

Progetto TN

Versione 1.0

03/01/06

Autori: S.Pintore, L.Salvaterra

Nota

Questo documento è in versione **preliminare** e sarà soggetto a revisioni del testo, oltre che a modifiche sostanziali per tenere conto dell'evoluzione del software della scheda, tuttora in corso. Ogni informazione riportata riguarda le schede TN-1 (Trasmission Node 1) installate nelle stazioni GAIA, alla data riportata in questo documento. Le informazioni riguardanti l'hardware sono praticamente definitive, salvo correzioni.

Convenzioni nel testo

In questo documento verranno seguite le seguenti convenzioni tipografiche:

- *corsivo* per i nomi di programmi
- **grassetto** per la specifica di linee di comando

Sistema Transmission Node 1

1. Introduzione

Questo primo capitolo è incentrato sulle caratteristiche hardware e software del sistema Transmission Node 1, d'ora in avanti detto TN-1. Dopo una veloce carrellata delle potenzialità del sistema verrà descritta la scheda su cui è basato il sistema e l'architettura software (sistema operativo e programmi sviluppati).

2. Generalità

La scheda TN-1, parte integrante della stazione sismica INGV GAIA (Geophysical All Inclusive Aquisitor), rende disponibili una serie di opzioni di connettività e di telemetria aggiuntive a quelle della scheda di acquisizione a 24 bit AGDF1 precedentemente progettata e prodotta dal Laboratorio CNT-INGV.

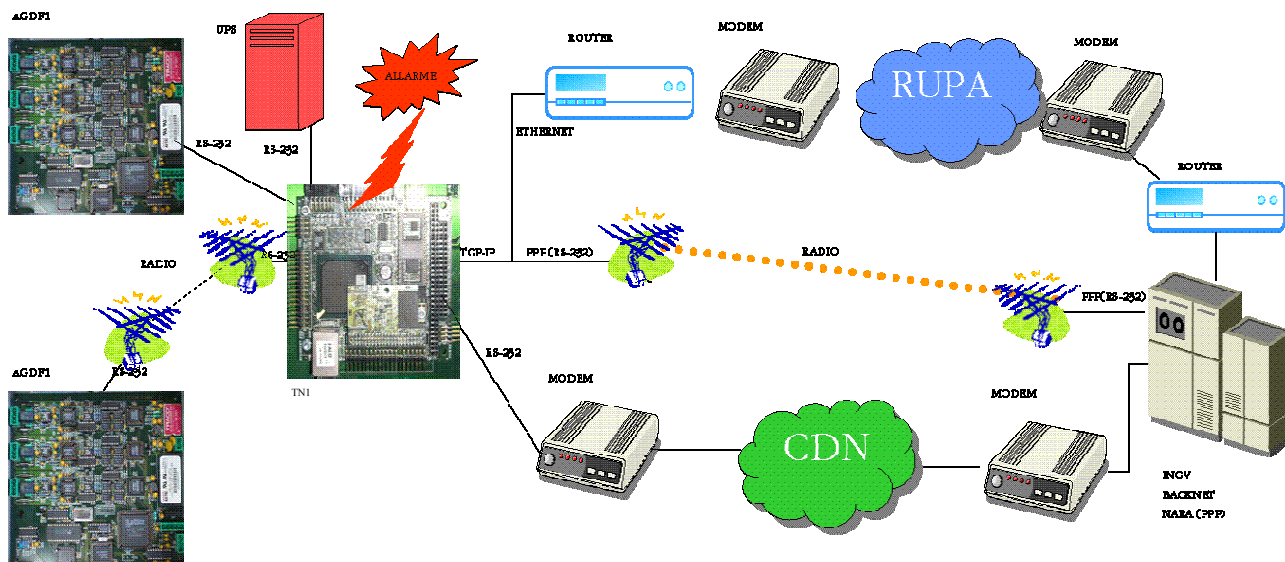


Figura 1: Schema logico generale del progetto Transmission Node

Il compito della scheda TN-1 all'interno del progetto GAIA è stato ampliato in fasi successive fino ad assumere, nella sua configurazione finale, una struttura complessa.

