6. Channels active	**: 3 or 6**	
7. First channel	**: 1**	
8. Output format type	**: 4**	
9. Gain ranging delay	**: 0**	
A. Sync time out	**: 0**	

To access this setup menu, read the Nanometrics manual.

6.4 Setup Nanometrics HRD24

This unit has a separate control port (RS232). The setup is menu driven. To work with SEISLOG the baud rate should be set to 19200, turn off the radio option (scrambler byte), sample rate 50 or 100 hz, number of bundles per packet must be set to 59.

6.5 Setup GeoSys/UiB GBV-116/316

This unit can be set up to work as both recorder and digitizer at the same time. To be used with SEISLOG as a digitizer, the AV setup program must be used to set up the serial output format defined in the GBV Operational Manual.

6.6 Setup Terra IDS-24 digitizer

The IDS-24 digitizer is set up with a separate program under MS-DOS. The IDSMenu program is part of the delivery from Terra Tech and documented in the IDS-24 manuals. Important parameters are: number of channels, type of format, number of bytes per sample and baud rate.

6.7 24 bit ILI

The EDL 2400 Telemetry Interface Unit, or 24-bit ILI, produced by Earth Data Ltd., may be used to receive data from digital modulators which may be connected directly, via modem, or via radio telemetry links. The ILI unit may be interfaced directly to a GPS clock, which is used to time tag the data internally.

In order to interface the ILI to a Seislog system, the Seislog PC should be fitted with a PIO-12 digital I/O interface board as supplied by Keithley Metrabyte, or with a corresponding product. This card utilizes an 8255 PPI (Programmable Peripheral Interface) integrated circuit.

The ILI is fitted with a number of interpolator modules, which receive the seismic data. N.B. Both single component and three component interpolator modules are available for use with the ILI, but they work at different frequencies. In order to work correctly with Seislog, **only three component modules should be used** with the ILI. Each module can handle three channels of seismic data, and may be used for a three-component set, or for one or more single components. The ILI can handle up to 16 interpolator modules, or 48 channels of seismic data.

The data from each interpolator module is collected by the control module and formatted into 1-second files for transfer to the PC. Each file contains a header section including time and status information, and a data section with 1 second of data from each channel.

The ILI is also equipped with a calibration module capable of providing a variety of signals, which can be switched to replace the incoming outstation signals.

Both 24 bit and 16 bit digital modulators have been found to work satisfactorily with the unit.

If the ILI has no GPS input, it will run, but will not sample at the correct rate, due to inaccuracies in its internal clock. In this case, the ILI will give the appearance of correct operation but will not provide accurate timing over an extended period.

The following section gives information on how to use the 24-bit ILI with Seislog. More information is available in the separate Instruction Manual for the 24-bit ILI, produced by the British Geological Survey.

6.7.1 Hardware and connections

In order to work correctly with Seislog, the switch and jumper settings on the PIO-12 card should be changed from the factory defaults. The Base Address switches should be set as follows:

Switch:	1	2	3	4	5	6	7	8
Setting:	OFF	ON	OFF	ON	ON	ON	ON	ON

This setting configures the card for an I/O base address of 0x280. The Interrupt Level Select should be set by means of a jumper on the card to 0x7. If these settings are not used, a change must be made to the Seislog source code. Note also that interrupt 0x7 is commonly used by the parallel port on a PC, and this will need to be disabled in order to use the above settings.

The main I/O connector of the PIO-12 card is a 37-pin D-type. This cable should connect the lower word of the ILI to port A of the 8255 chip and the upper word to port B. The ILI one-second interrupt is not used. The XREQ (request) line from the ILI is connected to the interrupt line on the PIO-12 card. The interrupt enable on the card is connected to bit 0 of port C (set this to 0 to enable IRQs). The ILI XACK (acknowledge) line is connected to bit 1 of port C. The ILI PDMA line is connected to bit 2 of port C (set this to 0).

It is best to connect the cable to the ILI AFTER the ILI and PC have been powered up; this avoids the possibility of damage being caused power being transmitted through the data lines.

A 9-way D-type socket is used to connect GPS RS-422 and 1 pulse per second signals to the ILI.

6.7.2 Configuration

The configuration of SEISLOG for ILI usage is achieved via the parameter file. The key parameters to set on page 1 are:

9.	Sample rate	0.0133

10.	Source Sampling Frequency	3
11.	Type of Clock	8
12.	Digitizing Unit	99

By setting parameter 11 (type of clock) to 8, direct time tagging via the ILI is enabled. This allows GPS input to the ILI to be used as basis for timing in SEISLOG.

On page 2, the individual SEISLOG channels should be configured. The configuration is controlled via the RC flag, which appears in the right hand column of the parameter file. Normally this flag is set to 1, but for an ILI configuration the RC flag is used to map input channels of the ILI to SEISLOG channels. The process works like this.

As many SEISLOG channels should be defined as there are input sources to the ILI. The channels are numbered sequentially. Each channel should have assigned as its RC flag a unique integer corresponding to the ILI input channel number, where:

- 1 is first interpolator module, vertical component;
- 2 is first interpolator module, N-S component;
- 3 is first interpolator module, E-W component;
- 4 is second interpolator module, vertical component;

and so on. It is important to remember that three channels exist for each interpolator module, even if all three channels are not used - i.e. RC flags 4, 5 and 6 correspond to the second module even if the first module only has a single component as input.

ILI inputs may be assigned to SEISLOG channels in any order. Conventionally, channels from related components are kept together, with the vertical component first, but more flexible arrangements are acceptable to the software if the user desires. However, in all circumstances you should avoid defining more SEISLOG channels than are actually present in the ILI.

The ILI device driver supplied with SEISLOG is called **ili_seis**. When the ILI driver starts up, it reads configuration information about the ILI from the SEISLOG parameter file, and uses this to determine how many channels are set up, the data length it expects, etc. It then reads a data stream from the ILI, and checks that the information in the header matches what it expects. If the data from the header indicates that the actual physical configuration is different from the parameter file, the driver makes a decision as to whether or not the configuration it has is valid.

For example, if more interpolator modules are present than have been configured for channels in the parameter file, this is allowable. The driver then adjusts its internal information to match the physical setup. This process is transparent to the user. Only after the expected vs. physical check is complete does it start to read data for storage by SEISLOG.

The driver will not run if:

- it detects a single component module fitted;
- More channels have been defined in SEISLOG than are available on the ILI.

In these cases, an appropriate error message will be issued to the user, and SEISLOG will stop.

6.8 Timing system

Normally a SEISLOG station is equipped with a high precision clock. This release of the software supports 3 different types of external clocks in addition to the internal CPU time.
The different clocks are specified in the parameter file as option 0-8:

OPTION	**TYPE OF CLOCK**
0	CPU clock
1	Garmin 35 1PPS support
2	Garmin 35 no 1 PPS support
4	Radiocode RSC8000
8	Timing by digitizer

6.9 How timing works with an external GPS clock with 1 PPS

The external clock transmits a time-string every second over a RS232 serial line. This time-string is controlled by the GPS receiver, and is correct to within microseconds when the GPS receiver is updated by satellite signals. The external clock also transmits a 1 one-pulse-per-second, 1pps, representing the same second sent over the RS232. The 1pps is correct in real time while the RS232 time-string suffers the delays which are normal over serial lines.
To achieve correct timing of the recorded data the following procedure is used.
In the timing programs a software clock is defined. This software clock contains two counters, one counting seconds and the other counting milliseconds. The millisecond counter is incremented by the CPUs timers. For every 1000 milliseconds, 1 second is added to the second counter. This scheme works with the accuracy of the CPU timer, which is normally not very good.
To synchronize the software clock with the GPS, the 1pps signal is connected to the parallel port of the computer. An interrupt routine detects this signal and the software clock can be synchronized every time this signal is received. In some cases the 1pps has to be modified to drive the interrupt on the parallel port. We have made a circuit that works with the Garmin-35 HVS. See figure after 6.10.
The software clock is first set to the current CPU time. If a GPS is connected to the serial port, the software clock is set to the time received from the GPS. This is done weather the time is synchronized by satellites or not, assuming that the time from the GPS is more accurate than the CPU clock at any time. This is of course not correct when the GPS not has been synchronized over a long period. The CPU time will be set to the time received by the GPS.
The software clock will not be updated by the GPS until the GPS is synchronized, and will run in ticks of 1 milliseconds controlled by the CPU. When the GPS becomes synchronized, it will transmit a 1 pulse per second with very high accuracy. This pulse is input on pin 10 on the parallel port. An interrupt routine will now adjust

the software clock to the exact second each time it receives the pulse. If the GPS again becomes un-synchronized, the 1 pulse per second will still be transmitted with a high degree of accuracy for at least 24 hours, and the software clock will be reasonably accurate for the same time. Each second buffer of data will have a flag telling if the GPS was synchronized at the time of recording.

If the pulse per second disappears, the software clock will be controlled by the accuracy of the tick rate of the CPU.

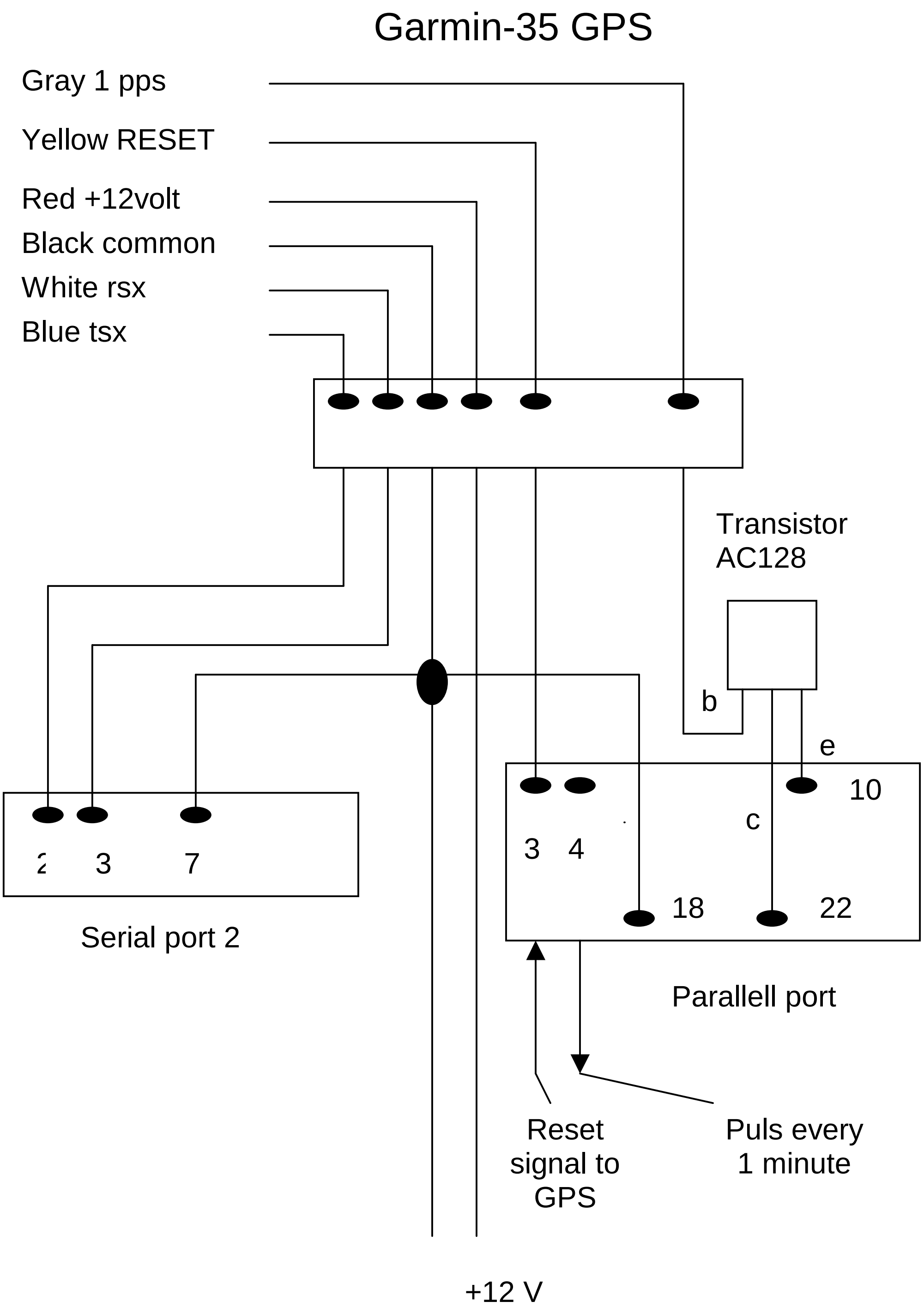
If the GPS stays un-synchronized for more than 24 hours, a reset signal is sent from the clock program to turn the power ‘off’ and ‘on’ on the GPS. Normally this will get the GPS in synchronized mode after a while. This can also be done manually with the program rst_gps.

