

Norwegian National Seismic Network

Technical Report No. 21



Departamento de Geofísica
Facultad de Ciencias Físicas y Matemáticas
Universidad de Chile



Institute of Geoscience
University of Bergen, Norway

SEISLOG – Linux Embedded systems (English summary)

Terje Utheim

Department of Earth Science/University of Bergen
terje.utheim@geo.uib.no

Ulises Vásquez Casanova

Departamento de Geofísica/Universidad de Chile
ulises@dgf.uchile.cl

Dept. of Earth Science, University of Bergen

Allégt. 41, N-5007 Bergen, Norway

Tel: +47-55-583600 Fax: +47-55-583660 E-mail: seismo@geo.uib.no

February 2006

Index

	Page
Introduction	3
1. Objectives	3
2. Implementation	3
3. Types of Embedded Systems	3
4. Laboratory test	6
4.1 Embedded w/ Digitiser "EarthData 3CH 24bit"	6
4.1.1 "ACME/FOX"	6
4.1.2 "GUMSTIX"	8
4.1.3 "TS-7260"	9
4.1.4 Test results	11
4.2 Embedded w/ Digitiser "SADC10-16bit"	12
4.2.1 "ACME/FOX"	12
4.2.2 "TS-7260"	13
4.2.3 Test results	15
4.3 Embedded w/ Digitiser "TDT3C"	15
4.3.1 "ACME/FOX"	15
4.3.2 "TS-7260"	16
4.3.3 Test results	18
5. Conclusions	18
6. Further testing in Bergen after visit to Chile	20

Introduction

The seismic stations at the University of Chile, Department of Geophysics, have up to now been running SEISLOG under the operating system QNX (commercial real time distribution). This has permitted a significant advancement compared to the previous network based on other systems, not only due to lower costs compared to other systems, but also the flexibility to adapt to different seismic investigation projects.

The SEISLOG QNX system has also been object to changes in technology and discontinuity due to new versions of software. QNX is a commercial licensed system and further development has been discontinued. This limits the possibility to realize new drivers for units like digitisers and GPS receivers. This fact limits the further development of SEISLOG under the QNX operating system.

This is where Linux turns out as a good alternative to implement the SEISLOG system which is already installed on almost all seismic stations in the country.

The main reasons to use the SEISLOG Linux is:

- Free distribution of software.
- Open source code.
- The existence of different distributions of the kernel.
- Wide range of support for different devices.
- Standard version can also be installed on different Embedded Systems.

1. Objectives

- Determine the embedded system most convenient for implementation of SEISLOG Linux on seismic stations
- Do laboratory tests to establish main advantages and disadvantages for the different systems.
- Establish procedures for configuration, installation and also functionality for each system with the digitisers: TDT3C 24bit, EarthData 3CH 24bit and SADC10 16bit.

2. Implementation

The migration of SEISLOG from QNX to Linux free distribution seems to be a powerful tool in the case of both standard PCs and for embedded systems. This include both the flexibility to support a wide variety of hardware and also the design of recorders and seismic stations based on embedded systems.

Several systems have been designed based on the small PC/104 Roadrunner II. However, the high cost and the relatively high energy consumption of these systems justify a change towards embedded systems described in this report.

3. Types of embedded systems

There exists a big variety of embedded devices with a wide range of technical characteristics and a wide price range. Most of them are orientated towards control and automation functions, and some of them with a stripped Linux operating system on board. This is one of the main characteristics that make it relatively easy for the user to design applications.

The implementation of SEISLOG-Linux in these systems seems to be a powerful tool to detect seismic events and to record seismic data on small cost effective systems with relatively low power consumption.

The embedded systems described in this report:

- ACME/FOX
- GUMSTIX
- TS-7260

Each of these systems contain in general: CPU, memory, I/O ports and a compact Linux operating system stored in memory. Programs must be compiled on a separate PC with a corresponding development system and cross-compiler installed. Executable files can be transferred to the embedded system for execution. However, one must always keep in mind that the memory recourses are limited on embedded systems.

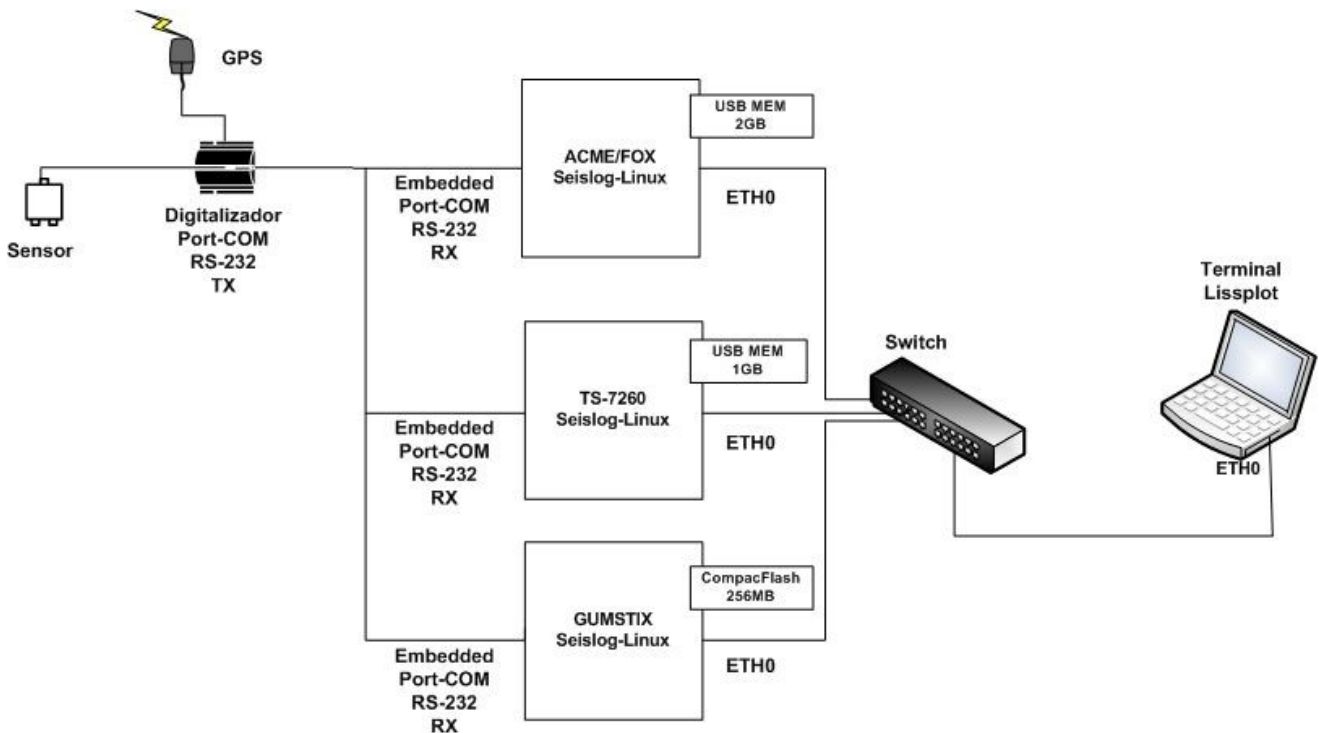
In order to show the main characteristics, advantages and disadvantages, the table below presents some of the criteria and parameters to decide the system most adequate to design a seismic station with the implementation of SEISLOG Linux.

Comparative table Embedded Systems for SEISLOG/LINUX

PARAMETERS	ACME/FOX	GUMSTIX	TS-7260
OS Linux	Linux Kernel 2.6.12	Linux Kernel 2.6.11	Linux Kernel 2..4.26
Services: network and other relevant software	-FTP Server -SSH Server (Only Terminal) -SCP -TELNET Server -PPPd -GCC external -HTTTPd	-SCP -SSH server (Only Terminal) -GCC external -IPtables -HTTTPd	-Telnetd -Ftpd -Cron -GCCext
Peripherals	-1xCOM -ETH0 10/100baseT -2xUSB 1.1 (pendrive. Tested: 2GBx1port) -2 extensions IDE,SCSI	-COM1 (Console) -COM2 -ETH0 10/100baseT -Socket Memory CFlash or MMC -1xUSB 1.1 (Terminal Serv)	-ETH0 10/100BaseT -3xCOM -2xUSB -Socket CompacFlash
Memory	-Flash OS=4MB -Memory RAM=16MB -External=PenDrive	-Flash OS=4MB -Memory RAM=64MB -External MMC/CF up to 512MB in test	-Flash OS=8MB de 32MB -Memory RAM=32/64MB
Processor	AXIS 100MHz	Intel XScale 400 MHz	CPU 200MHz
Consumption	5VDC, 12VDC 260mA=ETH0,seislog 300mA=ETH0,seislog,usb 270mA=seislog,usb	4,5-5VDC Model CF: 500mA+200mA Model MMC: 400mA	Power In=4,5-20vdc 5VDC/ 220mA=usb,seislog 190mA=ETH0,usb 300mA=ETH0,seislog,usb
Production country	ITALIA www.acmesystems.it	CANADA www.gumstix.com	USA www.embeddedarm.com
Dimensions	6,6x7,2x2cm	11x3,5x1,5cm	10x13cm
Price	EUR \$154.80(Italia)	Complete kit U\$216.00	U\$179.00 (RAM=32MB) U\$195.00 (RAM=64MB)

4. Laboratory test

The following configuration was used in the test:



The signal from the digitiser was input to each system in parallel. The program LISSPLOT was used from a separate PC over Telnet or SSH to monitor different components from each system.

4.1 Embedded w/ Digitiser "EarthData 3CH 24bit"

Pin-Out of the serial port used for the digitiser data.

EarthData		ACME/FOX		TS-7260		GUMSTIX	
2	TX	3	RX	2	RX	3	RX
3	RX	-	-	-	-	-	-
4	+12v	-	-	-	-	-	-
5	GND	5	GND	5	GND	5	GND

In the following some details of the necessary setup for the SEISLOG Linux for each system:

4.1.1 "ACME/FOX"

Mount the USB memory:

```
#mount -t vfat /dev/sda1 /mnt/1
```

One has to create a directory structure and the following files. All must be done as root:

```
/mnt/flash/SLG/  
/COM
```

```
/LOG
/PAR
/PRO
/RNG
/CAL
```

/mnt/flash/SLG/PAR/syspar

```
Events.....:/mnt/1/SLG/EVT
Events S-files.....:/mnt/1/SLG/EVT
Ringbuffers.....:/mnt/1/SLG/RNG
Ringbuffers S-files:/mnt/1/SLG/RNG
```

/mnt/flash/SLG/PAR/CHA_ED3

NETWORK NAME:

UIB

TRIGGER PARAMETERS:

```
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
      30      40      20      3      500
```

CHANNEL PARAMETERS:

SEQ	DIG	CH	MBUF	FSIZ	DAY	STAT	COMP	LOC	F-LOW	F-HIGH	STA	LTA	TRIG	DTRIG	DSK	REC
1	1	1	25	20	10	BERG	BHZ	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
2	1	2	25	20	10	BERG	BHN	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
3	1	3	25	20	10	BERG	BHE	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1

/mnt/flash/SLG/PAR/DIG_ED3

NO	UNIT	SER	PORT	BAUDRATE	IP-ADDRESS/DOMAIN	SOCKET	SRATE	TIMING	TIME-CORR
1	ED3_S		COM01	19200		-	50	2	0.0

The executable programs must be located in the path described below. "seislog" is the compiled program that do the data acquisition, detection and recording. It also send data to the server "srv" that supplies data for monitoring and plotting from a client (plotting port 4000/tcp). "seislog" and "srv" are transferred from the development system to the embedded system with SCP or FTP.