La scheda TN-1 svolge attualmente i seguenti compiti (fig. 1):

1. raccoglie uno o più flussi di ingresso provenienti da schede AGDF-1 attraverso collegamenti seriali (tramite cavo e/o ponte radio)
2. convoglia i flussi di input su: una linea seriale di uscita; un socket tcp remoto, dialogando con il sistema di acquisizione Backnet; un socket tcp remoto su seriale grazie all'uso del protocollo PPP;
3. gestisce l'aggiornamento della scheda AGDF-1 tramite terminale seriale, anche remotamente attraverso la rete IP;

4. monitora i flussi di input per eventuali anomalie dovute ad errori nel collegamento con l'AGDF-1, che possono verificarsi ad esempio in presenza di collegamenti in ponte radio e corregge se possibile tali errori di trasmissione;
5. monitora i flussi di input per riconoscere i possibili problemi di perdita di sincronismo dovuti a irregolarità nel funzionamento del modulo GPS connesso alle schede AGDF-1;
6. monitora le informazioni provenienti dal sistema di alimentazione UPS;
7. invia al sistema di controllo messaggi di allarme di stato in base ai parametri monitorati;
8. se equipaggiata di scheda Compact-Flash (CF) opzionale, consente la memorizzazione locale dei dati sismici;

In più, essendo la scheda TN-1 a tutti gli effetti un PC dotato di sistema operativo Linux:

9. consente, tramite protocolli di comunicazione “sicuri”, la comunicazione con stazioni di controllo remote;
10. consente l'aggiornamento dei software applicativi da parte della stazione di controllo remota, senza necessità di intervento di personale tecnico alla stazione sismica;
11. consente l'aggiornamento dello stesso sistema operativo, se necessario anche dalla stazione di controllo remota.

La scelta dell'hardware da utilizzare per l'implementazione della scheda TN-1 è caduta su un computer industriale in formato PC-104, il modello M543 prodotto dalla SECO srl.. Tale PC, è basato a sul processore della AMD ELAN SC520. Il progetto TN-1 ha implicato la personalizzazione del sistema operativo Linux adattandolo a tale scheda ed alle esigenze di connettività richieste. Il sistema operativo della scheda TN-1 risiede su un Disk On Module (DOM) da 32 Mbyte. Tale disco contiene l'immagine del file-system che è stata creata basandosi principalmente sulle distribuzioni di Linux disponibili in Internet, aggiornandone il kernel ed alcune librerie di sistema, prendendo questi da varie distribuzioni , Slackware e Mandrake principalmente.

Il kernel è stato ricompilato partendo dal sorgente nella versione 2.4.28. Il sistema è dotato dei server sftp, nfs, smb, http, dhcp, snmp, pppd, ssh, telnet e getty su una porta seriale. Attraverso gli ultimi tre è possibile accedere al sistema qualora non siano disponibili monitor e tastiera. Il root file system occupa approssimativamente 28 MB, ovvero circa l'88% del DOM su cui risiede. Il file system è dotato di vari strumenti di amministrazione del sistema (descritti più avanti) ed il text editor vi. I file di configurazione del sistema sono disponibili nella directory **/etc**, i file relativi al software di trattamento dati provenienti dall'acquisitore AGDF-1 si trovano nella directory **/applicativo**, mentre gli eventuali dati acquisiti e memorizzati nella compact flash stanno nella directory **/data** .

3. Descrizione hardware della scheda M543

La scheda PC104 (90 x 96 mm, con altezza di 15,2 mm) dalla SECO srl, chiamata M543, è basata sul processore della AMD ELAN SC520, un processore pienamente compatibile con l'Intel 486DX4/DX5 con 16 Kb di cache, funzionante a 100 o 133 MHz. Nella scheda si trovano un controllore per SDRAM (fino a 256MB montata direttamente sulla scheda) e per ROM/Flash. Compagnano anche le più comuni periferiche per PC, quali 3 porte seriali RS232C e una configurabile come RS232/RS422/RS485, una porta parallela bidirezionale standard (configurabile come SPP, EPP/ECP o, anche, come secondo connettore per Floppy Disk), un controller per floppy disk e per l'IDE, tastiera e mouse PS/2. Sulla scheda è anche disponibile un connettore DIL a 44 pin per l'installazione di DOM (Disk on Module – memoria a stato solido) visto dal SO come un vero e proprio hard disk. L'M543 è dotata anche di controller e relativo connettore, Ethernet con transfer rates di 10/100 Mbps.