A reset of the GPS is also done in the case that the clock program does not receive data from the serial port where the GPS is connected.

6.10 Minute pulse from parallel port

Parameter 19 can be set to a number greater than 0 if the user wants a 1-minute marker on the parallel port. The number indicates number of milliseconds duration of the pulse. 1000 will give a pulse of one second.

Synchronization circuit GPS



6.11 Check External Clock using the QTALK utility

Any output on a serial line can be monitored with a QNX utility called qtalk. To check f.ex. the output from a GPS clock, transmitting time strings, this program can be very useful. Normally when the clocks are turned on they start transmitting a character time string every second.
The serial port of the clock is always connected to the */dev/ser2* line on the SEISLOG system. To check that a time-string is transmitted from the clock your terminal console may be connected to the clock indirectly by using the utility qtalk.

Type: **qtalk -m /dev/ser2** and return.

A time-string should now be printed on your screen every second. This is a good indication that the timing system will synchronize the CPU time to give correct time in your data.
To get out of this check:
Type **CNTRL a** answer **q** to quit and return.

Each type of GPS clock normally has a setup program running under DOS or Windows. This program can also be used to get more detailed information of the status of the receiver. It should also be used for setup.

6.12 TROUBLE SHOOTING

Some combinations of parameters may corrupt the SEISLOG system. It may hang or start to work very slowly. There may of course be hardware errors causing some of the problems as well. In the next sections are listed some possible reasons for strange behavior of the system.

A good suggestion will always be to turn power off and on again.

System seems to go through the normal startup procedure but then enter a HALT state. When starting MON, it shows no activity.

For configurations with external digitizers:

Digitizer not connected to line */dev/ser1*.

No data transmitted from the external digitizer.

Check line */dev/ser1* with *qtalk -m /dev/ser1* to see if data is coming through.

Serial line could be damaged.

System seems to go through part of startup procedure and then enters a deadlock status. No response from console at all.

Check that the digitizing unit you have specified is the one in your configuration.

Check that the clock-type you have specified is the one in your configuration.

System never triggers. Not even on well-defined events.

Check that the parameter Minimum Number of Triggers is less or equal than the number of channels flagged for detection.

Trigger-ratio may be set to too high value.

Plotting real-time data using dpl or rtp. Program exits with an error message.

Check the sysinit.1 file that the following two lines are included:

```
int10 &
export WINFONTS=/home/seislog/pr/fonts
```

If not included, these commands can be tested directly from the console if you are logged in as superuser. Then try the dpl or rtp once more. If the plotting now works, you may edit the sysinit.1 file and include the two lines.

Plotting events with the plt program. Program exits with an error message.
or
All connections right, but data acquisition do not run.

This is probably because the mouse is not defined properly.
During installation of the operating system you are asked if you want to use a mouse on the system. If you answer yes on this question, the installation procedure will include a command **mousetrap start** in your sysinit.1 file. This command will search for a mouse connected to your system. If you have a ps2 mouse connected to a separate ps2 port, the startup file normally manage to find this mouse and you do not get any problems.
In the case you have a mouse connected to one of the serial ports, and at the same time have another serial device (clock or digitizer) connected to one of the serial ports, the **mousetrap start** command may get confused and try to define the clock or digitizer as the mouse. The following may happen:
The digitizer do not receive data, because the port is allocated by what the system thinks is a mouse. The clock do not receive data, because the port is allocated by what the system thinks is a mouse.
To at least get the data acquisition running, edit your sysinit.1 file and comment out the command **mousetrap start** and reboot the system.

7. Overview programs and utilities in SEISLOG

ACTIVE
Checks if SEISLOG is running or not. Output:
SEISLOG STOPPED or SEISLOG RUNNING
From console or terminal.

ACT_LOG
Writes timestamp to /home/error/up1 or /home/error/up2 at one minute intervals. On a restart (power-up or SRT) a message is written to the file. To check for stops or hangs, these files can be listed and checked.
Used by CAT_EVT

ASCC
Convert event-files to ASCII for file-transfer.
This program reads a specified seismic event, and converts the file into ASCII printable characters (See chapter 4 Communication). You also have the choice to compress the data and also to save the file on disk.

AUTO
Picks phases of recorded events. Not fully tested.
Started by CAT_EVT if parameter 26 is set to 1.
Can be run manually from console.

AUTO_ASC
Convert all event-files to ASCII for file-transfer.
Can be run from console or terminal.

AUTOASC
Convert event files to ASCII, BGS version
Can be run from console or terminal.

BCK_SRC
Make backup of updates to /home/src.
Can be run from console or terminal.

BUF_PIP
Start specified input driver, get ready buffer and time stamp data. Copy buffer to pipe.
Started by SRT.

C
Compile SEISLOG software. Without any argument this utility will compile all SEISLOG programs. With an argument for example *c srt* it will compile the program *srt* only.
Can be run from console or terminal.

CAT_EVT
Detected events are catalogued in the */home/events* directory
Started by SRT.

CAT_EXT
Extracted events are catalogued in the */home/events* directory
Started by SRT.

CHB
Check buffers in ringbuffer. Debug program.
Can be run from console or terminal.

CHK_PAR
Check parameters.

Used by SRT or can be started from console or terminal.

CHK_PPS Check if PPS is present. Debug only.
Can be started from console or terminal.

CHK_SNC Check if sync characters are present from digitizer. Debug only.
Can be started from console or terminal.

CHK_TAG Check IDX_LOG file. Debug only.
Can be started from console or terminal.

CLK_CPU Timing program using standard CPU time only.

CLK_GNP Timing program GARMIN GPS receiver. No 1 PPS.
Started by SRT.

CLK_GPP Timing program GARMIN GPPS. 1 PPS.
Started by SRT

CLK_MS1 Timing program RCC 8000A Radiocode clock BGS support

CLK_ONN Turn on printing of time info to console or terminal. Not supported.
Can be started from console or terminal.

CLK_OFF Turn off printing of time info to console or terminal. Not supported.
Can be started from console or terminal.

CLN_LNK Remove existing link in ringbuffer system before making new link.

CLS Clear screen

CNV_MAP Swaps x,y coordinates in map files. Debug only.
Can be started from console or terminal.

COPY_DOS Copies events from QNX partition to DOS partition, and converts filenames to 8 characters.
Can be started from console or terminal.

CPU Monitor CPU up-time. If CPU-monitoring has been activated in the startup file, this utility will list out information on when the system was started and for how long it has been active. NB! This does NOT say anything about if SEISLOG has been active or not. (See PWR). Not supported.
Can be started from console or terminal.

CPU_ACTI Program to log the time every 10-second. One record is written at startup and record number 2 is written every 10 second. Not supported.
Started by SRT.

CPU_DRF Program to measure drift factor of CPU clock. Not supported.

DATE_CHANGE Date handling routines for time-corrections within ILI 24 bit driver, BGS

DBG_ONN Turns on flag to print out debug information. Debug only.
Can be started from console or terminal.

DBG_OFF Turns off flag to stop print out debug information. Debug only.
Can be started from console or terminal.

DEC_MAT Program to do resampling of specified channels.
Started by DET_EVT.

DET_EVT Detection program.
Started by SRT.

DIG_GBV Program reading data from GBV digitizer.
Started by BUF_PIP

DIG_GE1 Program reading data from IDS-24 Terra Tech digitizer.
Started by BUF_PIP

DIG_HHH Program reading data from Nanometrics HRD24
Started by BUF_PIP

DIG_LOW Program reading data from GBV digitizer.
Started by BUF_PIP

DIG_ME2 A/D driver for Keithley Metrabyte DAS1202
Started by BUF_PIP

DIG_M18 A/D driver for Keithley Metrabyte DAS1802
Started by BUF_PIP

DIG_NAN Program reading data from Nanometrics auto gain unit, 3 and 6 channels.
Started by BUF_PIP

DIG_RAM Driver 12 channel digitizer (GeoSys)
Started by BUF_PIP

DIG_SYN Simulate digitizer.
Started by BUF_PIP

DIG_SER Reads data from shared memory that contains data from multiserial board, and create a standard SEISLOG buffer.

DIG_SER2 Reads data from shared memory that contains data from multiserial board, and create a standard SEISLOG buffer. IDS-24 based.
Started by BUF_PIP

DLY Force temporary time correction
Force a +/- delay on time. This utility may be used for testing purposes to find out what will be the correct value to use for TIME-CORRECTION in the parameter file. The program will ask for a floating-point value as input. If you enter a value of for example 0.062 it means that you actually adjust the time by 62 milliseconds. This value will be used until you set it to a new value or to the value specified in the parameter file when SEISLOG is restarted. You may monitor this value when running the MON utility.

DPL Plot specified channel in "real time"

DSK_ALL Allocate ringbuffer system.
Started by RE_INITI

DSK_INF Checks space on disk before allocating ringbuffer system. System use only.

EPI Epicenter location (very simple)

ERR_LOG Dump error log

EXT Extract data from ringbuffer system
Extract data from circle buffer. This program can be used in cases where the detection system has NOT recorded an event that should have been recorded. The ring buffer will contain continuous data from the last hours. How long time depend on the size of the ring buffer (parameter 15). Data may be extracted by block number or time. The extracted data will appear as a normal event, but the last character in the filename will be "X".

FPL Plot specified channel unfiltered and unfiltered in “near real time”.
Will only work if only one parameter set is active.