"seislog" = /mnt/flash/SLG/PRO

"srv" = /mnt/flash/SLG/PRO

Executing seislog:

Before actually starting SEISLOG one should check the serial port that will receive the data from the digitiser to see if it is opened for reading. In this case we are using the port ttyS0 which corresponds to COM01 in the parameter file above.

```
#ls -l /dev/ttyS*
```

```
crw-rw-rw- 1 root    tty      4,  64 Oct 24 2005 /dev/ttyS0
crw-rw-rw- 1 root    tty      4,  65 Oct 24 2005 /dev/ttyS1
crw-rw-rw- 1 root    tty      4,  66 Oct 24 2005 /dev/ttyS2
crw-rw-rw- 1 root    tty      4,  67 Oct 24 2005 /dev/ttyS3
```

```
#cat /dev/ttyS0
```

```
#cd /mnt/flash/SLG/PRO
```

```
#stty raw</dev/ttyS0      (normally handled by SEISLOG)
```

```
#stty 19200</dev/ttyS0 (normally handled by SEISLOG)
#./seislog -tag -d DIG_ED3 -c CHA_ED3 -t /mnt/flash
```

In the case of the EarthData digitiser, the time is included in the data stream.

4.1.2 "GUMSTIX"

Mount the MMC memory:

To mount the MMC memory on the Gumstix, one has to do a modification in the file /etc/profile:
 Uncomment #mmc-block (this is to mount the MMC memory automatically).

To mount the MMC card:

```
#mount /dev/mmc/mm0/part1 /mnt/mmc
```

One has to create a directory structure and the following files. All must be done as root:

```
/mnt/mmc/SLG/
    /COM
    /LOG
    /PAR
    /PRO
    /RNG
    /CAL
```

/mnt/mmc/SLG/PAR/syspar

```
Events.....:/mnt/mmc/SLG/EVT
Events S-files.....:/mnt/mmc/SLG/EVT
Ringbuffers.....:/mnt/mmc/SLG/RNG
Ringbuffers S-files:/mnt/mmc/SLG/RNG
```

/mnt/mmc/SLG/PAR/CHA_ED3

```
NETWORK NAME:
UIB
TRIGGER PARAMETERS:
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
      30          40                20          3          500
CHANNEL PARAMETERS:
SEQ DIG CH MBUF FSIZ DAY STAT COMP LOC F-LOW F-HIGH STA LTA TRIG DTRIG DSK REC
  1  1  1  25  20  10 BERG BHZ 00  2.0  8.0  4.0 400.0 2.5  1.5  1  1
  2  1  2  25  20  10 BERG BHN 00  2.0  8.0  4.0 400.0 2.5  1.5  1  1
  3  1  3  25  20  10 BERG BHE 00  2.0  8.0  4.0 400.0 2.5  1.5  1  1
*****
```

/mnt/mmc/SLG/PAR/DIG_ED3

```
NO UNIT SER_PORT BAUDRATE IP-ADDRESS/DOMAIN SOCKET SRATE TIMING TIME-CORR
  1 ED3_S COM03 19200 - - 50 2 0.0
*****
```

The executable programs must be located in the path described below. "seislog" is the compiled program that do the data acquisition, detection and recording. It also send data to

the server "srv" that supplies data for monitoring and plotting from a client (plotting port 4000/tcp). "seislog" and "srv" are transferred from the development system to the embedded system with SCP or FTP.

```
"seislog" = /mnt/mmc/SLG/PRO
"srv"     = /mnt/mmc/SLG/PRO
```

Executing seislog:

Before actually starting SEISLOG one should check the serial port that will receive the data from the digitiser to see if it is opened for reading. In this case we are using the port ttyS2 which corresponds to COM3 in the parameter file above.

```
#ls -l /dev/ttyS*
crw-rw-rw-  1 root    tty      4,  64 Oct 24  2005 /dev/ttyS0
crw-rw-rw-  1 root    tty      4,  65 Oct 24  2005 /dev/ttyS1
crw-rw-rw-  1 root    tty      4,  66 Oct 24  2005 /dev/ttyS2
crw-rw-rw-  1 root    tty      4,  67 Oct 24  2005 /dev/ttyS3
```

```
#cat /dev/ttyS2
```

```
#cd /mnt/mmc/SLG/PRO
#./seislog -tag -d DIG_ED3 -c CHA_ED3 -t /mnt/mmc
```

4.1.3 "TS-7260"

Mount USB memory:

Mount the modules needed to use the USB memory:

```
/usr/bin/loadUSBModules.sh
```

```
insmod usbcore
insmod pcipool
insmod usb-ohci
insmod usb-ohci-ep93xx
insmod scsi_mod
insmod sd_mod
insmod usb-storage;
```

```
$cd /usr/bin
$./loadUSBModules.sh
mount -t vfat /dev/scsi/host0/target0/lun0/part1 /mnt/cf
```

One has to create a directory structure and the following files. All must be done as root:

NB: In this case the SEISLOG executables can be installed in the internal flash memory, and the events and ringbuffers is recorded on the USB.

```
/root/SLG/
          /COM
```

```
/LOG
/PAR
/PRO
/RNG
/CAL
```

/root/SLG/PAR/syspar

```
Events.....:/mnt/cf/SLG/EVT
Events S-files.....:/mnt/cf/SLG/EVT
Ringbuffers.....:/mnt/cf/SLG/RNG
Ringbuffers S-files:/mnt/cf/SLG/RNG
```

/mnt/cf/SLG/PAR/CHA_ED3

NETWORK NAME:

UIB

TRIGGER PARAMETERS:

```
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
      30          40                20          3          500
```

CHANNEL PARAMETERS:

SEQ	DIG	CH	MBUF	FSIZ	DAY	STAT	COMP	LOC	F-LOW	F-HIGH	STA	LTA	TRIG	DTRIG	DSK	REC
1	1	1	25	20	10	BERG	BHZ	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
2	1	2	25	20	10	BERG	BHN	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
3	1	3	25	20	10	BERG	BHE	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1

/mnt/cf/SLG/PAR/DIG_ED3

NO	UNIT	SER_PORT	BAUDRATE	IP-ADDRESS/DOMAIN	SOCKET	SRATE	TIMING	TIME-CORR
1	ED3_S	COM02	19200	-	-	50	2	0.0

The executable programs must be located in the path described below. "seislog" is the compiled program that do the data acquisition, detection and recording. It also send data to the server "srv" that supplies data for monitoring and plotting from a client (plotting port 4000/tcp). "seislog" and "srv" are transferred from the development system to the embedded system with FTP.

```
"seislog" = /mnt/mmc/SLG/PRO
"srv"     = /mnt/mmc/SLG/PRO
```

Executing seislog:

Before actually starting SEISLOG one should check the serial port that will receive the data from the digitiser to see if it is opened for reading. In this case we are using the port ttyAM1 which corresponds to COM01 in the parameter file above.

NB: ttyAM1 is not a normal naming convention in Linux. A variable SYSTEM in the include file is used to indicate normal/non-normal naming convention. This variable has to be modified accordingly before compiling the SEISLOG application.

```
#ls -l /dev/ttyAM1
```

```
#cat /dev/ttyAM1
```

```
#cd /root/SLG/PRO
#./seislog -tag -d DIG_ED3 -c CHA_ED3 -t /root
```

Funcionality of CRON:

One may create a textfile with the name of the user running SEISLOG. In this case "root". The file must be stored in /var/spool/cron/crontabs/ with the correct parameters to start SEISLOG. The example below will try to start SEISLOG every minute. Then the crond daemon must be started:

```
$crond start
```

List the textfile:

```
$ cat /var/spool/cron/crontabs/root  
* * * * * /root/SLG/PRO/seislog -tag -d DIG_XXX -c CHA_XXX -t /root
```

In the case of the EarthData digitiser, the time is included in the data stream.

4.1.4 Test results.

DIG ED3/CHA ED3	ACME/FOX	TS-7260	GUMSTIX
Seislog 24hrs	-working continuously -65 events recorded .../evt/* -Events same as TS-7260	-working continuously -66 events recorded ... /evt/* -Events same as ACME/FOX	-stopped working -26 events recorded .../evt/* -Events different from the other two systems
Seislog 72hrs	-Problems with the serial port /dev/ttyS0, no recording of data.	-working continuously -197 events recorded ... /evt/*	-stopped working -SEISLOG halted -1 event recorded
Stability	-System medium stable	-System stable	-System unstable
Observations	-Serial port /dev/ttyS0 with problems, when activating SEISLOG on /dev/ttyS2.	-The system was 100% stable running SEISLOG in all tests so far. Most recorded events.	-This system was very unstable running SEISLOG, using bothMMC and Compact Flash.

NB:

Since the Gumstix system was very unstable and the power consumption was rather high, it was excluded for further testing.

4.2 Embedded c/ Digitiser "SADC10"

Digitiser with 4 channels, 16 bit, with GPS.
Pin-Out of the serial port used for the digitiser data.

SADC10		ACME/FOX		TS-7260	
2	TX	3	RX	2	RX
3	RX	2	TX	3	TX
-	+12v	-	-	-	-
-	GND	5	GND	5	GND

In the following some details of the necessary setup for the Seislog Linux for each system:

4.2.1 "ACME/FOX"

Mount the USB memory:

```
#mount -t vfat /dev/sda1 /mnt/1
```

One has to create a directory structure and the following files. All must be done as root:

```
/mnt/flash/SLG/
    /COM
    /LOG
    /PAR
    /PRO
    /RNG
    /CAL
```

/mnt/flash/SLG/PAR/syspar

```
Events.....:/mnt/1/SLG/EVT
Events S-files.....:/mnt/1/SLG/EVT
Ringbuffers.....:/mnt/1/SLG/RNG
Ringbuffers S-files:/mnt/1/SLG/RNG
```

/mnt/flash/SLG/PAR/CHA_SAD1

```
NETWORK NAME:
UIB
TRIGGER PARAMETERS:
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
    30          40                20          3          500
CHANNEL PARAMETERS:
SEQ DIG CH MBUF FSIZ DAY  STAT COMP LOC  F-LOW F-HIGH  STA  LTA TRIG DTRIG DSK REC
  1  1  1   25   20  10  BERG BHZ  00   2.0   8.0   4.0 400.0  2.5  1.5  1  1
  2  1  2   25   20  10  BERG BHN  00   2.0   8.0   4.0 400.0  2.5  1.5  1  1
  3  1  3   25   20  10  BERG BHE  00   2.0   8.0   4.0 400.0  2.5  1.5  1  1
*****
```

/mnt/flash/SLG/PAR/DIG_SAD1

```
NO  UNIT SER_PORT BAUDRATE                IP-ADDRESS/DOMAIN SOCKET SRATE TIMING TIME-CORR
  1 SA1_S   COM01   38400                -          -    50      2      0.0
*****
```

```
"seislog" = /mnt/flash/SLG/PRO
"srv"     = /mnt/flash/SLG/PRO
```

```
#ls -l /dev/ttyS*
```

```
#cat /dev/ttyS0
```

```
#cd /mnt/flash/SLG/PRO
```

Executing SEISLOG:

```
#stty raw</dev/ttyS0      (normally handled by SEISLOG)
```

```
#./seislog -tag -d DIG_SAD1 -c CHA_SAD1 -t /mnt/flash
```

4.2.2 "TS-7260"

Mount USB memory:

Mount the modules needed to use the USB memory:

```
/usr/bin/loadUSBModules.sh
```

```
insmod usbcore
insmod pcipool
insmod usb-ohci
insmod usb-ohci-ep93xx
insmod scsi_mod
insmod sd_mod
insmod usb-storage;
```

```
$cd /usr/bin
```

```
./loadUSBModules.sh
```

```
mount -t vfat /dev/scsi/host0/target0/lun0/part1 /mnt/cf
```

One has to create a directory structure and the following files. All must be done as root:

NB: In this case the SEISLOG executables can be installed in the internal flash memory, and the events and ringbuffers is recorded on the USB.

```
/root/SLG/
```

```
    /COM
    /LOG
    /PAR
    /PRO
    /RNG
    /CAL
```

/root/SLG/PAR/syspar

```
Events.....:/mnt/cf/SLG/EVT
Events S-files.....:/mnt/cf/SLG/EVT
Ringbuffers.....:/mnt/cf/SLG/RNG
Ringbuffers S-files:/mnt/cf/SLG/RNG
```