Altre specifiche tecniche sono:

- Controller Interrupt con 15 linee di Interrupt
- Controller DMA con 4 canali
- Standard Timer Controller con 3 timer programmabili
- Watchdog Timer con generazione di un reset hardware
- Speaker Output
- Real Time Clock con batteria di backup
- Alimentazione a 5VDC \pm 10%
- Assorbimento tipico di 700 mA @ 100MHz, +5V

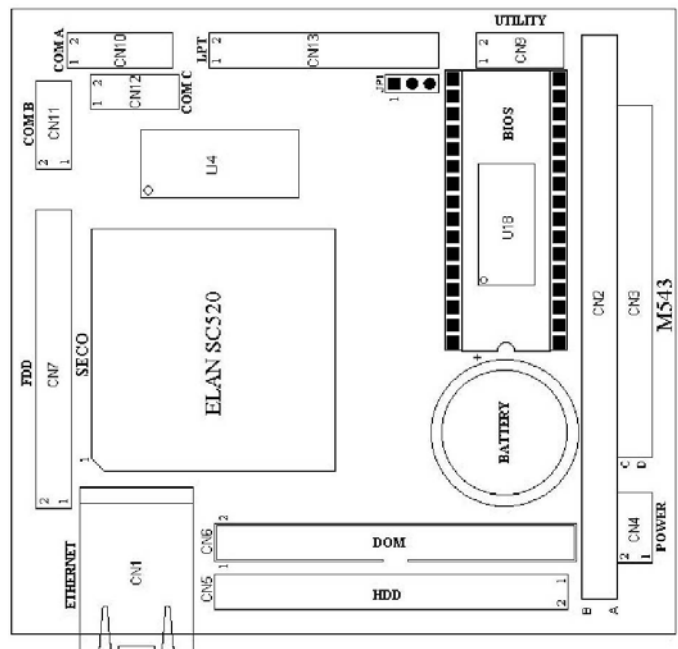


Figura 2: Schema dei connettori della scheda M543

Nella figura 2 è possibile vedere i connettori della scheda M543: si notano due connettori (chiamati HDD e DOM) che afferiscono allo stesso canale IDE, un connettore (POWER) per l'alimentazione in alternativa all'utilizzo del bus PC104 (CN2 e CN3), tre connettori per le seriali (COM A, COM B e COMC) in cui in COM A ci sono le prime due seriali del sistema, il connettore ETHERNET, i connettori per il floppy FDD e la porta parallela LPT, e infine, un connettore (chiamato UTILITY) in cui vi sono i pin per la tastiera, il mouse e l'audio. Vi è montata, inoltre, una batteria di backup per l'RTC.

4. Informazioni sul Kernel

Il kernel è chiamato vmlinuz-2.4.28TN-1 ed è configurato e compilato per CPU 486. Fornisce il supporto per i moduli caricabili, il supporto network, il supporto generico PCI, PNP e automounter. I dispositivi supportati sono: IDE disk, IDE cdrom loopback, RAM disk, initrd, network block device.

I dispositivi a caratteri supportati sono: console, virtual terminal, standard generic serial, Unix 98 pty e mouse PS2.

Il processore AMD Elan SC520 è dotato di un watchdog timer hardware, il cui driver è stato da noi modificato per la compilazione nel source tree prescelto. La libreria libc utilizzata è la libc-2.2.5.

I console drivers sono : VGA text console, frame buffer, VESA VGA and VGA 16 graphic. I file system supportati sono: VFAT, ISO9660, MS Joliet, NTFS (in read only), EXT2 e EXT3.

I network file systems sono: nfs, nfs-server and smb. I protocolli di rete installati sono: TCP/IP, DHCP, socket filtering, multicasting, firewalling, ppp.

5. Installazione del Disk On Module

L'installazione, effettuata nei laboratori INGV dagli stessi sviluppatori, prevede l'inserimento nel sistema di sviluppo (attualmente è una TN-1 completa di scheda video e dotato di Hard Disk con sistema operativo Linux e sorgenti degli applicativi) del DOM vuoto da programmare. Dopo aver avviato il sistema da hard disk si esegue uno script, che ha automatizzato la creazione del file system sul DOM, chiamato "**dom-install**". Tale script (riportato di seguito) esegue in sequenza delle operazioni che consentono la creazione del file system sul DOM in formato EXT3, il check del file system creato, la copia di tutti i file del sistema operativo e la configurazione del settore di boot per l'avvio. Al termine dell'installazione il DOM è pronto per l'uso e per essere configurato secondo le necessità.