FRE This utility calls the system utility "df" which is documented in the QNX manuals. It writes out the capacity of the disk, how many sectors (512 bytes) are free and how many sectors are contiguous.

GEN_KOEF Generates filter coefficients based on the filter parameter directly.

HELP Prints out an overview of all programs and utilities.
Can be run from console or terminal

HEX Conversion program decimal=> hexadecimal, decimal=>binary.

ILI_SEIS Driver for the 24 bit ILI (Earth Data) See instructions in ILI manual. British Geological Survey.

ILI_24B.H Header file for ILI 24 bit driver, BGS

INT Interrupt routine for ILI 24 bit driver, BGS

LGS List event log
Can be run from console or terminal

LGS_RES Same as for LGS but for resampled data
Can be run from console or terminal

LNK_DOS Make DOS filenames as links to /home/events in /home/lnkdos

MAP Plots digitized map /home/seislog/pr/input.map
Started by EPI or from console

MDC Monitor DC levels
Started by console or terminal

MELDING Sends mail to address specified within program.
Started by CAT_EVT if specified by parameter file.

MEM Print out total and free memory.
Can be started by console or terminal

MIN_MAX Finds the time span in the ringbuffer system. Used by EXT

MIN_PLS Sends out minute pulse on parallel port

MKMAIL Generate e-mail text to be sent
Started by CAT_EVT

MON Monitor data acquisition system continuously

MON_PIP This utility is may be used to monitor time-strings from data-buffers written into the memory pipe. As for MON_CIRC you get

both the time-string for the buffer and the time-string for the last update from the external clock.
Can be started from console or terminal

Example:

MON_PIP

```
 0 05:43:01.906 01/08/1994 213     SET 05:43:00.619 01/08/1994 213
 1 05:43:02.906 01/08/1994 213     SET 05:43:01.619 01/08/1994 213
 2 05:43:03.906 01/08/1994 213     SET 05:43:02.619 01/08/1994 213
 3 05:43:04.906 01/08/1994 213     NSY 05:43:03.619 01/08/1994 213
 4 05:43:05.921 01/08/1994 213     NSY 05:43:04.619 01/08/1994 213
 5 05:43:06.906 01/08/1994 213     SET 05:43:05.619 01/08/1994 213
 6 05:43:07.906 01/08/1994 213     SET 05:43:06.619 01/08/1994 213
 7 05:43:08.906 01/08/1994 213     TMO 05:43:07.619 01/08/1994 213
 8 05:43:09.906 01/08/1994 213     TMO 05:43:08.619 01/08/1994 213
 9 05:43:10.906 01/08/1994 213     SET 05:43:09.619 01/08/1994 213
```

MON_SER This utility is may be used to monitor time-strings from data-buffers written into the multiserial memory pipe .
Can be started from console or terminal

Example:

MON_SER

```
 0 05:43:01.906 01/08/1994 213     SET
 1 05:43:02.906 01/08/1994 213     SET
 2 05:43:03.906 01/08/1994 213     SET
 3 05:43:04.906 01/08/1994 213     SET
 4 05:43:05.921 01/08/1994 213     SET
 5 05:43:06.906 01/08/1994 213     SET
 6 05:43:07.906 01/08/1994 213     SET
 7 05:43:08.906 01/08/1994 213     SET
 8 05:43:09.906 01/08/1994 213     SET
 9 05:43:10.906 01/08/1994 213     SET
```

MUL_SER2 Read 1 second buffers from multiserial board and place each buffer into its separate memory pipe.

OFF Turn manual trigger off

ONN Normally SEISLOG will only start the program CAT_EVT when an event has been detected. However it may often be convenient to record and catalogue data on command, e.g. when calibrating.
This utility turns on the recording flag. The next buffer being filled will be catalogued as the start of an "event" which will last until the command OFF is issued (to turn the recording off). The event will now be treated by SEISLOG as a normal event except for an indicator in the last character in the filename "M", which indicates that it is a manual trigger.

PAR Modify PARAM file and read parameters into shared memory.
Can be started from console.

PAR_SER Program to edit parameter file for multiserial support.
Can be started from console

PAR_SMP Program to edit parameter file for re-sampling
Can be started from console

PIC Not supported. Plot specified event on screen

PLT Plot events.
Can be started from console

PLTBGS Plot events.
Can be started from console

PRGHEAD Program header

PRIO Change priority on process. Debug only.
Can be started from console or terminal

PRL List parameter files. This program will list the specified parameter file to your screen. PRL 3 will list parameter file #3, PRL 1 or PRL will list parameter file #1.
Can be started from console or terminal

PRT_ONN Turn on print out on screen on events.
Can be started from console

PRT_OFF Turn off print out on screen on events.
Can be started from console

PWR Program to check the up-time of SEISLOG. SEISLOG-monitoring is normally activated by the SRT command. This utility will list out information on when SEISLOG was started and for how long it has been active.
Can be started from console or terminal

QNX_DIST Make installable distribution of SEISLOG
Can be started from console

RD_COLOR Reads color definitions for graphics

RD_PARAM Read parameter file(s).
Can be started from console or terminal

RD_PAR_SER Reads additional parameter file multiserial option

Can be started from console or terminal

RE_INITI This program resets the detection number to -1, error number and the detection record file. It should normally **only** be used during installation or when changing the size of the ringbuffer, digitizing unit, samplerate or the number of channels.

RE_START Enable watchdog timer after a specified number of seconds to force a reboot

REM_EVTS Remove all seismic events created by SEISLOG.

RESET_NM Nanometrics only: Reset unit, enter setup menu.

RNG_VER Verify ringbuffer system.

RST_GPS Turn off and on power on GPS when no data is received. Works only when supported by GPS clock.
Can be started from console or terminal

RST_SYST Enable watchdog timer via shared memory. And causes the program watchdog.c to reboot the system.

RTP This program plots any selection of the channels that are flagged for detection. **(Only among channels flagged for detection)** The data are plotted in real time. There is a limitation on how many channels can be plotted, depending on the data throughput (number of channels, sample rate etc.). In cases where a lot of channels are being plotted, some data will be lost, but the picture will give a reasonable impression of the data. The reason for this is that the data-acquisition and event recording always will have higher priority.
Can be started from console

SEI_ACT Make log of time every 10 second to check up time for SEISLOG.

SLOG Subroutine library

SRC_DST Make update distribution to floppy.

SRT This is the procedure to start SEISLOG. The command is entered and the data acquisition begins according to the parameter settings.

STP Stop data acquisition.

STS Print some current values from data module

SYS_BOOT	Reboot SEISLOG PC
SYS_WAT	Checks if SEISLOG is active.
TEKLIB	Library Tektronix
TIMEXROU	Time routine library
TS	Reads PDETNR and DETNR, BGS
VER	Print out version number of SEISLOG

8. System configuration and Licenses

Hardware:

PC

Standard PC 386/486/586 Floating Point Coprocessor optional
2 extra serial lines (strappable IRQs)

Software Licenses:

QNX Operating System
Watcom C compiler

9. Default .profile files

When you log in on one of the SEISLOG accounts, a predefined login file is executed automatically. In QNX this file is stored in the accounts home directory as .profile.

.profile files for the SEISLOG accounts:

For the **SEISLOG** account **/home/seislog/.profile**

```
#default .profile
PS1=SEISLOG
PATH=:/bin32:/bin:/usr/bin:/usr/ucb:/usr/local/bin:/home/seislog
echo ‘edit the file .profile if you want to change your environment.’
```

For the **EVENTS** account **/home/events/.profile**

```
#default .profile
PS1=EVENTS
PATH=:/bin32:/bin:/usr/bin:/usr/ucb:/usr/local/bin:/home/seislog
echo ‘edit the file .profile if you want to change your environment.’
```

For the **SEISMO** account **/home/work/.profile**

```
#default .profile
PS1=SEISMO
PATH=:/bin32:/bin:/usr/bin:/usr/ucb:/usr/local/bin:/home/seislog
echo ‘edit the file .profile if you want to change your environment.’
```

10. sysinit.1 file

Example /etc/config/sysinit.1 file:

```
set -i
export TZ=utc00
rtc -l hw
Dev &
Dev.con -n 4 &
reopen //0/dev/con1
kbd /etc/config/kbd/Norway
#-------------------------------------------------------------------------------
# for systems with only 2 serial ports (default after installation)
#-------------------------------------------------------------------------------
Dev.ser &
#-------------------------------------------------------------------------------
# for systems with 2 extra serial ports. The setup of the board
# has to be in accordance with the next line. Here 3f8,4 is
# serial port 1, 2f8,3 is serial port 2 and 3e8,5 serial port 3
# the numbers specify address and interrupt
#------------------------------------------------------------------------------
Dev.ser 3f8,4 2f8,3 3e8,5&
Dev.par &
Dev.pty &
Fsys.floppy &
#------------------------------------------------------------------------------
# for systems with a SCSI controller supporting tape the next
# line has to be included. The name of the driver has to be in
# accordance with the actual SCSI controller installed.
#------------------------------------------------------------------------------
Fsys.aha7scsi &
Net &
Net.ether1000 -l 1 -p 320 -i 10 &
Socket pcseis2 &
ifconfig en1 pcseis2 up
inetd &
#-----------------------------------------------------------------
# Include fonts for the graphic programs
#-----------------------------------------------------------------
export WINFONTS=/home/seislog/fonts
#-----------------------------------------------------------------
# Include mouse support on arrows (only on console)
#-----------------------------------------------------------------
Mouse ps2
#-----------------------------------------------------------------
# Include DOS file access from QNX
#-----------------------------------------------------------------
Dosfsys &
#-----------------------------------------------------------------
# Must be included to support graphics on console
```

```
#----------------------------------------------------------------
int10 &
Pipe &
emu87 &
#netmap -f
#freeze -cdz /etc/logo.F
#----------------------------------------------------------------
# Send RESET to modem if connected
#----------------------------------------------------------------
#echo "ATZ">/dev/ser3
#----------------------------------------------------------------
# Keeps track of when SEISLOG is active
#----------------------------------------------------------------
#/home/seislog/sei_act &
#----------------------------------------------------------------
# Re-boot & Re-start every 24-hour
#----------------------------------------------------------------
#/home/seislog/re_start &
#----------------------------------------------------------------
# Starts SEISLOG data acquisition
#----------------------------------------------------------------
#/home/seislog/srt &
#----------------------------------------------------------------
# Prepare for remote vt100 via modem on serial line /dev/ser3
#----------------------------------------------------------------
#tinit -T /dev/con* -t /dev/con1 -c "termdef vt100" -T /dev/ser3 &
tinit -T /dev/con* -t /dev/con1  &
```

11. Format descriptions

- **RINGBUFFER**
- **DETECTION RECORD**
- **EVENT FILES**

The ringbuffer is a file located in directory */home/buffer* on hard disk.
The ring buffer is allocated by the **RE_INITI** program on hard disk. The file is filled with data buffers from the A/D driver or from the programs supporting the digitizers sending data on serial lines. Data written to the file is retrieved from one of two buffers allocated in the shared data module. In the data module the two buffers are declared as 4 byte integers. The data buffers contain two basic parts: The HEADER and the DATA. The HEADER is 64 bytes long and contains a time stamp. The DATA part is of constant length for one parameter setting, but may vary with number of channels and sample rate. The data may be recorded as 2- or 4-byte integers.
At the beginning of every ringbuffer file is an event header identical to the header of ordinary events. This is to be able to treat each ring buffer file as if it was a normal event. This can be very useful when processing continuous data.