/mnt/cf/SLG/PAR/CHA_SAD1

```

NETWORK NAME:
UIB
TRIGGER PARAMETERS:
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
      30      40      20      3      500
CHANNEL PARAMETERS:
SEQ DIG CH MBUF FSIZ DAY  STAT COMP LOC  F-LOW F-HIGH  STA  LTA TRIG DTRIG DSK REC
  1  1  1   25   20  10  BERG BHZ  00    2.0   8.0   4.0 400.0  2.5  1.5  1  1
  2  1  2   25   20  10  BERG BHN  00    2.0   8.0   4.0 400.0  2.5  1.5  1  1
  3  1  3   25   20  10  BERG BHE  00    2.0   8.0   4.0 400.0  2.5  1.5  1  1
*****

```

/mnt/cf/SLG/PAR/DIG_SAD1

```

NO  UNIT  SER_PORT  BAUDRATE  IP-ADDRESS/DOMAIN  SOCKET  SRATE  TIMING  TIME-CORR
  1  SA1_S  COM02    38400      -             -      50     2       0.0
*****

```

```

"seislog" = /root/SLG/PRO
"srv"     = /root/SLG/PRO

```

Executing SEISLOG:

```
#ls -l /dev/ttyAM1
```

```
#cat /dev/ttyAM1
```

```
#cd /root/SLG/PRO
#./seislog -tag -d DIG_SAD1 -c CHA_SAD1 -t /root
```

Funcionalidad de CRON:

One may create a text file with the name of the user running SEISLOG. In this case "root". The file must be stored in /var/spool/cron/crontabs/ with the correct parameters to start SEISLOG. The example below will try to start SEISLOG every minute. Then the crond daemon must be started:

```
$crond start
```

List the textfile:

```
$ cat /var/spool/cron/crontabs/root
* * * * * /root/SLG/PRO/seislog -tag -d DIG_XXX -c CHA_XXX -t /root
```

In the case of the SADC10 digitiser, the time is included in the data stream.

4.2.3 Test results.

DIG_SAD1/CHA_SAD1	ACME/FOX	TS-7260
Seislog 24hr	-Not operating, not possible to read from serial port.	-working continuously -50 events recorded ... /evt/*
Stability	- not operating	-100 % stable
Observations	-Not possible to read from serial port /dev/ttyS0	- no observations

4.3 Embedded w/ Digitiser "TDT3C"

Digitiser with 3 channels, 24 bit, no GPS. In this case the "ntplib" program was used to get correct timing. This option depend on a permanent connection to Internet.

Pin-Out of the serial port used for the digitiser data.

TDT3C		ACME/FOX		TS-7260	
2	TX	3	RX	2	RX
3	-	-	-	-	-
-	+12v	-	-	-	-
-	GND	5	GND	5	GND

4.3.1 "ACME/FOX"

Mount the USB memory:

```
#mount -t vfat /dev/sda1 /mnt/1
```

One has to create a directory structure and the following files. All must be done as root:

```
/mnt/flash/SLG/
    /COM
    /LOG
    /PAR
    /PRO
    /RNG
    /CAL
```

/mnt/flash/SLG/PAR/syspar

```
Events.....:/mnt/1/SLG/EVT
Events S-files.....:/mnt/1/SLG/EVT
Ringbuffers.....:/mnt/1/SLG/RNG
Ringbuffers S-files:/mnt/1/SLG/RNG
```

/mnt/flash/SLG/PAR/CHA_TDT

```
NETWORK NAME:
UIB
```

```

TRIGGER PARAMETERS:
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
      30          40          20          3          500
CHANNEL PARAMETERS:
SEQ DIG CH MBUF FSIZ DAY  STAT COMP LOC  F-LOW F-HIGH  STA  LTA TRIG DTRIG DSK REC
  1  1  1  25  20  10  BERG BHZ  00   2.0   8.0  4.0 400.0 2.5  1.5  1  1
  2  1  2  25  20  10  BERG BHN  00   2.0   8.0  4.0 400.0 2.5  1.5  1  1
  3  1  3  25  20  10  BERG BHE  00   2.0   8.0  4.0 400.0 2.5  1.5  1  1
*****

```

/mnt/flash/SLG/PAR/DIG_TDT

```

NO  UNIT  SER_PORT  BAUDRATE  IP-ADDRESS/DOMAIN  SOCKET  SRATE  TIMING  TIME-CORR
  1  TDT_S  COM01      9600          -          -      50      2      0.0
*****

```

"seislog" = /mnt/flash/SLG/PRO

"srv" = /mnt/flash/SLG/PRO

The following programs are is executed to synchronize the time with ntpd server. In the test a Brazilian server was used:146.164.48.5.

"ntpcient" =/mnt/flash/SLG/PRO

"adjust2" =/mnt/flash/SLG/PRO

"logtime2" =/mnt/flash/SLG/PRO

"sync2" =/mnt/flash/SLG/PRO

Check permissions on serial port:

```
#ls -l /dev/ttyS*
```

Setting the serial port:

```
#stty raw</dev/ttyS0      (normally handled by SEISLOG)
```

```
#stty 9600</dev/ttyS0    (normally handled by SEISLOG)
```

```
#cat /dev/ttyS0
```

```
#cd /mnt/flash/SLG/PRO
```

Executing SEISLOG:

```
#./seislog -tag -d DIG_TDT -c CHA_TDT -t /mnt/flash -m 146.164.48.5 &
```

- With the option "&" SEISLOG is running in the background.

4.3.2 "TS-7260"

Mount USB memory:

Mount the modules needed to use the USB memory:

```
/usr/bin/loadUSBModules.sh
```

```

insmod usbcore
insmod pcipool
insmod usb-ohci
insmod usb-ohci-ep93xx
insmod scsi_mod
insmod sd_mod
insmod usb-storage;

```

```

$cd /usr/bin
$./loadUSBModules.sh

```



```
mount -t vfat /dev/scsi/host0/target0/lun0/part1 /mnt/cf
```

One has to create a directory structure and the following files. All must be done as root:

NB: In this case the SEISLOG executables can be installed in the internal flash memory, and the events and ringbuffers is recorded on the USB.

```
/root/SLG/
```

```
  /COM
  /LOG
  /PAR
  /PRO
  /RNG
  /CAL
```

/root/SLG/PAR/syspar

```
Events.....:/mnt/cf/SLG/EVT
Events S-files.....:/mnt/cf/SLG/EVT
Ringbuffers.....:/mnt/cf/SLG/RNG
Ringbuffers S-files:/mnt/cf/SLG/RNG
```

/mnt/cf/SLG/PAR/CHA_TDT

```
NETWORK NAME:
```

```
UIB
```

```
TRIGGER PARAMETERS:
```

```
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
      30          40                20          3          500
```

```
CHANNEL PARAMETERS:
```

```
SEQ DIG CH MBUF FSIZ DAY  STAT COMP LOC  F-LOW F-HIGH  STA  LTA TRIG DTRIG DSK REC
  1  1  1  25  20  10  BERG BHZ  00    2.0   8.0   4.0 400.0  2.5  1.5  1  1
  2  1  2  25  20  10  BERG BHN  00    2.0   8.0   4.0 400.0  2.5  1.5  1  1
  3  1  3  25  20  10  BERG BHE  00    2.0   8.0   4.0 400.0  2.5  1.5  1  1
*****
```

/mnt/cf/SLG/PAR/DIG_TDT

```
NO  UNIT SER_PORT BAUDRATE          IP-ADDRESS/DOMAIN SOCKET SRATE TIMING TIME-CORR
  1  TDT_S  COM02    9600                -      -    50     2     0.0
*****
```

```
"seislog" = /root/SLG/PRO
```

```
"srv" = /root/SLG/PRO
```

Executing SEISLOG:

```
#ls -l /dev/ttyAM1
```

```
#cat /dev/ttyAM1
```

```
#cd /root/SLG/PRO
```

```
#./seislog -tag -d DIG_SAD1 -c CHA_SAD1 -t /root
```

Funcionalidad de CRON:

One may create a text file with the name of the user running SEISLOG. In this case "root". The file must be stored in /var/spool/cron/crontabs/ with the correct parameters to start SEISLOG. The example below will try to start SEISLOG every minute. Then the crond daemon must be started:

\$crond start

List the text file:

```
$ cat /var/spool/cron/crontabs/root  
* * * * * /root/SLG/PRO/seislog -tag -d DIG_XXX -c CHA_XXX -t /root
```

In the case of the digitiser "TDT3C", the time will be synchronized with the "NTP" protocol, which is still under testing for the TS-7260 processor.

4.3.3 Test results.

DIG_TDT/CHA_TDT	ACME/FOX	TS-7260
Seislog 96hr	-System working perfect with automatic startup.	-working continuously -No interruptions working with the crond deamon.
Stability	-System stable	-System stable
Observations	-In these last tests we have discovered that the serial port buffer fills up and then block for more incoming data. If however, the data cable is connected just before SEISLOG starts, the buffer will not fill up before SEISLOG starts reading.	-No observations.

5. Conclusions

After testing the three systems and considering only stability and functionality with different digitisers, the TS-7260 is the clear number one. This is also the conclusion considering that the systems to be designed are mainly to be designed for use as automatic field stations.

Power consumption:

According to the tests, the Gumstix system was excluded from further testing due to rather high power consumption (700mA) depending on type of memory. This system requires all together 3 different board to be compatible with the other two systems tested. The total power consumption was around the double of the other two.

In the case of the TS-7260 and ACME/FOX, the power consumption is very similar for both systems, around 300mA even down to 190mA on the TS-7260.

The ACME/FOX require a power supply with very low noise to function well on different peripherals, and can only use 5VDC. The TS-7260 permits a power supply from 4.5-20VDC and is not sensible to noise as for the ACME/FOX.

In the case that the systems will be powered by 12VDC solar panels it is no need for a DC/DC converter for the TS-7260. For the ACME/FOX this is necessary.

Stability:

Concerning stability, the TS-7260 showed to be the most stable compared to the other two systems. Running SEISLOG with the crond daemon it did not stop during any of the tests. The server srv also worked continuously sending data to the LISSPLOT client. File transfers of events did not interrupt the data-acquisition.

In the case of the ACME/FOX, once the system is running correctly (reading the serial port) it works rather well. The problem seems to be that the RS-232 buffer fill up and then block the input of data. To make the system start correctly one has to connect the data cable in right moment when SEISLOG starts to read data. This is to avoid a data over-run in the buffer. See paragraph 6 for further testing in Bergen.

File transfers from the ACME/FOX can be done with SCP or FTP, but produce problems when you want to transfer high volumes of data. There are no problems transferring for example 1-3 events, however with a high number of events, SEISLOG halts and the system has to be restarted. An alternative way of operation is to stop the SEISLOG application, change the USB memory and restart the system.

We assume the reason for this problem is that SEISLOG and SRV plus system utilities like FTP is on the limit of what the ACME/FOX can handle with the current recourses (memory, CPU speed).

Startup:

TS-7260:

Startup of SEISLOG can be done with the crond daemon which is part of the standard kernel. Described in the tests above.

ACME/FOX:

Startup can be done through the RunLevels as defined under Linux.

Digitisers:

The digitisers used in the tests were: EarthData PS-2400, SADC10 and TDT3C.

Depending on what system you will run SEISLOG, the executable files must be created on a separate PC with a cross-compiler.

Events and ringbuffers are recorded in MiniSeed format, Steim1 compression. To process data with SEISAN you need the correct patch that support this format.

The SEISLOG Linux also supports other digitisers such as EarthData 6CH, GVB-316, SADC20,SADC30, Kinometrics Makalu

GPS v/s NTP:

In the case that the digitiser includes a GPS with the correct time recorded in the data stream, the NTP has no importance.

In the case that the digitisers do not include the time in the data stream (for example TDT3C), the time is taken from the CPU time. To correct the CPU time, a NTP-client program is used. This utility connects to a public NTP server and adjusts the CPU time.

Advantages / Disadvantages:

TS-7260:

The main advantages for this system:

- Robust in all aspects.
- Low power consumption.
- Funcionality to handle the data (events).
- Network services that permit multiple sessions without interrupting the functionality of the rest of the system.
- Does not need any extra devices or interfaces.
- CPU faster than ACME/FOX.
- Good documentation.