The detection record is a small summary of the main specifics of an event. It includes information like date and time of detection, the filename of the event, the detection number etc. The detection record is stored in a file allocated in directory /home/buffer. The name of the file is EVENT. The file is used as a circular file and has space for 1000 consecutive detection records before the first is overwritten. The format of one record is described below.
The record length is constant, 512 bytes.

The event files are stored in directory */home/events*. One file per detected event is stored here under a unique name. The file contains two basic parts: The HEADER and the DATA.
The HEADER here is not the same as the one described for the circle buffer.
The HEADER includes information about the event and also a copy of the detection record. The DATA part is a number of data buffers retrieved from the circle buffer, including the time stamping. The record length is always the same as the one for the circle buffer, but variable. The HEADER part is always at least 1024 bytes, but extended to be a multiple of the length of the data buffer. No information is recorded after byte 1024 in the header.

11.1 Data buffer header

DATA BUFFER HEADER

byte no.	ASCII characters			
01-04	**0**	**7**	**:**	**0**
05-08	**4**	**:**	**4**	**5**
09-12	**.**	**4**	**0**	**6**
13-16	**‘ ‘**	**0**	**1**	**/**
17-20	**0**	**9**	**/**	**1**
21-24	**9**	**9**	**4**	**‘ ‘**
25-28	**2**	**4**	**4**	**‘ ‘**
29-32	**S**	**E**	**T**	**‘ ‘**
33-36	**0**	**7**	**:**	**0**
37-40	**4**	**:**	**4**	**5**
41-44	**.**	**0**	**4**	**3**
45-48	**‘ ‘**	**0**	**1**	**/**
49-52	**0**	**9**	**/**	**1**
53-56	**9**	**9**	**4**	**‘ ‘**
57-60	**2**	**4**	**4**	**CR**
61-64	**NUL**	**NUL**	**NUL**	**NUL**
65-68	**data**	**data**	**data**	**Data**
69-72	**data**	**data**	**data**	**Data**
73-76	**data**	**data**	**data**	**Data**
.	**.**	**.**	**.**	**.**
.	**.**	**.**	**.**	**.**

The first 64 bytes are written as formatted binary ASCII characters. The data part is written as unformatted binary data.
The HEADER part in the example shows a typical timestamp of a data buffer:

07:04:45.406 01/09/1994 244 SET 07:04:45.043 01/09/1994 244
HH:MM:SS.MSC DD/MO/YEAR DOY STS HH:MM:SS.MSC DD/MO/YEAR DOY

HH	hour
MM	minutes
SS	seconds
MSC	milliseconds
DD	day
MO	month
YR	year
DOY	day of year
STS	status clock, SET-satelite fix, NSY- no satelite fix, TMO – not receiving anything from clock port. Next in the format is the time received from the clock.

11.2 Detection record

DETECTION RECORD

byte no. ASKII character

01-16	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**2**	**4**	**3**	**‘ ‘**
17-32	**1**	**9**	**9**	**4**	**‘ ‘**	**2**	**4**	**1**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**8**	**‘ ‘**	**‘ ‘**	**2**	**9**
33-48	**‘ ‘**	**‘ ‘**	**‘ ‘**	**9**	**‘ ‘**	**‘ ‘**	**3**	**3**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**5**	**.**	**9**	**2**	**1**
49-64	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**9**	**9**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**9**	**9**
65-80	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**1**	**2**	**.**	**0**	**0**	**0**
81-96	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**7**	**1**	**5**	**3**	**9**	**.**	**3**	**2**	**8**
97-112	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**5**	**5**	**0**	**4**	**.**	**8**	**6**	**7**
113-128	**‘ ‘**	**‘ ‘**	**1**	**2**	**8**	**2**	**3**	**7**	**3**	**2**	**4**	**8**	**.**	**0**	**0**	**0**
129-144	**‘ ‘**	**‘ ‘**	**‘ ‘**	**1**	**.**	**0**	**0**	**0**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**2**	**.**	**0**	**0**	**0**
145-160	**‘ ‘**	**‘ ‘**	**‘ ‘**	**3**	**.**	**0**	**0**	**0**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**4**	**.**	**0**	**0**	**0**
161-176	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**4**	**2**
177-192	**‘ ‘**	**‘ ‘**	**‘ ‘**	**0**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**1**
193-208	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**1**	**9**
209-224	**9**	**4**	**0**	**8**	**_**	**2**	**9**	**_**	**0**	**9**	**3**	**2**	**_**	**5**	**8**	**T**
225-240	**.**	**W**	**N**	**N**	**_**	**0**	**6**	**_**	**2**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**
241-256	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**
.	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**
.	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**
496-512	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**	**‘ ‘**

byte no.	**variable name**	**number of bytes**	**format**
001-016	detection number	15	askii I15
017-020	year	4	askii I4
021-024	day of year	4	askii I4
025-028	month	4	askii I4
029-032	day of month	4	askii I4
033-036	hours	4	askii I4
037-040	minutes	4	askii I4
041-048	seconds.milliseconds	8	askii F8.3
049-056	beamnumber (not in use)	8	askii I8
057-064	filternumber (not in use)	8	askii I8
065-080	duration seconds	16	askii F16.3
081-096	STA	16	askii F16.3
097-112	LTA	16	askii F16.3
113-128	amplitude	16	askii F16.3
129-136	frequency (n.d. set to 1)	8	askii F8.3
137-144	velocity (n.d. set to 2)	8	askii F8.3
145-152	azimuth (n.d. set to 3)	8	askii F8.3
153-160	power (n.d. set to 4)	8	askii F8.3
161-176	discblock addr. detection	16	askii I16
177-184	flag	8	askii I8
185-192	detection sample (n.d.)	8	askii I8
193-206	spaces	16	askii I16
207-233	filename	27	askii A27

11.3 Event file header/Ringbuffer header

EVENT FILE HEADER or RINGBUFFER HEADER

byte no. ASKII characters

001-016	**W**	**N**	**N**	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	**1**	**9**
017-032	**9**	**4**	sp	**2**	**4**	**2**	sp	sp	**8**	sp	**3**	**0**	sp	sp	sp	sp
033-048	Sp	**8**	sp	sp	**9**	sp	**2**	**0**	**.**	**7**	**3**	**4**	sp	sp	sp	sp
049-064	Sp	Sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	Sp	**6**
065-080	Sp	Sp	sp	sp	sp	sp	sp	**0**	**.**	**0**	**2**	**0**	**0**	**0**	**0**	**0**
081-096	sp	sp	sp	sp	**3**	**6**	**5**	**0**	sp	sp	sp	sp	sp	sp	Sp	**4**
097-112	**5**	**6**	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	**1**	**2**	**6**	**4**
113-128	sp	**1**	**2**	**6**	**4**	sp	sp	sp	sp	sp	sp	sp	sp	sp	**5**	**0**
129-144	**‘5‘**	**‘5‘**	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp
145-160	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp
161-176	**B**	**E**	**R**	sp	**S**	sp	sp	**Z**	**B**	**E**	**R**	sp	**L**	sp	sp	**Z**
177-192	**E**	**S**	**G**	sp	**S**	sp	sp	**Z**	**A**	**S**	**K**	sp	**S**	sp	sp	**Z**
193-208	**A**	**S**	**K**	sp	**S**	sp	sp	**N**	**T**	**E**	**G**	sp	**S**	sp	sp	**Z**
209-224	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp
225-240	**.**	**.**	**.**	.	**.**	.	.	**.**	**.**	**.**	**.**	.	**.**	**.**	**.**	**.**
689-704	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	sp	**3**	**3**
705-720	sp	sp	**9**	**4**	sp	**2**	**4**	**2**	**.**	**.**	**.**	**.**	**.**			

byte no.	**variable name**	**number of bytes**	**format**
001-003	network name	3	askii A3
015-018	year	2	askii I4
020-022	day of year	3	askii I3
024-025	month	2	askii I2
027-028	day	2	askii I2
033-034	hour	2	askii I2
036-037	minute	2	askii I2
039-044	seconds.msecs	6	askii F6.3
063-064	number of channels	2	askii I2
065-080	samplerate msecs	16	askii F16.7
081-088	no. of samp. each channel	8	askii I8
089-096	bytes per sample	8	askii I8
097-098	unit	2	askii I2
105-112	bytes per buffer	8	askii I8
113-117	bytes in header	5	askii I5
123-128	samples/buffer/channel	6	askii I6
129	ringbuffer status 5,1,0	1	askii A1
130	ringbuffer status 5,1,2	1	askii A1
161-164	station name ch1	4	askii I4
165-168	station component ch1	4	askii I4
169-172	station name ch2	4	askii I4
173-176	station component	4	askii I4
177-..			

689... from here the detection record is included as described under detection record on the previous page.

11.4 Resampled data file

HEADER

byte no.								ASKII character								
001-016	**1**	**9**	sp	**5**	**2**	sp	**1**	**0**	**.**	**1**	**2**	**2**	sp	sp	sp	sp
017-032		**4**	sp	**1**	**0**	sp	**9**	**8**	sp	sp	sp	sp	sp	sp	sp	sp
033-048						**7**	**2**	**0**								**3**
049-064							**1**	**0**							**1**	
065-080	**0**	**.**	**0**	**2**	**0**	**0**	**0**	**0**	**4**	**3**				**6**	**0**	**0**
081-096								**2**	sp	sp	sp	sp	sp	sp	**L**	**P**
097-112				**0**	**.**	**1**	**0**	**0**				**0**	**.**	**1**	**0**	**0**
113-128	**L**			**Z**	sp	**L**			**N**	sp	**L**			**E**	sp	sp
129-144	**S**	**R**	**1**		sp	**S**	**R**	**2**		sp	**S**	**R**	**3**		sp	sp
145-160								**7**						**4**	**2**	**0**

byte no.	**variable name**	**number of bytes**	**format**
001-002	hour	2	askii I2
004-005	minute	2	askii I2
007-012	seconds.milliseconds	6	askii F6.3
017-018	day of month	2	askii I2
020-021	month	2	askii I2
023-024	year	2	askii I2
033-040	recordlength (bytes)	8	askii I8
041-048	number of channels	8	askii I8
049-056	no. of minutes per file	8	askii I8
057-064	resampling frequency	8	askii I7
065-072	actual sample interval	10	askii F10.8
073-080	total samples per channel	8	askii I8
081-088	number of filter poles	8	askii I8
095-096	filter type	2	askii A2
097-104	filter low	8	askii F8.3
105-112	filter high	8	askii F8.3
113-116	component 1.channel	4	askii A4
118-121	component 2.channel	4	askii A4
123-126	component 3.channel	4	askii A4
129-132	station name 1.channel	4	askii A4
134-137	station name 2.channel	4	askii A4
139-142	station name 3.channel	4	askii A4
145-152	current number of minutes	8	askii I8
153-160	current no. of samples/ch	8	askii I8

DATA

The data samples are written in binary format as 4 byte integers.