Disadvantages:

- The kernel does not include the service "SSH" for safe sessions, only access with "Telnet".
- Price a bit higher than ACME/FOX.

ACME/FOX:

The main advantages for this system:

- The kernel includes security tools like "iptables" and "SSH".
- Easy configuration.
- National distributor (Chile).
- Very active user group.
- Size

Disadvantages:

- Less documentation than the TS-7260.
- Instability serial ports.
- Less robust than TS-7260.
- Need an extra interface to convert the serial line to true RS232.
- Seems to be very sensitive to power supply with low noise.

Things to do:

- Automatic startup of SEISLOG
- Improve the "NTP-client" program for both ACME/FOX and TS-7260.
- Accurate testing of the timing with NTP-client with minute-pulse
- Solve the serial port problem for ACME/FOX.
- Develop an application similar to a "Firewall" (iptables), to protect the systems when they are connected to the Internet.
- Modify SEISLOG to recognize the /dev/ttyAM1. Standard is ttyS0.

6. Further testing in Bergen after visit to Chile

A digitiser of type SADC24 bit with the ACME/FOX integrated was supplied from SARA to do further testing with the ACME/FOX. This unit has a good regulated power supply. After two weeks of testing we have not seen any of the problems with the serial lines as we saw in Chile. We assume this has to do with the power supply.

With this dramatic change in behaviour, we have decided to include the ACME/FOX processor as a candidate together with the TS-7260 for implementation of SEISLOG Linux for embedded systems.

It is possible that the two systems can serve different purposes: The ACME/FOX as an integrated part of a field recorder, and the TS-7260 as part of a field recorder, but also as a permanent field station.

This will require software support and documentation for both systems.



Departamento de Geofísica
Facultad de Ciencias Físicas y Matemáticas
Universidad de Chile



Institute of Geoscience
University of Bergen, Norway

SEISLOG – Linux Sistemas embedded

Terje Utheim

Department of Earth Science/University of Bergen
terje.utheim@geo.uib.no

Ulises Vásquez Casanova

Departamento de Geofísica/Universidad de Chile
ulises@dgf.uchile.cl

Febrero 2006

Índice.

	Página
Introducción	3
1. Objetivos	3
2. Necesidad de Implementación	4
3. Tipos de Sistemas Embedded	5
4. Test de Laboratorio	7
4.1 Embedded c/ Digitalizador "EarthData 3CH 24bit"	7
4.1.1 "ACME/FOX"	7
4.1.2 "GUMSTIX"	9
4.1.3 "TS-7260"	10
4.1.4 Resultados de las pruebas	12
4.2 Embedded c/ Digitalizador "SADC10-16bit"	12
4.2.1 "ACME/FOX"	12
4.2.2 "TS-7260"	13
4.2.3 Resultados de las pruebas	15
4.3 Embedded c/ Digitalizador "TDT3C"	15
4.3.1 "ACME/FOX"	16
4.3.2 "TS-7260"	17
4.3.3 Resultados de las pruebas	18
5. Conclusiones	19
Apéndice 1: Configuración ACME/FOX	22
Apéndice 2: Configuración TS-7260	25
Apéndice 3: GLOSARIO	29

Introducción

Actualmente las estaciones sismológicas del Depto. De Geofísica de la U. de Chile han estado funcionando con OS QNX en tiempo real (distribución unix comercial), esto ha permitido un avance significativo con respecto a los sistemas que estaban funcionando previamente. Esto no sólo por el hecho de un sistema de registro de menor costo con respecto a otros, sino que además flexible y adaptado a los requerimientos de cada uno de los proyectos de investigación sismológicos.

Como todo sistema, esta sujeto a la evolución tecnológica y a la discontinuidad por nuevas versiones de software. En este sentido, QNX al ser un sistema comercial con licencia, limita las posibilidades de poder realizar nuevos drivers (controladores de dispositivos como; digitalizadores, GPSs principalmente), con otras versiones de QNX, dejando amarradas las necesidades a cierto tipo de HW, ya que el manejo de dispositivos y periféricos de QNX es limitada.

Es aquí donde Linux aparece como una alternativa viable para poder implementar el sistema de registro, que actualmente esta siendo empleado por la mayoría de las estaciones sismológicas instaladas en el país, como lo es "Seislog".

Las principales motivaciones para usar Linux en estaciones sismológicas son;

- Software de libre distribución.
- Código fuente abierto.
- Existencia de una amplia gama de versiones para un mismo núcleo.
- Soporte para manejo de dispositivos y periféricos amplio.
- Posibilidades de actualizaciones.
- Sistema operativo Standard en "Embedded".

1. Objetivos

- Determinar el sistema *embedded* más adecuado para la implementación de seislog-linux en estaciones sismológicas.

- Realizar pruebas de laboratorio que permitan establecer las principales ventajas y desventajas de un sistema u otro.

- Establecer el procedimiento de configuración, instalación y funcionamiento de cada sistema embedded, con los digitalizadores "TDT3C 24bit", "EarthData 3CH 24bit" y "SADC10 16bit"

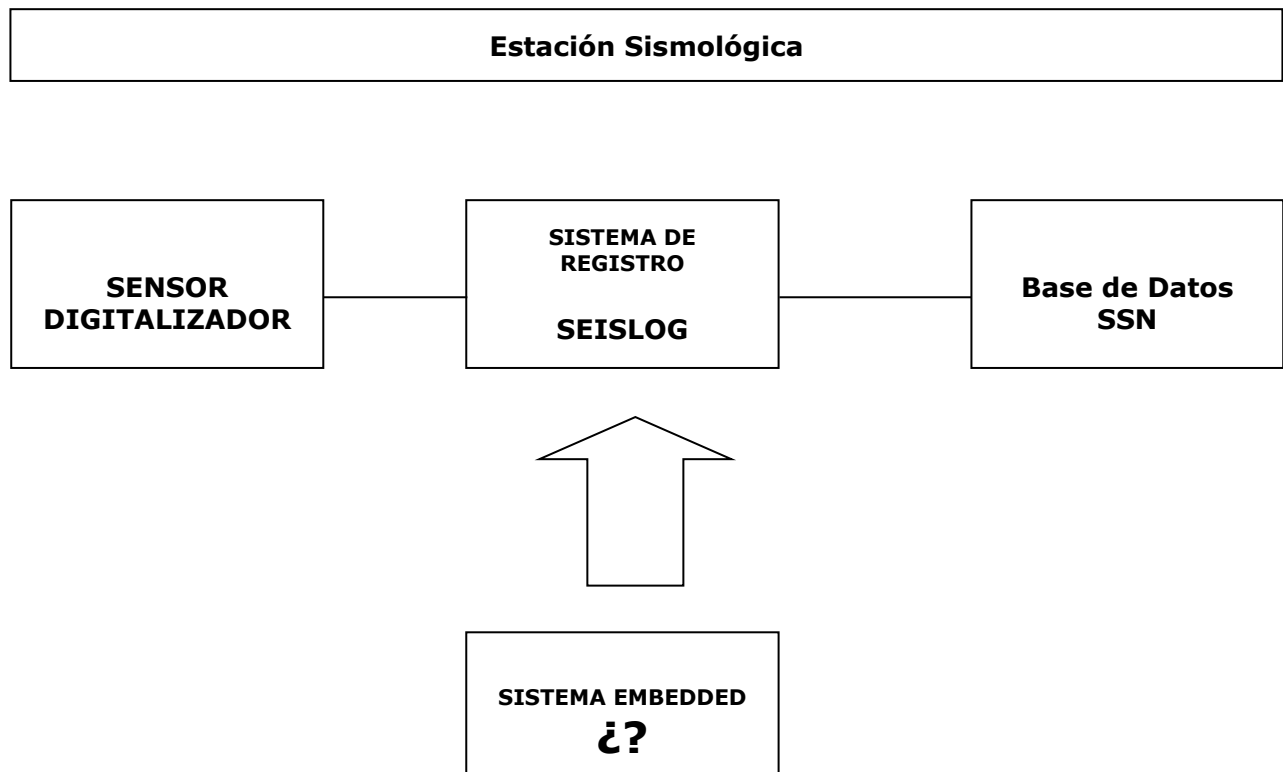
2. Necesidad de Implementación

Actualmente las estaciones sismológicas del Departamento de Geofísica de la U. de Chile, están funcionando con OS QNX, sistema operativo de 1999 con licencia. La evolución y desarrollo de tecnología de HW para PCs han hecho que este sistema operativo sea cada vez más restringido a su uso con cierto tipo de HW y que con el tiempo, como todo sistema, quedara sujeto a la obsolescencia.

Por lo anterior, la migración de "Seislog" desde OS.QNX a OS. Linux de libre distribución, tanto a nivel de PCs y de sistemas *embedded*, aparece como una poderosa herramienta de recambio tecnológico, dado su funcionalidad y flexibilidad en la adaptación de una amplia variedad de HW.

Actualmente existen estaciones sismológicas funcionando con el sistema *embedded* "PC/104-Roadrunner II", sin embargo el alto costo en comparación con los nuevos sistemas y alto consumo de energía, hacen necesario un recambio en este sentido.

El siguiente diagrama muestra el objetivo de implementar estos sistemas dentro de una estación sismológica.



3. Tipos de Sistemas Embedded

Actualmente en el mercado existen una gran variedad de dispositivos *embedded* con un amplio rango de características técnicas y de costos, y en su mayoría están orientados al las funciones de control y automatización, empleando como sistema base LINUX.

Esta es una de las principales características que hacen posible poder lograr adaptar estos sistemas las necesidades y aplicaciones del usuario.

En el caso para la implementación de estaciones sismológicas, estos sistemas en conjunto, con *Seislog-Linux* aparecen como una herramienta poderosa de detección y captura de datos sísmicos.

Los sistemas *embedded* escogidos para este propósito son:

- ACME/FOX
- GUMSTIX
- TS-7260

Cada uno de estos sistemas, en general, están compuestos de periféricos y un sistema operativo compacto, que permite correr programas y servicios compilados externamente, es decir, en un PC y posteriormente los archivos ejecutables son introducidos en la memoria del sistema, dado que la capacidad de memoria flash es mínima, característico de estos sistemas.

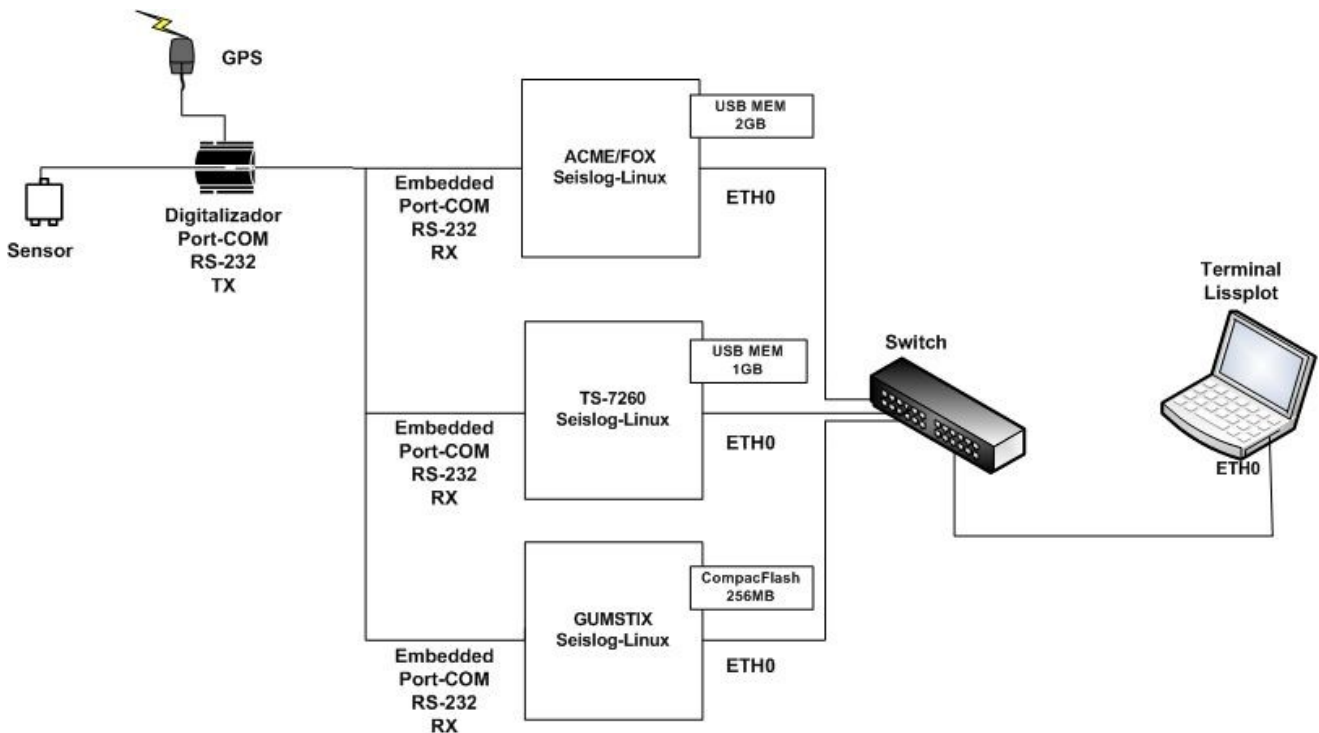
En orden a mostrar las principales características, ventajas y desventajas de uno y otro, a continuación se presenta una tabla con los principales criterios o parámetros para poder decidir el sistema más adecuado para la implementación de una estación sismológica con "Seislog".