12. Useful QNX commands and utilities

The following are some of the basic commands and utilities in the QNX operating system. These are for the first time user to enter the QNX world. Use the QNX documentation for more detailed information. For the following examples the terminal line is set to VT100:

cd /home/work	Change to the work directory
cd /home/events	Change to the events directory
list onn.c	List the file on your terminal
vedit test.c	Activate the editor and make a new or edit the old file.

cp pg1.c pg2.c	Copy file
rm pg1.c	Remove a file from your current directory
pwd	Show current source directory

In some cases it is convenient to have all of the output to terminal saved in a separate file. This is done by simply redirecting the standard output to the file you wish:

cat prg1.c>savelist The listing of prg1.c is redirected (>) to savelist.
See more about redirecting in the QNX manuals.

13. Setup Modem

To have modems work properly there are some steps that have to be done in advance. In the /etc/config/sysinit.1 the command tinit at the bottom of the file has to be modified to include the login facility on the specific line where the modem will be connected:

tinit –T /dev/con* -t /dev/con1 –c ”termdef vt100” –T /dev/ser3 &

The /dev/ser3 indicates the port where the modem can be connected.

Below is described the setup of a modem of type US-Robotics Sportsler Flash.
In this case we connect the modem to serial port /dev/ser1 and set the line to support vt100 emulation.
To set up the modem parameters we connect directly to the modem with the utility qtalk:

qtalk –m /dev/ser1

and start with the command:

at

The modem will normally answer with OK. If it does not answer it might be that the modem is already in ”quiet” mode. To verify this, issue the command **atq0** and **ate1** and then the command **at** once more. If the modem answers OK you are in contact with the modem and may proceed with the setup. If not, there are probably some problems with the cable you are using.
To proceed with the set up you issue the following commands:

ats0=3 answer call after 3 rings
the modem answers **OK**

ats19=1 send hang-up signal if no activity on line for 1 minute
the modem answers **OK**

ate0 turn off local echo
the modem answers **OK**

atq1 set modem in quiet mode
the modem do not respond (because its now in quiet mode)

at&w0 store your settings in NVRAM profile 0
the modem do not respond (because its now in quiet mode)

The next command is not part of the profile, but ensures that the profile above is loaded and used on restart.

aty0 loads NVRAM profile 0 at restart

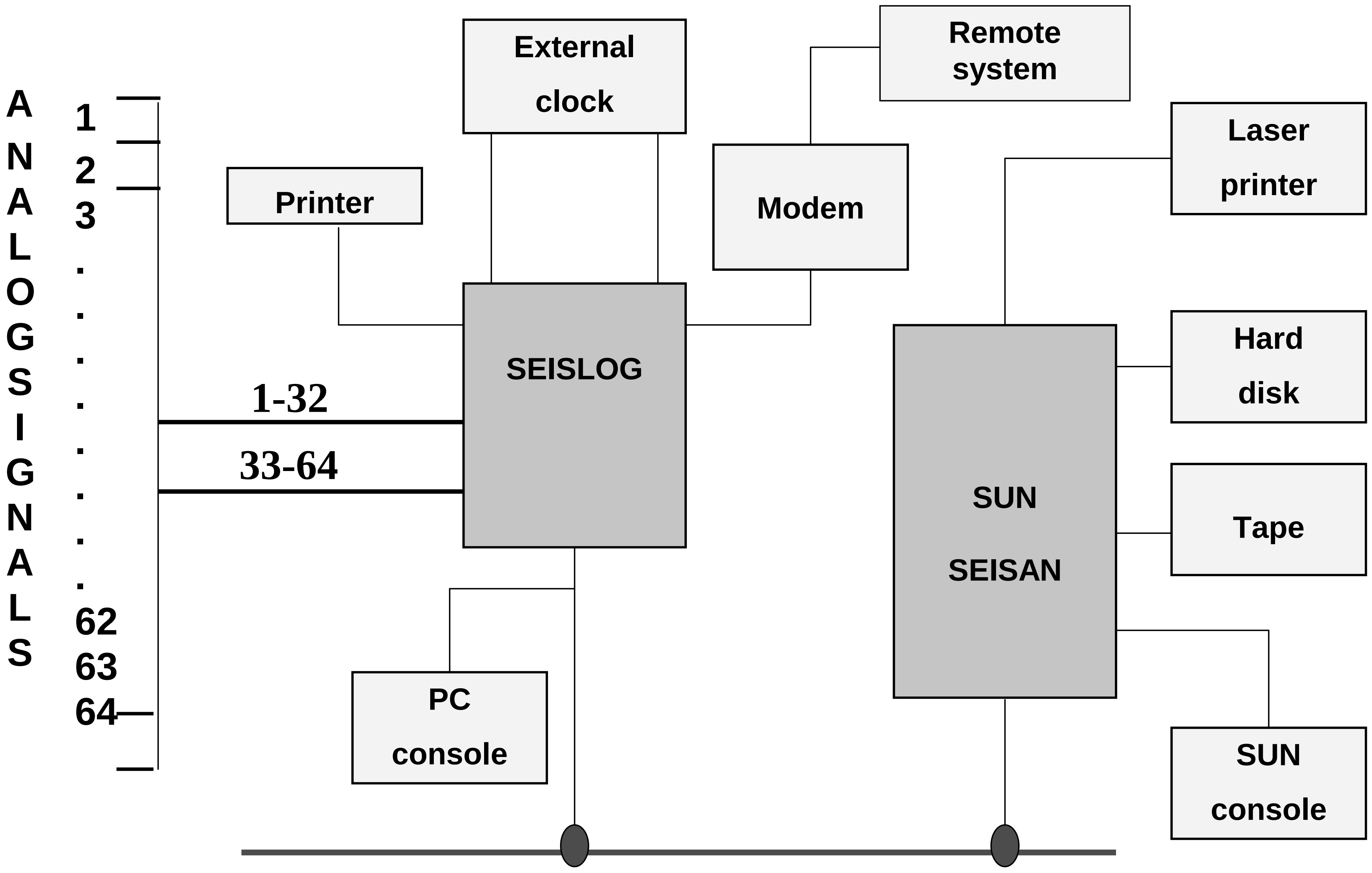
The modem will now answer external calls after 3 rings and try to establish a connection. The serial port is set to a speed of 9600 baud by default. A character has to be issued (enter) to get a response from the remote machine. The machine will answer with the standard login sequence. Standard commands can now be issued to the remote machine as if connected directly as a vt100 terminal. The logout sequence is by logout as normal. If you are logged in via f.ex. the EM4105 terminal emulator from a PC, the logout command will disconnect you from the remote PC, but the modem connection is still active. If you enter a ”enter” character, the login sequence will be re-activated, and you can log in to the system once more. If you want to log out and disconnect the modem connection you issue +++ rapidly and you will be talking directly to the modem. Your local modem will answer with OK, and you now issue the command **ATH** to hang-up the line. If you just issue the logout command and do not touch the keyboard for 1 minute, the remote modem will disconnect the line automatically. This will also happen when you are logged in as normal and there are no activity on the line for 1 minute.
If you are connected to the remote machine through SEISNET, more or less the same things will happen. SEISNET automatically logs in under the account EVENTS and the prompt EVENT shows up on your screen. Now you can do your commands as normally. If you issue a logout command, SEISNET will automatically log you out and also disconnect the modem connection. If there are no activity on the line for 1 minute, the remote modem will do the disconnection as described above.

Different modems has different ways of setting the disconnect option. Normally you can find this option described for another S-register in the modem manual.

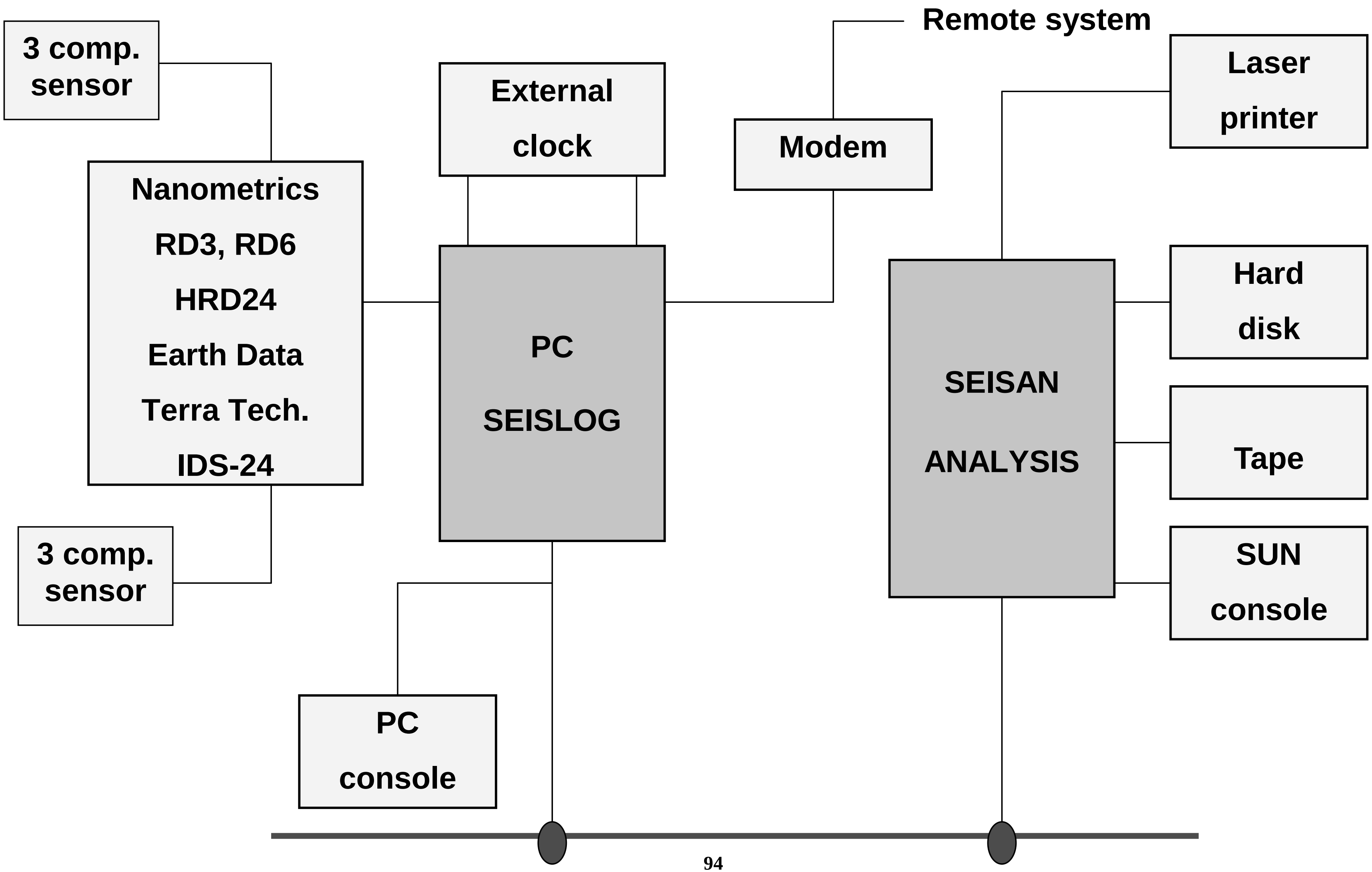
For this setup, standard cables delivered with the modem were used.