Cuadro Comparativo Sistemas Embedded para SEISLOG/LINUX

PARAMETROS	ACME/FOX	GUMSTIX	TS-7260
OS Linux	Linux Kernel 2.6.12	Linux Kernel 2.6.11	Linux Kernel 2..4.26
Servicios de Red y Paquetes relevantes	-FTP Server -SSH Server (Solo Terminal) -SCP -TELNET Server -PPpd -GCC externo -HTTpd	-SCP -SSH server (Solo Terminal) -GCC externo -IPtables -HTTpd	-Telnetd -Ftpd -Crond -GCCext
Periféricos	-1xCOM -ETH0 10/100baseT -2xUSB 1.1 (pendrive hasta 2GBx1port) -2 extensiones IDE,SCSI	-COM1 (Consola) -COM2 -ETH0 10/100baseT -Socket Memoria Cflasf o MMC -1xUSB 1.1 (Terminal Serv)	-ETH0 10/100BaseT -3xCOM -2xUSB -Socket CompacFlash
Memoria	-Flash OS=4MB -Memoria RAM=16MB -Externa=PenDrive	-Flash OS=4MB -Memoria RAM=64MB -Externa MMC/CF hasta 512MB	-Flash OS=8MB de 32MB -Memoria RAM=32/64MB
Procesador	AXIS 100MHz	Intel XScale 400 MHz	CPU 200MHz
Consumo	5VDC, 12VDC 260mA=ETH0,seislog 300mA=ETH0,seislog,usb 270mA=seislog,usb	4,5-5VDC Modelo CF: 500mA+200mA Modelo MMC: 400mA	Power In=4,5-20vdc 5VDC/ 220mA=usb,seislog 190mA=ETH0,usb 300mA=ETH0,seislog,usb
Procedencia	ITALIA www.acmesystems.it	CANADA www.gumstix.com	USA www.embeddedarm.com
Dimensiones	6,6x7,2x2cm	11x3,5x1,5cm	10x13cm
COSTO	EUR \$154.80(Italia)	Kit completo U\$216.00	U\$179.00 (RAM=32MB) U\$195.00 (RAM=64MB)

4. Test de Laboratorio

Las pruebas realizadas a cada sistema embedded, se muestran en la siguiente figura:



A cada sistema se le ingresó una señal en paralelo del digitalizador. Monitoreado en red con un servidor de Terminal ya sea vía telnet o ssh y corriendo "Lissplot" para plotear las trazas por canales independientes.

4.1 Embedded c/ Digitalizador "EarthData 3CH 24bit"

Pin-Out de las puertas seriales de captura de datos.

EarthData		ACME/FOX		TS-7260		GUMSTIX	
2	TX	3	RX	2	RX	3	RX
3	RX	-	-	-	-	-	-
4	+12v	-	-	-	-	-	-
5	GND	5	GND	5	GND	5	GND

A continuación se detallan los archivos de configuración necesarios para el seteo de Seislog-Linux:

4.1.1 "ACME/FOX"

Montar Memoria USB:

```
#mount -t vfat /dev/sda1 /mnt/1
```

Se debe crear una estructura de directorios y transferir los archivos siguientes, TODO COMO ROOT:

```

/mnt/flash/SLG/
    /COM
    /LOG
    /PAR
    /PRO
    /RNG
    /CAL

```

/mnt/flash/SLG/PAR/syspar

```

Events.....:/mnt/1/SLG/EVT
Events S-files.....:/mnt/1/SLG/EVT
Ringbuffers.....:/mnt/1/SLG/RNG
Ringbuffers S-files:/mnt/1/SLG/RNG

```

/mnt/flash/SLG/PAR/CHA_ED3

NETWORK NAME:

UIB

TRIGGER PARAMETERS:

PRE-EVENT	POST-EVENT	ARRAY-PROPAGATION	MIN-TRIGGERS	MAX-RECORDING
30	40	20	3	500

CHANNEL PARAMETERS:

SEQ	DIG	CH	MBUF	FSIZ	DAY	STAT	COMP	LOC	F-LOW	F-HIGH	STA	LTA	TRIG	DTRIG	DSK	REC
1	1	1	25	20	10	BERG	BHZ	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
2	1	2	25	20	10	BERG	BHN	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
3	1	3	25	20	10	BERG	BHE	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1

/mnt/flash/SLG/PAR/DIG_ED3

NO	UNIT	SER	PORT	BAUDRATE	IP-ADDRESS/DOMAIN	SOCKET	SRATE	TIMING	TIME-CORR
1	ED3_S	COM01		19200		-	50	2	0.0

Los programas ejecutables para correr el sistema deben estar en las rutas descritas a continuación. "seislog", es el programa compilado con los drivers y "srv", es el servidor para plotear las trazas desde un cliente con java a través del puerto 4000/tcp. La transferencia de estos programas se debe realizar mediante el protocolo SCP o FTP:

```

"seislog" = /mnt/flash/SLG/PRO
"srv"     = /mnt/flash/SLG/PRO

```

Ejecutando seislog:

Previamente se debe chequear que la puerta serial este habilitada recibiendo datos y posteriormente ejecutar seislog direccionando los canales y tipo de digitalizador.

En este caso se esta utilizando el primer puerto serial "ttyS0" que corresponde la convención "COM1"

```
#ls -l /dev/ttyS*
```

```

crw-rw-rw- 1 root   tty      4,  64 Oct 24 2005 /dev/ttyS0
crw-rw-rw- 1 root   tty      4,  65 Oct 24 2005 /dev/ttyS1
crw-rw-rw- 1 root   tty      4,  66 Oct 24 2005 /dev/ttyS2
crw-rw-rw- 1 root   tty      4,  67 Oct 24 2005 /dev/ttyS3

```

```
#cat /dev/ttyS0
```

```
#cd /mnt/flash/SLG/PRO
```

```
#stty raw</dev/ttyS0
```

```
#./seislog -tag -d DIG_ED3 -c CHA_ED3 -t /mnt/flash
```

En el caso del digitalizador EarthData, el tiempo viene incluido en la señal de datos que entrega el puerto serial y por tanto es independiente del clock de la CPU.

4.1.2 "GUMSTIX"

Montaje de la Memoria MMC:

Para el montaje de la memoria MMC, en gumstix se debe hacer una modificación en el archivo /etc/profile :

Descomentarizar #mmc-block (esto para que el montaje sea automático en el arranque)

Para montar la tarjeta MMC:

```
#mount /dev/mmc/mm0/part1 /mnt/mmc
```

Al igual que el sistema anterior, se debe crear una estructura de directorios y transferir los archivos siguientes, TODO COMO ROOT:

```
/mnt/mmc/SLG/  
    /COM  
    /LOG  
    /PAR  
    /PRO  
    /RNG  
    /CAL
```

/mnt/mmc/SLG/PAR/syspar

```
Events.....:/mnt/mmc/SLG/EVT  
Events S-files.....:/mnt/mmc/SLG/EVT  
Ringbuffers.....:/mnt/mmc/SLG/RNG  
Ringbuffers S-files:/mnt/mmc/SLG/RNG
```

/mnt/mmc/SLG/PAR/CHA_ED3

NETWORK NAME:

UIB

TRIGGER PARAMETERS:

```
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING  
    30          40                20          3          500
```

CHANNEL PARAMETERS:

SEQ	DIG	CH	MBUF	FSIZ	DAY	STAT	COMP	LOC	F-LOW	F-HIGH	STA	LTA	TRIG	DTRIG	DSK	REC
1	1	1	25	20	10	BERG	BHZ	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
2	1	2	25	20	10	BERG	BHN	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
3	1	3	25	20	10	BERG	BHE	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1

/mnt/mmc/SLG/PAR/DIG_ED3

NO	UNIT	SER	PORT	BAUDRATE	IP-ADDRESS/DOMAIN	SOCKET	SRATE	TIMING	TIME-CORR
1	ED3_S	COM03		19200		-	50	2	0.0

Los programas ejecutables para correr el sistema deben estar en las rutas descritas a continuación. "seislog", es el programa compilado con los drivers y "srv", es el servidor para plotear las trazas desde un cliente con java a través del puerto 4000/tcp. La transferencia de estos programas se debe realizar mediante el protocolo SCP unicamente:

"seislog" = /mnt/mmc/SLG/PRO

"srv" = /mnt/mmc/SLG/PRO

Ejecutando seislog:

Previamente se debe chequear que la puerta serial esté habilitada recibiendo datos y posteriormente ejecutar seislog direccionando los canales y tipo de digitalizador.

En este caso se está utilizando el primer puerto serial "ttyS2" que corresponde a la convención "COM3"

```
#ls -l /dev/ttyS*
```

```
crw-rw-rw-  1 root    tty      4,  64 Oct 24  2005 /dev/ttyS0
crw-rw-rw-  1 root    tty      4,  65 Oct 24  2005 /dev/ttyS1
crw-rw-rw-  1 root    tty      4,  66 Oct 24  2005 /dev/ttyS2
crw-rw-rw-  1 root    tty      4,  67 Oct 24  2005 /dev/ttyS3
```

```
#cat /dev/ttyS2
```

```
#cd /mnt/mmc/SLG/PRO
```

```
#./seislog -tag -d DIG_ED3 -c CHA_ED3 -t /mnt/mmc
```

4.1.3 "TS-7260"

Montar Memoria USB:

Carga de módulos para uso de memoria USB:

```
/usr/bin/loadUSBModules.sh
```

```
insmod usbcore
insmod pcipool
insmod usb-ohci
insmod usb-ohci-ep93xx
insmod scsi_mod
insmod sd_mod
insmod usb-storage;
```

```
$cd /usr/bin
```

```
./loadUSBModules.sh
```

```
mount -t vfat /dev/scsi/host0/target0/lun0/part1 /mnt/cf
```

Se debe crear una estructura de directorios y transferir los archivos siguientes, TODO COMO ROOT:

NOTA: En este caso se debe instalar los programas ejecutables de seislog en la flash interna del sistema, dado que la memoria lo permite, y solo los eventos y ringbuffer en la memoria USB.

```
/root/SLG/
```

```
  /COM
  /LOG
  /PAR
  /PRO
  /RNG
  /CAL
```

/root/SLG/PAR/syspar

```
Events.....:/mnt/cf/SLG/EVT
Events S-files.....:/mnt/cf/SLG/EVT
Ringbuffers.....:/mnt/cf/SLG/RNG
Ringbuffers S-files:/mnt/cf/SLG/RNG
```

/mnt/cf/SLG/PAR/CHA_ED3

NETWORK NAME:

UIB

TRIGGER PARAMETERS:

```
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
      30          40                20          3          500
```

CHANNEL PARAMETERS:

```
SEQ DIG CH MBUF FSIZ DAY STAT COMP LOC F-LOW F-HIGH STA LTA TRIG DTRIG DSK REC
  1  1  1  25  20  10 BERG BHZ 00  2.0  8.0  4.0 400.0 2.5  1.5  1  1
  2  1  2  25  20  10 BERG BHN 00  2.0  8.0  4.0 400.0 2.5  1.5  1  1
  3  1  3  25  20  10 BERG BHE 00  2.0  8.0  4.0 400.0 2.5  1.5  1  1
*****
```

/mnt/cf/SLG/PAR/DIG_ED3

```
NO UNIT SER_PORT BAUDRATE IP-ADDRESS/DOMAIN SOCKET SRATE TIMING TIME-CORR
  1 ED3_S COM02 19200 - - 50 2 0.0
*****
```

Los programas de ejecutables de para hacer correr el sistema deben estar en las rutas descritas a continuación. "seislog", es el programa compilado con los drivers y "srv", es el servidor para plotear las trazas desde un cliente con java a través del puerto 4000/tcp. La transferencia de estos programas se debe realizar mediante el protocolo FTP:

"seislog" = /root/SLG/PRO

"srv" = /root/SLG/PRO

Ejecutando seislog:

Previamente se debe chequear que la puerta serial esté habilitada recibiendo datos y posteriormente ejecutar seislog direccionando los canales y tipo de digitalizador.

En este caso se está utilizando el primer puerto serial "ttyAM1" que corresponde la convención "COM1"

```
#ls -l /dev/ttyAM1
```

```
#cat /dev/ttyAM1
```

```
#cd /root/SLG/PRO
```

```
#!/seislog -tag -d DIG_ED3 -c CHA_ED3 -t /root
```

Funcionalidad del "CRON":

Se debe crear un archivo de texto con el nombre del usuario donde se corre el seislog, en este caso "root" en /var/spool/cron/crontabs/ con los parámetros de arranque de seislog.

Luego correr el cron:

```
$crond start
```

```
$ cat /var/spool/cron/crontabs/root
```

```
* * * * * /root/SLG/PRO/seislog -tag -d DIG_XXX -c CHA_XXX -t /root
```

En el caso del digitalizador EarthData, el tiempo viene incluido en la señal de datos que entrega el puerto serial y por tanto es independiente del clock de la CPU.

4.1.4 Resultados de las pruebas.

DIG_ED3/CHA_ED3	ACME/FOX	TS-7260	GUMSTIX
Seislog en 24hrs	-Funcionamiento continuo -65 archivos registrados .../evt/* -Eventos iguales a TS-7260	-Funcionamiento continuo -66 archivos registrados .../evt/* -Eventos iguales a ACME/FOX	-Funcionamiento interrumpido -26 archivos registrados .../evt/* -Eventos Distintos a los otros dos sistemas
Seislog en 72hrs	-Problemas con la puerta serial /dev/ttyS0, sin captura de datos.	-Funcionamiento continuo -197 archivos registrados en .../evt/*	-Funcionamiento interrumpido -Proceso seislog detenido -1 archivo registrado
Estabilidad	-Sistema medianamente estable	-Sistema estable	-Sistema inestable
Observaciones	-Puerta serial /dev/ttyS0 con problemas, en proceso de activación puerta /dev/ttyS2.	-El sistema se mantuvo corriendo seislog en todas las pruebas hasta aquí realizadas, siendo el sistema con 100% de estabilidad y mayor cantidad de registros.	-Este sistema se mantuvo muy inestable mientras se hizo correr seislog, esto tanto a nivel del uso de CompacFlash y a nivel de MMC.

NOTA:

Dado que "gumstix" presentó problemas de inestabilidad y alto consumo de energía, quedó descartado para las siguientes pruebas.

4.2 Embedded c/ Digitalizador "SADC10"

Digitalizador de 4 canales, 16 bit, con GPS.

Pin-Out de las puertas seriales de captura de datos.

SADC10		ACME/FOX		TS-7260	
2	TX	3	RX	2	RX
3	RX	2	TX	3	TX
-	+12v	-	-	-	-
-	GND	5	GND	5	GND

A continuación se detallan los archivos de configuración necesarios para el seteo de Seislog-Linux:

4.2.1 "ACME/FOX"

Montar Memoria USB:

```
#mount -t vfat /dev/sda1 /mnt/1
```

Se debe crear una estructura de directorios y transferir los archivos siguientes, TODO COMO ROOT:

```
/mnt/flash/SLG/  
    /COM  
    /LOG  
    /PAR  
    /PRO  
    /RNG  
    /CAL
```

/mnt/flash/SLG/PAR/syspar

```
Events.....:/mnt/1/SLG/EVT  
Events S-files.....:/mnt/1/SLG/EVT  
Ringbuffers.....:/mnt/1/SLG/RNG  
Ringbuffers S-files:/mnt/1/SLG/RNG
```

/mnt/flash/SLG/PAR/CHA_SAD1

NETWORK NAME:

UIB

TRIGGER PARAMETERS:

PRE-EVENT	POST-EVENT	ARRAY-PROPAGATION	MIN-TRIGGERS	MAX-RECORDING
30	40	20	3	500

CHANNEL PARAMETERS:

SEQ	DIG	CH	MBUF	FSIZ	DAY	STAT	COMP	LOC	F-LOW	F-HIGH	STA	LTA	TRIG	DTRIG	DSK	REC
1	1	1	25	20	10	BERG	BHZ	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
2	1	2	25	20	10	BERG	BHN	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
3	1	3	25	20	10	BERG	BHE	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1

/mnt/flash/SLG/PAR/DIG_SAD1

NO	UNIT	SER_PORT	BAUDRATE	IP-ADDRESS/DOMAIN	SOCKET	SRATE	TIMING	TIME-CORR
1	SA1_S	COM01	38400	-	-	50	2	0.0

"seislog" = /mnt/flash/SLG/PRO

"srv" = /mnt/flash/SLG/PRO

#ls -l /dev/ttyS*

#cat /dev/ttyS0

#cd /mnt/flash/SLG/PRO

Ejecutando Seislog:

#stty raw</dev/ttyS0

#!/seislog -tag -d DIG_SAD1 -c CHA_SAD1 -t /mnt/flash

4.2.2 "TS-7260"

Montar Memoria USB:

Carga de módulos para uso de memoria USB:

/usr/bin/loadUSBModules.sh

insmod usbcore

insmod pcipool

```
insmod usb-ohci
insmod usb-ohci-ep93xx
insmod scsi_mod
insmod sd_mod
insmod usb-storage;
```

```
$cd /usr/bin
$./loadUSBModules.sh
mount -t vfat /dev/scsi/host0/target0/lun0/part1 /mnt/cf
```

Se debe crear una estructura de directorios y transferir los archivos siguientes, TODO COMO ROOT:

NOTA: En este caso se debe instalar los programas ejecutables de seislog en la flash interna del sistema, dado que la memoria lo permite, sólo los eventos y ringbuffer se registran en la memoria USB.

```
/root/SLG/
  /COM
  /LOG
  /PAR
  /PRO
  /RNG
  /CAL
```

/root/SLG/PAR/syspar

```
Events.....:/mnt/cf/SLG/EVT
Events S-files.....:/mnt/cf/SLG/EVT
Ringbuffers.....:/mnt/cf/SLG/RNG
Ringbuffers S-files:/mnt/cf/SLG/RNG
```

/mnt/cf/SLG/PAR/CHA_SAD1

NETWORK NAME:

UIB

TRIGGER PARAMETERS:

```
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
      30      40      20      3      500
```

CHANNEL PARAMETERS:

SEQ	DIG	CH	MBUF	FSIZ	DAY	STAT	COMP	LOC	F-LOW	F-HIGH	STA	LTA	TRIG	DTRIG	DSK	REC
1	1	1	25	20	10	BERG	BHZ	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
2	1	2	25	20	10	BERG	BHN	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
3	1	3	25	20	10	BERG	BHE	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1

/mnt/cf/SLG/PAR/DIG_SAD1

NO	UNIT	SER	PORT	BAUDRATE	IP-ADDRESS/DOMAIN	SOCKET	SRATE	TIMING	TIME-CORR
1	SA1_S	COM02	38400	-	-	50	2	0.0	

"seislog" = /root/SLG/PRO

"srv" = /root/SLG/PRO

Ejecutando seislog:

```
#ls -l /dev/ttyAM1
```

```
#cat /dev/ttyAM1
```

```
#cd /root/SLG/PRO
```

```
#./seislog -tag -d DIG_SAD1 -c CHA_SAD1 -t /root
```

Funcionalidad del "CRON":

Se debe crear un archivo de texto con el nombre del usuario donde se corre el seislog, en este caso "root" en /var/spool/cron/crontabs/ con los parámetros de arranque de seislog.

Luego correr el cron:

```
$crond start
$ cat /var/spool/cron/crontabs/root
* * * * * /root/SLG/PRO/seislog -tag -d DIG_XXX -c CHA_XXX -t /root
```

En el caso del digitalizador "SADC10", el tiempo viene incluido en la señal de datos que entrega el puerto serial y por tanto es independiente del clock de la CPU.

4.2.3 Resultados de las pruebas.

DIG_SAD1/CHA_SAD1	ACME/FOX	TS-7260
Seislog en 24hrs	-Sin operación, en proceso de habilitar puerta serial.	-Funcionamiento continuo -50 archivos registrados .../evt/*
Estabilidad	- s/f	-Sistema muy estable
Observaciones	-Puerta serial /dev/ttyS0 con problemas, en acceso a la velocidad baud rate.	- s/obs

4.3 Embedded c/ Digitalizador "TDT3"

Digitalizador de 3 canales, 24 bit, sin GPS. En este caso se emplea la sincronización por protocolo de tiempo, a través de "ntplib"

Pin-Out de las puertas seriales de captura de datos.

TDT3C		ACME/FOX		TS-7260	
2	TX	3	RX	2	RX
3	-	-	-	-	-
-	+12v	-	-	-	-
-	GND	5	GND	5	GND

4.3.1 "ACME/FOX"

Montar Memoria USB:

```
#mount -t vfat /dev/sda1 /mnt/1
```

Se debe crear una estructura de directorios y transferir los archivos siguientes, TODO COMO ROOT:

```
/mnt/flash/SLG/  
    /COM  
    /LOG  
    /PAR  
    /PRO  
    /RNG  
    /CAL
```

/mnt/flash/SLG/PAR/syspar

```
Events.....:/mnt/1/SLG/EVT  
Events S-files.....:/mnt/1/SLG/EVT  
Ringbuffers.....:/mnt/1/SLG/RNG  
Ringbuffers S-files:/mnt/1/SLG/RNG
```

/mnt/flash/SLG/PAR/CHA_TDT

NETWORK NAME:

UIB

TRIGGER PARAMETERS:

```
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING  
    30          40                20          3          500
```

CHANNEL PARAMETERS:

SEQ	DIG	CH	MBUF	FSIZ	DAY	STAT	COMP	LOC	F-LOW	F-HIGH	STA	LTA	TRIG	DTRIG	DSK	REC
1	1	1	25	20	10	BERG	BHZ	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
2	1	2	25	20	10	BERG	BHN	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
3	1	3	25	20	10	BERG	BHE	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1

/mnt/flash/SLG/PAR/DIG_TDT

NO	UNIT	SER_PORT	BAUDRATE	IP-ADDRESS/DOMAIN	SOCKET	SRATE	TIMING	TIME-CORR
1	TDT_S	COM01	9600	-	-	50	2	0.0

"seislog" = /mnt/flash/SLG/PRO

"srv" = /mnt/flash/SLG/PRO

Los siguientes programas son para sincronizar el tiempo con un servidor "ntpd". Para el caso de análisis, se realizo con el servidor 146.164.48.5 (Brazil).