14. Configurations and different types of networks

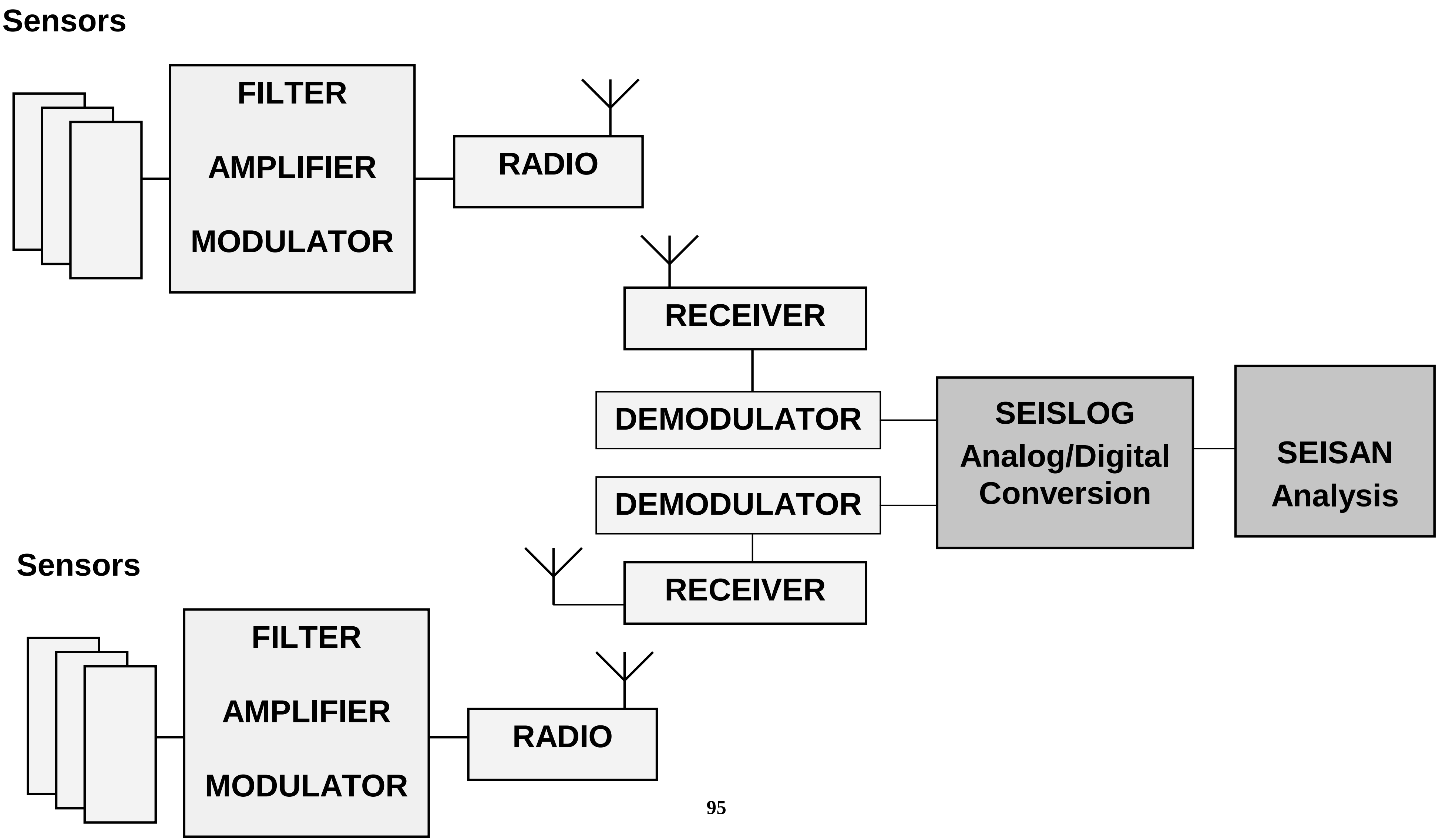
SEISLOG with A/D and Sun SEISAN system



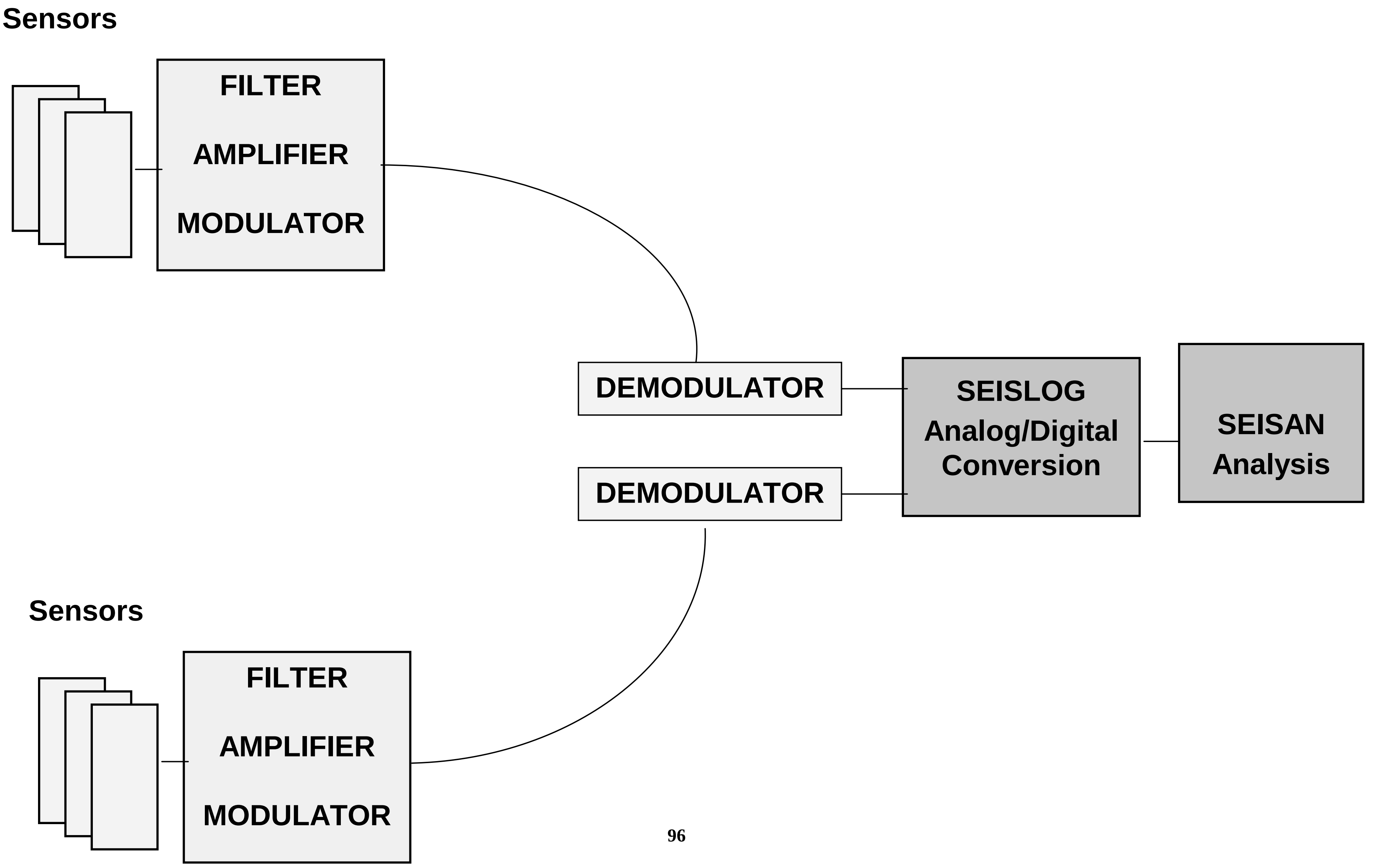
SEISLOG with different digitizers and SUN SEISAN system