"ntpclient" =/mnt/flash/SLG/PRO

"adjust2" =/mnt/flash/SLG/PRO

"logtime2" =/mnt/flash/SLG/PRO

"sync2" =/mnt/flash/SLG/PRO

Chequeo de los permisos de la puerta serial:

```
#ls -l /dev/ttyS*
```

Habilitación de la puerta serial:

```
#stty raw</dev/ttyS0
```

```
#cat /dev/ttyS0
```

```
#cd /mnt/flash/SLG/PRO
```

Ejecutando Seislog:

```
#./seislog -tag -d DIG_TDT -c CHA_TDT -t /mnt/flash -m 146.164.48.5 &
```

- Con la opción "&" queda corriendo en background el sistema.

4.3.2 "TS-7260"

Montar Memoria USB:

Carga de módulos para uso de memoria USB:

```
/usr/bin/loadUSBModules.sh
```

```
insmod usbcore
insmod pcipool
insmod usb-ohci
insmod usb-ohci-ep93xx
insmod scsi_mod
insmod sd_mod
insmod usb-storage;
```

```
$cd /usr/bin
```

```
./loadUSBModules.sh
```

```
mount -t vfat /dev/scsi/host0/target0/lun0/part1 /mnt/cf
```

Creando estructura:

```
/root/SLG/
  /COM
  /LOG
  /PAR
  /PRO
  /RNG
  /CAL
```

/root/SLG/PAR/syspar

```
Events.....:/mnt/cf/SLG/EVT
Events S-files.....:/mnt/cf/SLG/EVT
Ringbuffers.....:/mnt/cf/SLG/RNG
Ringbuffers S-files:/mnt/cf/SLG/RNG
```

/mnt/cf/SLG/PAR/CHA_TDT

NETWORK NAME:

UIB

TRIGGER PARAMETERS:

```
PRE-EVENT POST-EVENT ARRAY-PROPAGATION MIN-TRIGGERS MAX-RECORDING
      30      40      20      3      500
```

CHANNEL PARAMETERS:

SEQ	DIG	CH	MBUF	FSIZ	DAY	STAT	COMP	LOC	F-LOW	F-HIGH	STA	LTA	TRIG	DTRIG	DSK	REC
1	1	1	25	20	10	BERG	BHZ	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
2	1	2	25	20	10	BERG	BHN	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1
3	1	3	25	20	10	BERG	BHE	00	2.0	8.0	4.0	400.0	2.5	1.5	1	1

/mnt/cf/SLG/PAR/DIG_TDT

NO UNIT SER_PORT BAUDRATE

IP-ADDRESS/DOMAIN SOCKET SRATE TIMING TIME-CORR

```
1 TDT_S COM02 9600
```

```
- - 50 2 0.0
```

"seislog" = /root/SLG/PRO

"srv" = /root/SLG/PRO

Ejecutando seislog:

```
#ls -l /dev/ttyAM1
```

```
#cat /dev/ttyAM1
```

```
#cd /root/SLG/PRO
```

```
#./seislog -tag -d DIG_SAD1 -c CHA_SAD1 -t /root
```

Funcionalidad del "CRON":

Se debe crear un archivo de texto con el nombre del usuario donde se corre el seislog, en este caso "root" en /var/spool/cron/crontabs/ con los parámetros de arranque de seislog. Luego correr el cron:

```
$crond start
```

```
$ cat /var/spool/cron/crontabs/root
```

```
* * * * * /root/SLG/PRO/seislog -tag -d DIG_XXX -c CHA_XXX -t /root
```

En el caso del digitalizador "TDT3C", el tiempo se sincroniza con el protocolo "NTP", que para el sistema TS-7260, aún está en proceso de desarrollo.

4.3.3 Resultados de las pruebas.

DIG_TDT/CHA_TDT	ACME/FOX	TS-7260
Seislog en 96hrs	-Sistema con arranque automático, corriendo perfectamente.	-Funcionamiento continuo -Sin interrupciones, corriendo bajo crond.
Estabilidad	-Sistema estable	-Sistema muy estable
Observaciones	-En estas últimas pruebas, se han presentado problemas en la puerta serial, debido a que se llena rápidamente el buffer de éste, haciendo que quede tomada la puerta serial y no permite el ingreso de los datos. Para que esto no ocurra, se conectó la señal de datos justo momentos antes de arrancar seislog.	-Sin problemas de buffer en la puerta serial.

5. Conclusiones

Después de realizar los test a los tres sistemas, y considerando únicamente la estabilidad y funcionalidad con los diferentes instrumentos (digitalizadores), como criterio principal, el "TS-7260", es el sistema más estable, robusto y funcional.

Lo anterior, pensando en que estos sistemas estarán orientados principalmente al trabajo de campo (terreno), donde deben funcionar como estaciones autónomas.

Consumo:

De acuerdo a los test, el sistema GUMSTIX quedó descartado y no se siguió realizando el análisis, ya que el consumo de energía es muy alto respecto de los otros dos equipos, llegando incluso a los 700mA, dependiendo del tipo de memoria que se esté empleando, siendo Compact-Flash la memoria con mayor consumo (tabla comparativa). Este sistema al requerir un board base, más el waysmall y el de red, en total alrededor del doble del consumo que los otros dos sistemas.

En el caso del TS-7260 y ACME/FOX, el consumo de energía es muy similar entre ambos, alrededor de los 300mA, bajando incluso a los hasta los 190mA en el TS-7260.

El ACME/FOX requiere de una fuente de muy bajo ruido, para que funcionen correctamente todos sus periféricos, y sólo soporta alimentación de 5vDC. El TS-7260 permite una alimentación en rango más amplio (4,5-20 vDC) y no se requiere de una fuente de energía de bajo ruido.

Por otra parte, si se va a emplear estos sistemas "embedded", en el campo, (paneles solares – baterías – controlador de carga), el sistema de energía entrega 12vDC, por lo que el TS-7260 no requeriría de un convertidor de 12v a 5v como en el caso de ACME/FOX, sino alimentación directa del controlador de carga.

Estabilidad:

En cuanto a la estabilidad, el sistema TS-7260 resalta por lo robusto respecto a los otros dos sistemas. Corriendo seislog con "crond", en ninguno de los test se detuvo o dejó de registrar datos, funcionando ininterrumpidamente. Además, para realizar la transferencia de los eventos, vía FTP, teniendo el servidor "srv" activado (lissplot), sesiones abiertas (telnet), y con seislog corriendo, ésta nunca se cortó o produjo algún problema de registro que afectara el funcionamiento del sistema.

En el caso del ACME/FOX, una vez que el sistema seislog se encuentra corriendo, no se detiene, funciona correctamente, en todos sus parámetros. El problema principal, dice relación con el modo de cómo se arranca el sistema. El buffer de la puerta serial "ttyS0", tiene limitantes, esto es, cuando se conecta el digitalizador antes del "booting", ello hace que el buffer se llene y bloquee la puerta, de modo tal que provoca que sea imposible capturar los datos que entrega el digitalizador. Esto sucede sólo mientras el digitalizador esté entregando datos. Para poder lograr que el sistema de registro funcione correctamente y pueda ser

cargado desde el inicio, fue necesario conectar el digitalizador en el momento justo antes de arrancar seislog. (Esto aun esta en fase de prueba).

Con respecto a la transferencia de los eventos del ACME/FOX, emplea los protocolos "SCP" y "FTPD". En ambos protocolos se producen problemas de transferencia de los archivos cuando son volúmenes altos de datos. Si son pocos archivos (por ej: 1, 2 o 3), no se producen problemas de transferencia, con seislog corriendo y el servidor "srv" activo. Si son muchos archivos, independiente del tamaño de estos, el sistema se detiene por completo y hay que reiniciar. Por lo tanto, necesariamente hay que extraer la memoria USB y sacar los datos directamente de ella, deteniendo el registro de continuo de seislog. Esto es un aspecto importante a considerar si se requiere tener registro continuo de datos.

Arranque:

El arranque del TS-7260, está determinado por el "crond" incluido su versión de kernel para el sistema antes mencionado, y configurado a cada minuto, hora, día del mes, día de la semana (* * * * *). La ejecución es manual al momento de arrancar el TS.

Para el ACME/FOX, el arranque está dentro de los "Run Levels" y al momento de arrancar el sistema, éste se carga automáticamente.

Digitalizadores:

Los digitalizadores utilizados en los test fueron; "EarthData PS-2400", "SADC10" y "TDT3C".

Los archivos ejecutables de seislog dependen en qué sistema embedded se emplee. Se crea una compilación externa para cada uno, independiente de qué versión de kernel tenga uno u otro.

En el caso del ACME/FOX, se hizo necesario actualizar el núcleo del sistema del kernel 2.4 al kernel 2.6, para que el sistema corriera correctamente, específicamente en la captura de datos por puertos las seriales.

En los análisis de las trazas de los archivos registrados (minised) de cada sistema, se realizó con "Seisan 8.1", pero para poder ver correctamente estas trazas, el programa debe tener el parche adecuado.

Seislog-Linux para los sistemas embedded, aquí analizados, soporta varios tipos de digitalizadores aparte de los antes mencionados, como por ejemplo; "EarthData 6CH", GVB-316, etc.

GPS v/s NTP:

En este tema, cuando se tiene que la estación sismológica funciona con un digitalizador con tiempo, es decir que incluya un "GPS", el protocolo "ntp" no es importante, ya que el tiempo viene junto con en el dato que entrega el digitalizador. En el caso de los digitalizadores que no incluyen el tiempo, como es el caso del "TDT3C", éste se sincroniza con el reloj de la CPU y éste a su vez con el reloj interno. Para corregir los errores de retardos o adelantos, se debe emplear el protocolo de red de tiempo "NTP", en modo cliente, para conectar el sistema a un servidor de tiempo "NTPD" (modo UTC). Este servidor debe estar ojala dentro de la región. De acuerdo con las pruebas realizadas, las diferencias de sincronía entregadas en el "Debug" de tiempo de seislog registradas, estaban dentro del orden de "0.00X"seg.

Ventajas / Desventajas:

TS-7260:

Dentro de las principales ventajas de este sistema están:

- Su robustez en todo aspecto.
- Bajo consumo.
- Funcionalidad para la manipulación de datos (eventos).
- Servicios de red que permiten y múltiples sesiones que no impiden el funcionamiento del resto del sistema.
- No requiere de algún dispositivo adicional, ya que en el board está todo compactado.
- Procesador más rápido que el del ACME/FOX.

- Muy buena documentación.

En cuanto a las desventajas:

- El kernel no incluye el servicio "SSH" para hacer sesiones seguras, solamente permite acceso vía "Telnet".
- El arranque automático de seislog está en proceso de aplicación.
- Costo levemente superior respecto al ACME/FOX.

ACME/FOX:

Dentro de las ventajas de este sistema respecto del antes mencionado:

- El kernel incluye herramientas de seguridad como "iptables" y "SSH".
- Arranque automático del sistema seislog.
- Mayor facilidad de configuración de los periféricos y programas.
- Distribuidor a nivel nacional.

Con respecto a las desventajas:

- Poca documentación respecto a la que tiene el TS-7260.
- Inestabilidad debido al problema del buffer de la puerta serial en la captura de datos.
- Menos robustez que el TS-7260.
- Requiere de un dispositivo adicional al board para conectar el digitalizador a la puerta serial (adaptador de consola).
- Funcionalidad óptima sólo con una fuente de bajo ruido, y no admite un voltaje de alimentación distinto a 5v.

Temas Pendientes:

Dentro de los temas por solucionar, están:

- Arranque automático de Seislog y mejoramiento del "client-protocol" de "NTP" en el TS-7260 para el funcionamiento con digitalizadores sin tiempo (gps).
- Desarrollar una aplicación que permita, mientras el digitalizador está activo y conectado a la puerta serial del ACME/FOX, antes de arrancar seislog, no bloquear la puerta serial producto de la acumulación de datos en el buffer.
- Desarrollar una aplicación "Firewall" (iptables), que permita proteger a los sistemas mientras estén en red, ya que por defecto vienen totalmente desprovistos de alguna herramienta de seguridad de red.