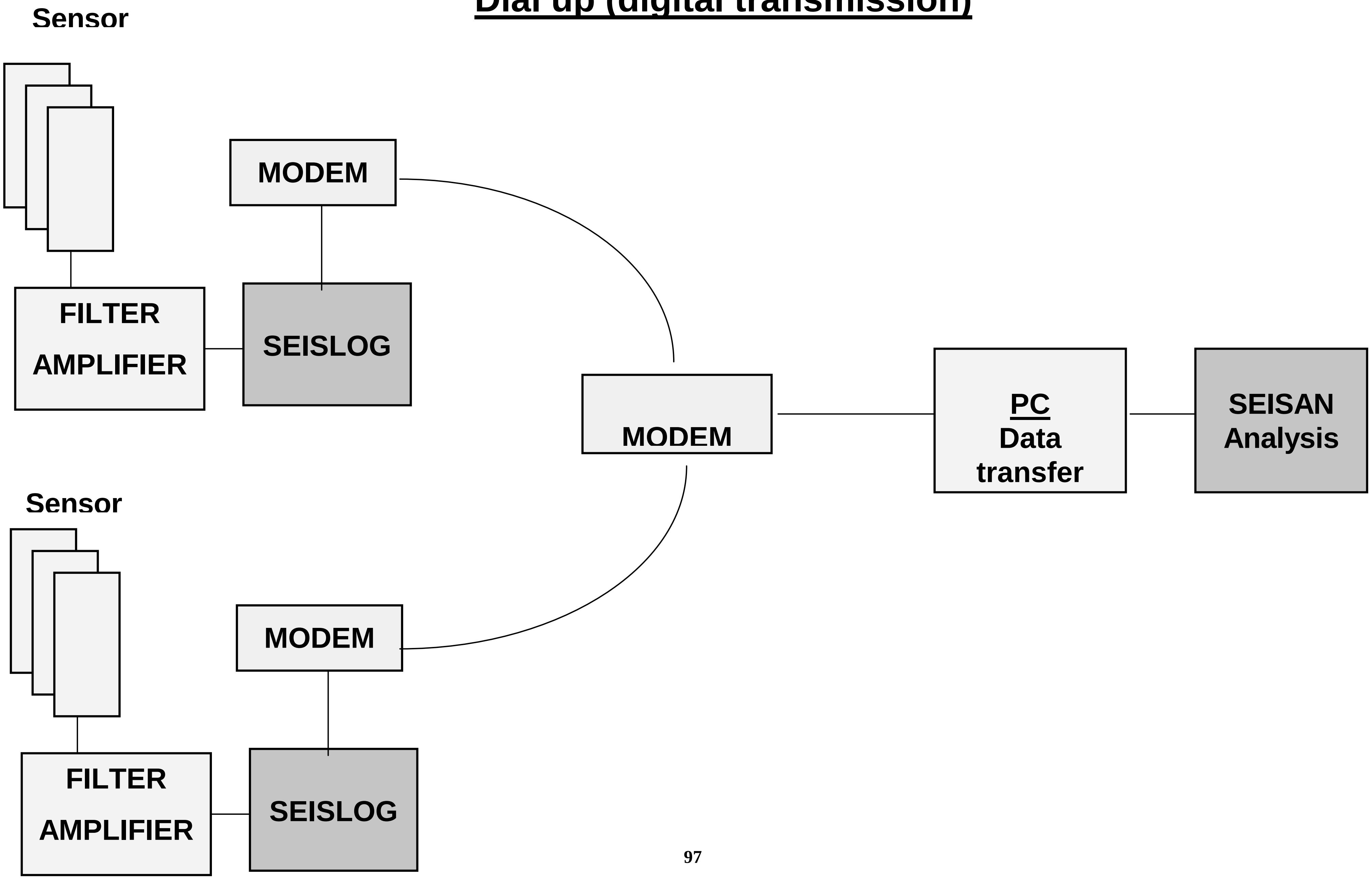
Analogue radio transmission



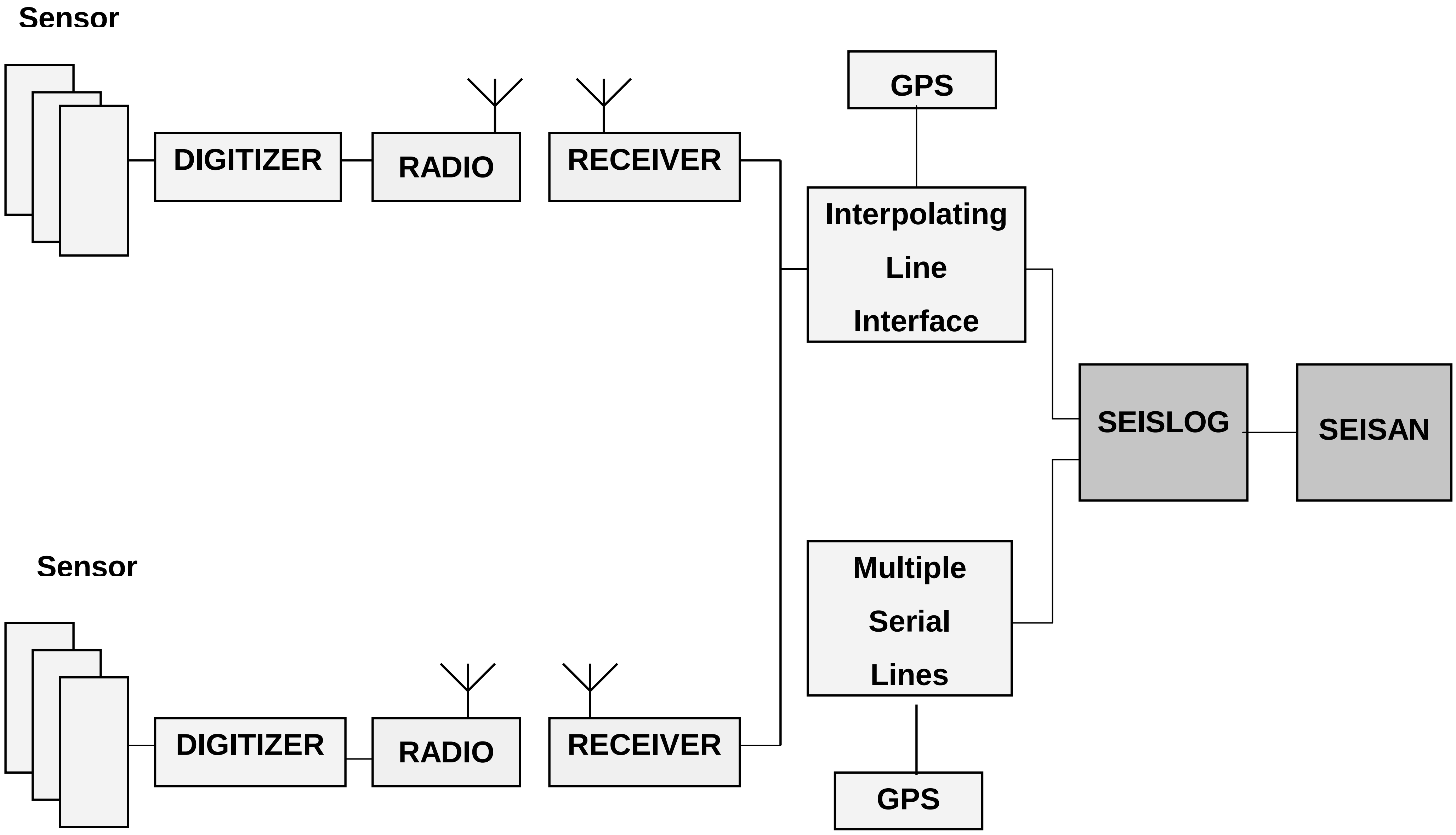
Analog leased telephone lines



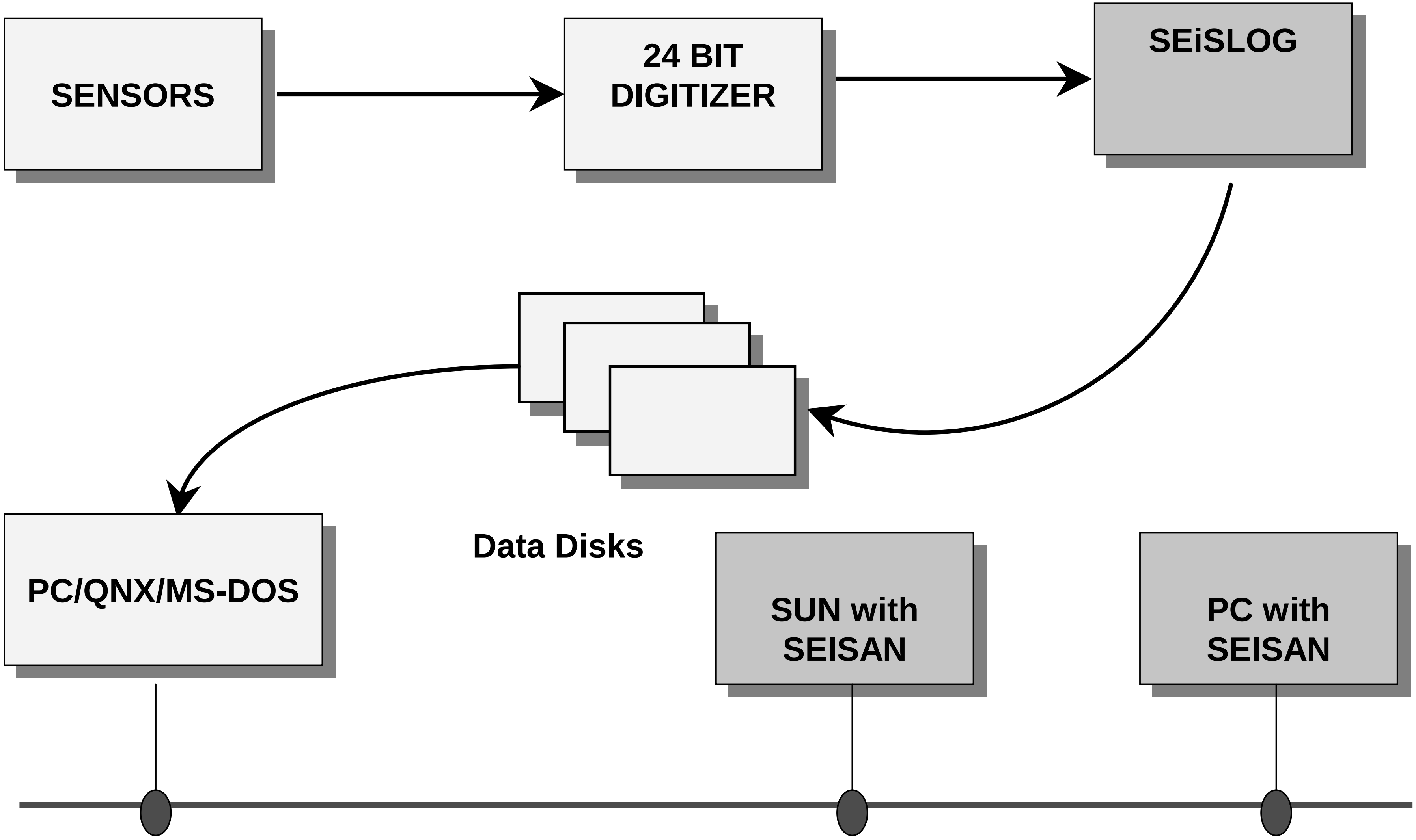
Dial up (digital transmission)



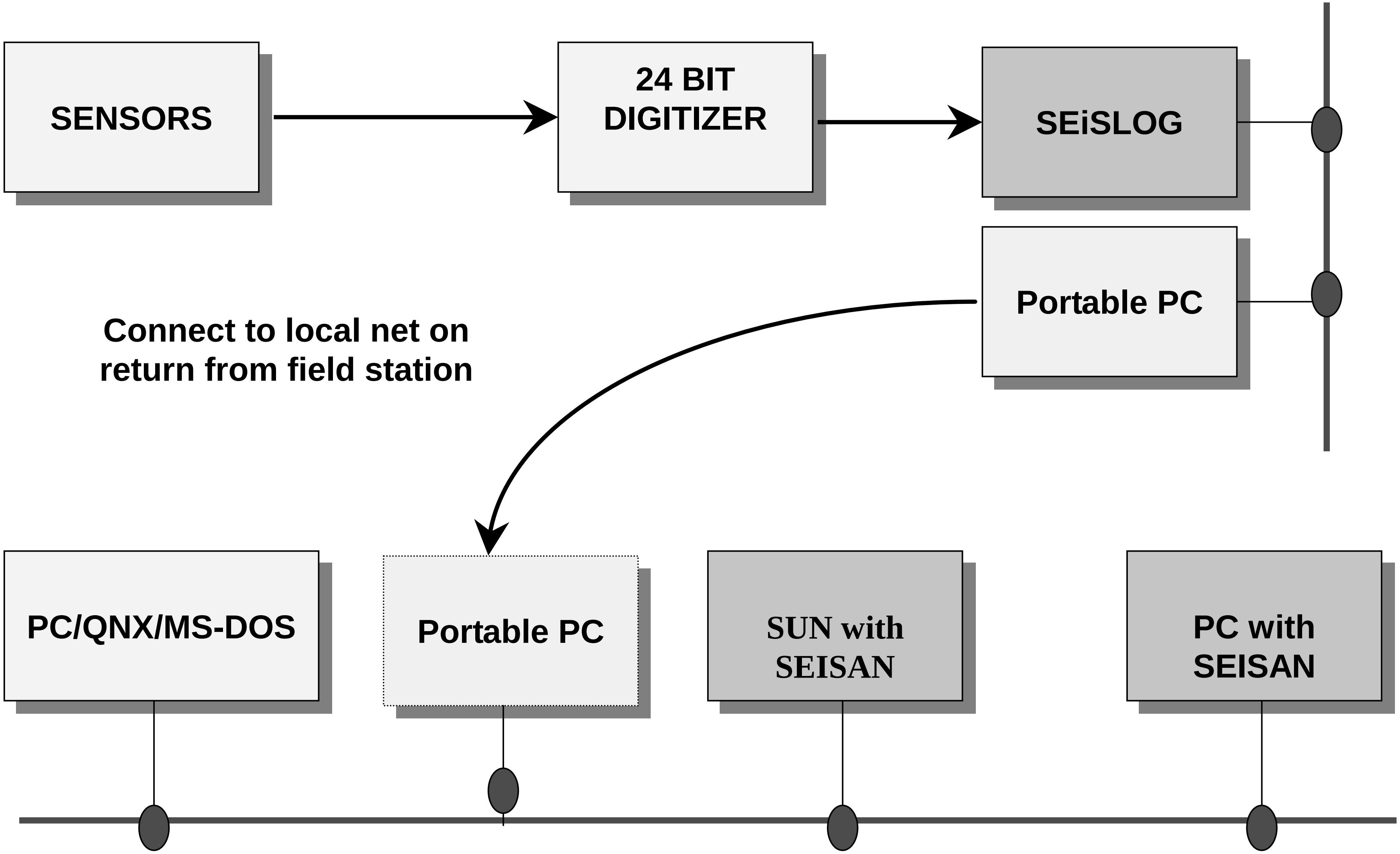
Digital radio transmission



Data from Portable Stations to Data Center Offline



Data from remote stations to data center offline via portable PC



SENSORS

24 BIT DIGITIZER

SEiSLOG

Portable PC

Connect to local net on return from field station

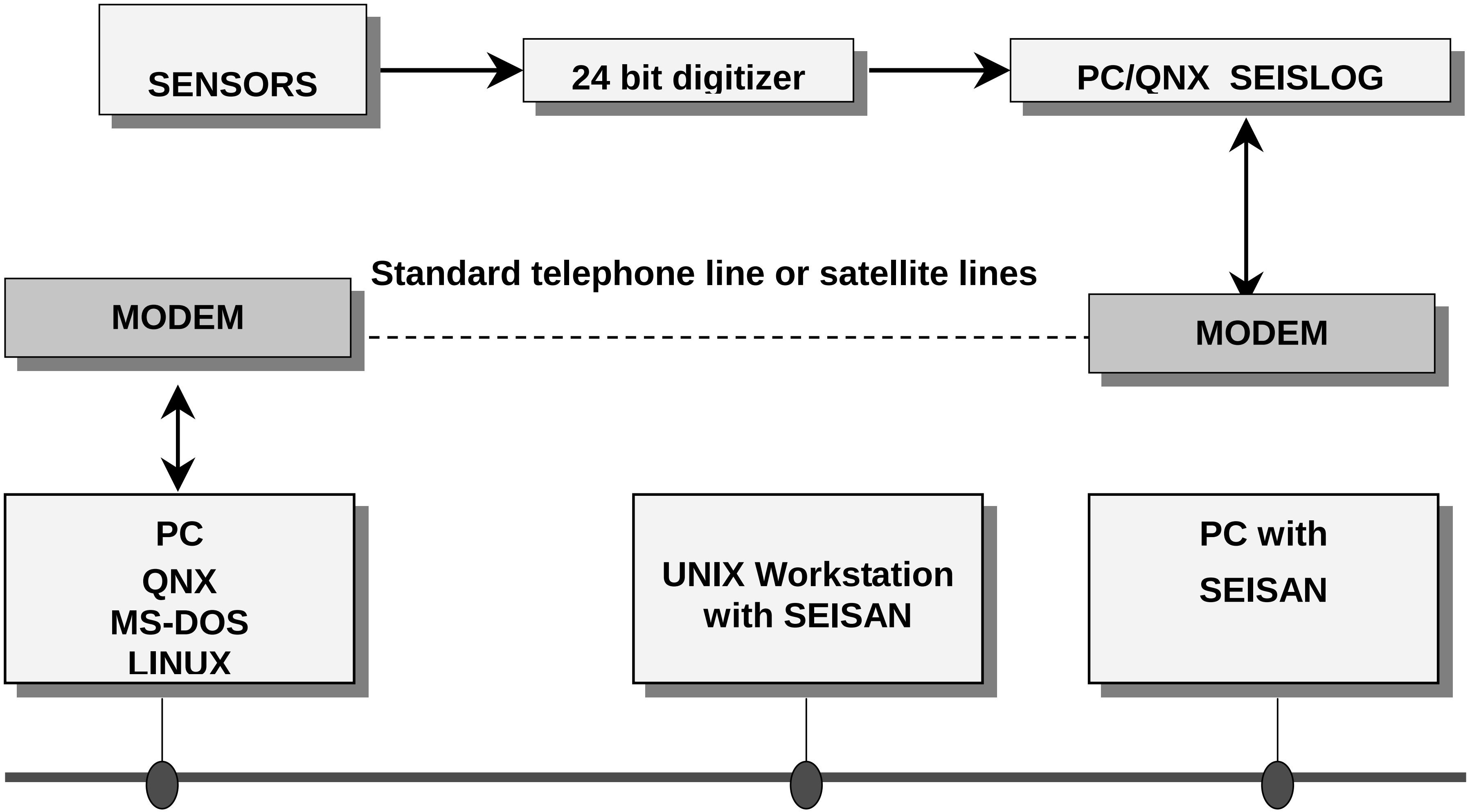
PC/QNX/MS-DOS

Portable PC

SUN with SEISAN

PC with SEISAN

Data from remote stations to data center online through modems.



SENSORS

24 bit digitizer

PC/QNX SEISLOG

Standard telephone line or satellite lines

MODEM

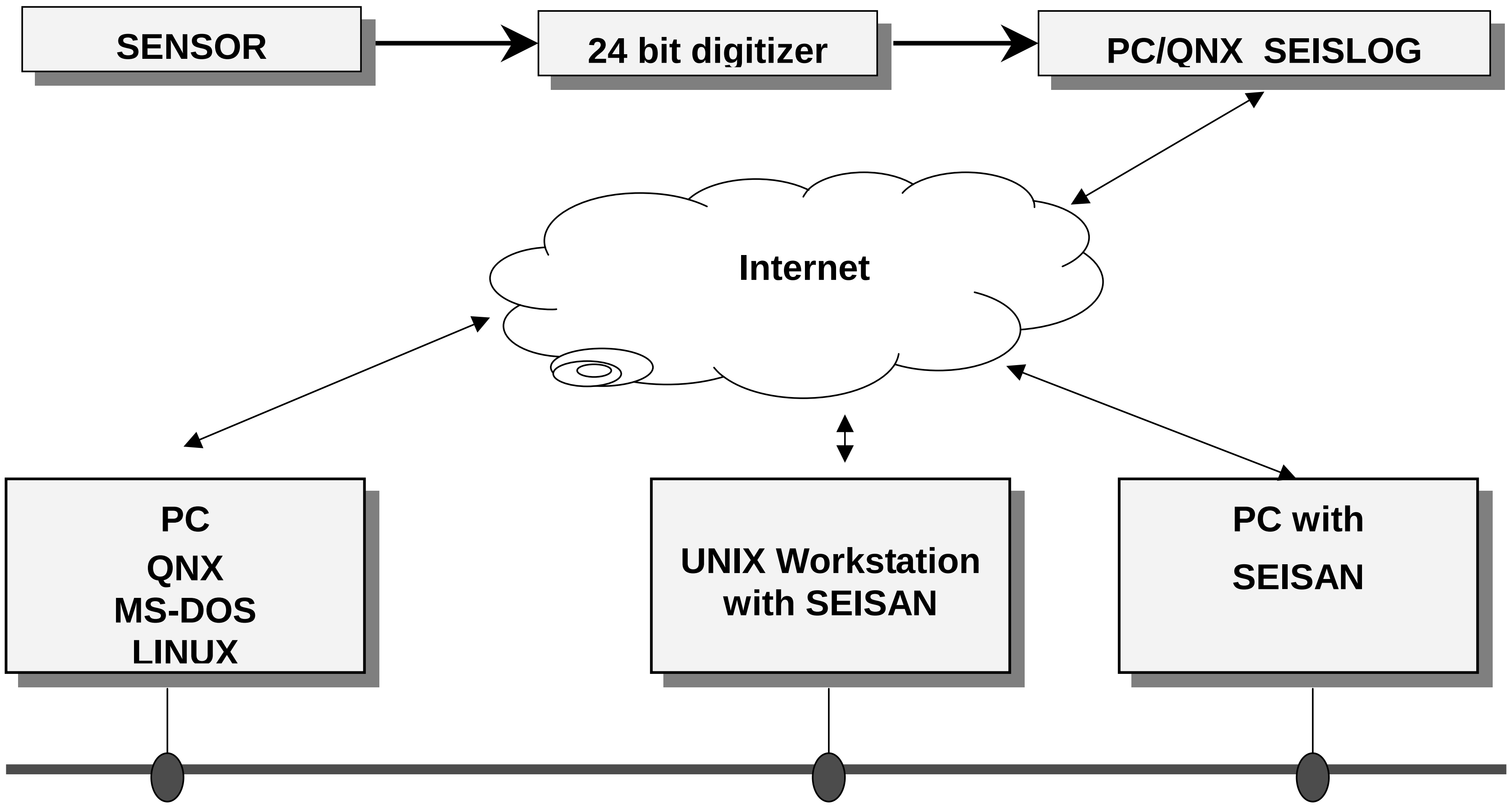
MODEM

PC
QNX
MS-DOS
LINUX

UNIX Workstation
with SEISAN

PC with
SEISAN

Data from remote stations to central through Internet



SENSOR

24 bit digitizer

PC/QNX SEISLOG

Internet

PC
QNX
MS-DOS
LINUX

UNIX Workstation
with SEISAN

PC with
SEISAN

15. Installations

British Geological Survey, Edinburgh, Scotland

School of Cosmic Physics, Geophysics Section, Dublin, Ireland

Instituto per la Geofisica della Litosfera, Milano, Italy

Universidad de Costa Rica, Laboratorio de Sismologia, San Jose, Costa Rica.

Universidad Nacional, Observatorio Vulcanologico y Sismologico de Costa Rica, San Jose, Costa Rica.

Universidad Nacional Autonoma de Honduras, Tegucigalpa, Honduras.

Universidad de Panama, Instituto de Geociencias IGC-UPA, Panama.

Instituto de Sismologia, Vulcanologia y Meteorologia,INSIVUMEH, Guatemala

INETER, Managua, Nicaragua

Universidad de Chile, Departemento de Geofisica, Santiago Chile.

Universidad de Chile, IDIEM, Santiago Chile

Kort og Matrikkelstyrelsen, Geodætisk Division, Seismology, Copenhagen, Denmark.

Uppsala University, Seismological Department, Uppsala, Sweden.

Departement of Geological Survey and Mines, Entebbe, Uganda.

FUNVISIS, Caracas, Venezuela.

Universidad del Oriente, Cumana, Venezuela.

UNAM, Mexico