Por lo tanto, hasta aquí, de acuerdo a lo antes concluido, se puede decir que el sistema que mejor se adapta a los requerimientos de implementación para una estación sismológica, es el "TS-7260".

Apéndice 1: Configuración ACME/FOX

CAMBIO DE DIRECCION IP

Por defecto el ACME/FOX está configurado con la IP falsa: **192.168.0.90/24**. Para cambiar esta dirección se deben modificar los siguientes parámetros:

Para verificar la configuración de la interfaz de Red ETH0:

```
# ifconfig eth0 x.x.x.x
```

Modificar los archivos:

```
/etc/conf.d/net.eth0
IP="192.168.0.90"
NETMASK="255.255.255.0"
BROADCAST="192.168.0.255"
GATEWAY="192.168.0.1"
```

```
/etc/resolv.conf
DNS
```

Aquí, ambos archivos deben modificarse agregando los Servers-

```
/etc/resolv.conf.def
# nameserver x.x.x.x
```

REVERTIR A LOS VALORES POR DEFECTO:

Para volver a los valores por defecto del sistema, por una mala manipulación o archivos dañados, se deben seguir los siguientes pasos:

- Apagar el ACME/FOX
- Mantener presionado el SW1
- Encender el ACME/FOX
- Mantener presionado el SW1 durante aproximadamente 10 seg.

Este es un Reset del sistema a los valores por defecto. Todos los valores de /etc quedarán restituidos a sus valores de fábrica.

MONTAR USB-MEMORY

```
#mount -t vfat /dev/sda1 /mnt/1
#cd /mnt/1
```

HABILITACIÓN PUERTA SERIAL/CONSOLA "/dev/ttyS0" PARA CAPTURA DE DATOS

Editar el archivo en /etc/inittab.

Comentar última línea para deshabilitar la consola en el puerto serial "0".

```
# The runlevels used by axis are:
# 0 - Halt
# 1 - Single user mode
# 2 - Multiuser without network
# 3 - Full multiuser mode
# 4 - unused
# 5 - unused
# 6 - Reboot

id:3:initdefault:

# System initialization.
si::sysinit:/etc/init.d/rc sysinit

# The initscripts.
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6

# Add your own stuff below
# A file (/tmp/start_ok) just to indicate that the product is running.
start_ok:3:once:/bin/start_ok.sh

# Uncomment the following line to get a console shell.
#sh:3:respawn:/bin/sh ####Dejar con comentario
```

INICIO AUTOMATICO DE LOS PROGRAMAS EN EL ARRANQUE

Para iniciar un "Daemon", al arrancar el sistema automáticamente en el booting, deben agregar las líneas de instrucción en:

/etc/init.d/rc

```
#!/bin/sh

if [ "$1" = sysinit ]; then
    echo "Starting system initialization."

    for file in /etc/rcS.d/S??*; do
        if [ -x "$file" ]; then
            $file start
        fi
    done

    echo "System initialization is done."
else
    runlevel="$RUNLEVEL"
    [ -z "$runlevel" ] && runlevel="$1"
```

```

if [ -z "$runlevel" ]; then
    echo "Usage: $0 <runlevel>" >&2
    exit 1
fi

prevlevel=$PREVLEVEL
[ -z "$prevlevel" ] && prevlevel=N

echo "New runlevel: $runlevel"

if [ -d /etc/rc$runlevel.d ]; then
    if [ "$prevlevel" != N ]; then
        for file in /etc/rc$runlevel.d/K??*; do
            # Only stop what isn't already stopped.
            if [ ! -e "/etc/rc$prevlevel.d/${file##*/}" ] \
                && [ -x "$file" ]; then
                $file stop
            fi
        done
    fi

    for file in /etc/rc$runlevel.d/S??*; do
        # Only start what isn't already started.
        if [ ! -e "/etc/rc$prevlevel.d/${file##*/}" ] && \
            [ -x "$file" ]; then
            $file start
        fi
    done
fi

mount -t vfat /dev/sdal /mnt/1; ###Instrucción 1
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx; ###Instrucción 2
...
...                               ###Instrucción X
echo "All scripts for runlevel $runlevel are done."
fi

exit 0

```

IMPORTANTE:

- La alimentación del este sistema debe ser con una fuente de +5v regulada de bajo ruido.
- El buffer del puerto ttyS0 se llena al momento del arranque si está conectado el digitalizador. Este debe conectarse una vez que el kernel ya se ha cargado (Led Rojo). En proceso de análisis.

Apéndice 2: Configuración TS-7260

CAMBIO DE DIRECCION IP

Estos son los archivos de configuración de la interfaz de red (valores por defecto):

\$/etc/sysconfig/ifcfg-eth0

```
DEVICE=eth0 # Name of ethernet interface
IPADDR=192.168.0.50 # Internet Protocol address of this ethernet interface.
NETMASK=255.255.255.0 # Used with NETWORK to determine which IP's are local.
NETWORK=192.168.0.0 # Used with NETMASK to determine which IP's are local.
BROADCAST=192.168.0.255 # Broadcast IP for system wide messages.
BOOTPROTO=static # Static IP (change `static` to `DHCP`)
ENABLE="yes" # Load device on boot.
```

El TCP/IP network settings estan configurados en el archivo:

\$/etc/sysconfig/network_cfg

```
NETWORKING=yes # Enable networking on startup
GATEWAY="192.168.0.1" # Gateway for internet access.
GW_DEV=eth0 # Default gateway
HOSTNAME=ts7200 # Host name of this computer.
BOOTPROTO=no
FORWARD_IPV4=no
DEFRAG_IPV4=no
```

El TCP/IP name server esta configurada en el archivo:

```
$/etc/resolv.conf
Nameserver 192.168.0.1 # Name server for domain name lookups.
```

POR DEFECTO ES SISTEMA ESTA CONFIGURADO CON LA DIRECCION: "192.168.0.50".

MONTAR USB-MEMORY

Este script carga los módulos necesarios para montar la memoria USB:

```
/usr/bin/loadUSBModules.sh
insmod usbcore
insmod pcipool
insmod usb-ohci
insmod usb-ohci-ep93xx
insmod scsi_mod
insmod sd_mod
insmod usb-storage;
```

Esta instrucción permite hacer el montaje de la unidad de memoria USB una vez conectada:

```
#!/bin/mount -t vfat /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/cf
```

CROND

Se requiere crea un script "root" con las siguientes instrucciones:

```
/var/spool/cron/crontabs/root
```

```
* * * * * /root/SLG/PRO/seislog -tag -d DIG_XXX -c CHA_XXX -t /root
```

HABILITACIÓN PUERTA SERIAL/CONSOLA "/dev/ttyAM1" PARA CAPTURA DE DATOS

Editar el archivo en /etc/inittab.

Comentar línea color azul para deshabilitar la consola en el puerto serial "COM2".

```
#
# inittab          This file describes how the INIT process should set up
#                  the system in a certain run-level.
#                  Please read a standard system's man pages for inittab
#

# Runlevels. The runlevels used by RedHatSystems are:
# 0 - halt (DO_NOT set initdefault to this)
# 1 - Maintaince (single user) mode.
# 2 - Multiuser, no networking support
# 3 - Full multiuser mode
# 4 - No standard. Used for weird/special runlevel
# 5 - Full multiuser with fbdev text console started
# 6 - reboot (DO_NOT set initdefault to this)
#

# Default Runlevel to Go into
dflt:3:initdefault:

# System Startup. Do these process before
# anything else
strt::sysinit:/etc/rc.d/rcS.sysinit
#smpl::sysinit:/path/to/process

# Process to do for each specific Runlevel
#smpl:4:actionkeyword:/path/to/process arguments
r0:0:wait:/etc/rc.d/rcS 0
r1:1:wait:/etc/rc.d/rcS 1
r2:2:wait:/etc/rc.d/rcS 2
r3:3:wait:/etc/rc.d/rcS 3
r4:4:wait:/etc/rc.d/rcS 4
r5:5:wait:/etc/rc.d/rcS 5
r6:6:wait:/etc/rc.d/rcS 6

# Things to run once specified runlevel is entered
#ud::once:/sbin/update
#
# Start console logins on both COM Ports
#
lgn0:12345:respawn:/sbin/getty -L 115200 ttyAM0
#lgn1:12345:respawn:/sbin/getty -L 115200 ttyAM1

#
# Start Virtual Terminal Logins
#
vty1:5:respawn:/sbin/getty 38400 vc/0
###vty2::respawn:/sbin/mingetty vc/2

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -r now
```

INICIO AUTOMATICO DE LOS PROGRAMAS EN EL ARRANQUE

Para iniciar un "Daemon", al arrancar el sistema automáticamente en el booting, deben agregar las líneas de instrucción en:

```
etc/rc.d/rcS.sysinit
```

```
# Technologic Systems   date: 8.03.04
# This file is used by inittab and init. Place here all the stuff you want
# to be done before any other system startup scripts are executed
# that will affect the system as a whole or are needed by the system

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

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

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

#start devfsd
/sbin/devfsd /dev

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

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

/usr/bin/loadUSBModules.sh
#/bin/mount -t vfat /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/cf

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

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

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

if [ -f /www/apache/logs/httpd.pid ]; then
    rm -f /www/apache/logs/httpd.pid
```



```

fi
echo "MOUNT USB"
sleep 5
/bin/mount -t vfat /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/cf
$ clear
$ cat /etc/rc.d/rcS.sysinit
#!/bin/sh
#
# /etc/rc.d/rcS.sysinit
#   Technologic Systems   date:  8.03.04
# This file is used by inittab and init. Place here all the stuff you want
# to be done before any other system startup scripts are executed
# that will affect the system as a whole or are needed by the system

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

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

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

#start devfsd
/sbin/devfsd /dev

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

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

/usr/bin/loadUSBModules.sh
#/bin/mount -t vfat /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/cf

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

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

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

if [ -f /www/apache/logs/httpd.pid ]; then
    rm -f /www/apache/logs/httpd.pid

```

```
fi
echo "MOUNT USB"
sleep 5
/bin/mount -t vfat /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/cf
INSTRUCCION 2.....
INSTRUCCION 3.....
.....
```

Apéndice 3: GLOSARIO

Embedded: Conjunto de hardware y software, que permite desarrollar una función específica. Se caracterizan por su bajo consumo y tamaño pequeño.

TDT3C: Nombre que se le da al digitalizador de tres componentes 24bit, desarrollado por Ramón Ortiz del Departamento de Vulcanología "CSIC", España.

EarthData: Marca del digitalizador de tres componentes 24bit, Modelo: PS-2400.

SADC10: Modelo de digitalizador de 16bit, multicanal, de origen Italiano.

PC/104-Radrunner II: Modelo estándar "PC/104" embedded.

ACME/FOX: Sistema embedded de origen Italiano, utilizado en los test.

GUMSTIX: Sistema embedded de origen Canadiense, utilizado en los test.

TS-7260: Sistema embedded de origen Estadounidense, utilizado en los test.

SCP: Secure Copy Protocol. Protocolo que emplea los módulos de SSH hacer las trasferencia de archivos.

PPPd: Daemon, Protocolo Punto a Punto

GCC: Compilador de C, para distribuciones Linux.

HTTPd: Daemon, Servidor Web.

IPtables: Módulo que permite crear reglas para filtrar el trafico TCP/IP, en Linux.

CROND: Daemon, que permite ejecutar comandos, scripts, archivos, en forma programada.

Syspar: Archivo de configuración de Seislog-Linux donde se especifican las rutas donde se guardan os ringbuffer y eventos.

CHA_XXX: Archivo de configuración de Seislog-Linux que especifica por ejemplo; los canales, nombres componentes, triggers, tiempos de preevento y posteventos.

DIG_XXX: Archivo de configuración Seislog-Linux que especifica; el tipo de digitalizador, baudrate, serial port, tiempo, muestras.

PRO: Directorio correspondiente a Seislog-Linux donde están guardados los archivos ejecutables de Seislo.