DOM-INSTALL

```
#The device /dev/dom = /dev/hdb
#Create file system
# -c -c check before create
# j ext3
# J size=1024 minimum size for the journal
#- m 0 no reserved blocks
echo "DOM configuration, please wait"
DOM="/dev/dom"
```

```

DOM_DEVICE="/dev/dom1"
SIZE=$(sfdisk -s $DOM)

echo "$DOM Dimension: $SIZE"
echo "Partitioning $DOM"

sfdisk $DOM << EOF
0,,,*,
;
;
;
EOF

SIZE=$(sfdisk -s $DOM_DEVICE)

echo "$DOM Dimension: $SIZE"
echo "Creating ext3 file system"
mke2fs -c -j -J size=1 -m 0 $DOM_DEVICE
tune2fs -e continue $DOM_DEVICE

#Copy all the file system default contents
ALL_NEEDED=$1
MOUNT_POINT="/home/sviluppo/seco-system/dom-ext3"

echo "Creating destination directory"
mkdir -p $MOUNT_POINT
mount -t ext3 $DOM_DEVICE $MOUNT_POINT

echo "Copying all needed .."
cp -a $ALL_NEEDED/* $MOUNT_POINT

echo "Chrooting and running lilo"
chroot $MOUNT_POINT lilo -v -v -v -C etc/lilo.chroot
lilo -q
umount $DOM_DEVICE
echo "Done"

```

6. L'architettura software

Il software sviluppato prevede vari livelli operativi svolti da processi distinti e comunicanti tra loro. In questo modo si è cercato di realizzare una struttura flessibile in grado di adattarsi alle esigenze del progetto senza di volta in volta intaccare parti già definite e collaudate. Così, è stato possibile proseguire lo sviluppo software, attraverso varie correzioni, modifiche e aggiornamenti, anche dopo aver fatto le prime installazioni delle stazioni GAIA IP.

C'è quindi separazione tra il software dedicato alla comunicazione con l'acquisitore (AGDF-1), in seguito indicato *master2bn*, e il software per la trasmissione del dato sismico col protocollo WF su tcp, *splitter*.

Il software realizzato consiste dei seguenti programmi C :

1. *setup*
2. *master2bn*
3. *splitter*
4. *ppups*

e degli script bash:

5. *agdfd*
6. *start.sh*
7. *delete.sh*
8. *compressor.sh*

Per semplicità tutti gli eseguibili dei programmi, gli script e gli altri file di configurazione del software si trovano nella directory **/applicativo**.

SETUP

Il programma *setup* ha lo scopo di creare il file di configurazione per il programma *master2bn*, *config.staz*. Questo file di configurazione contiene la struttura (C) descritta nel file *setup.h*.

Questo programma può essere avviato in modo interattivo lanciandolo col nome *i-setup*, oppure può essere avviato in modo non interattivo specificandone le opzioni da riga di comando.

Nella modalità interattiva il programma pone all'operatore una serie di domande relative a:

- Numero di schede AGDF collegate
- Uso della trasmissione dati TCP-IP
- Uso della trasmissione dati su porta seriale

e per ogni scheda collegata in seguito chiede:

- Nome
- Baud rate scheda

- Numero di canali
- Da trasmettere (y/n)
- Grandezza file di log in minuti

Il programma può essere avviato anche in una modalità non interattiva che prevede l'immissione dei parametri direttamente da riga di comando.

La sintassi del comando è la seguente:

“setup eth baud_rate_out board_name baud_rate ch_num tx file_size”

I parametri da specificare sono:

- eth [y/n] - specificare y oppure n se si vuole abilitare o no la trasmissione tcp-ip
- baud_rate_out - inserire la velocità della porta seriale di output, 0 per non trasmettere sulla porta seriale.

Dopo questi due parametri fissi segue una lista di parametri di configurazione relativa a ciascuna scheda AGDF-1 che si intende collegare, quindi per ogni scheda AGDF-1 collegata vanno specificati nell'ordine:

- board_name – inserire il nome della scheda connessa alla porta seriale (sigla di stazione)
- baud_rate – inserire la velocità della porta seriale, ovvero la velocità in output della scheda AGDF-1
- ch_num – numero di canali presenti sulla scheda
- tx [y/n] – specificare y oppure n se si vuole abilitare o meno la trasmissione di questa scheda AGDF-1
- file_size – indicare il numero di minuti di registrazione dei file di dati da archiviare localmente. Indicare 0 (zero) se si vuole disabilitare l'archiviazione locale.

Esempio:

Vogliamo creare il file di configurazione per la stazione di Augusta a cui sono collegate due schede AGDF-1, da trasmettere in TCP-IP e non in seriale. La prima scheda, di nome AGST_, trasmette a 19200 b/s, su 3 canali, va trasmessa, ma non archiviata localmente. La seconda scheda, di nome MARE_, trasmette a 9600 b/s, su un solo canale, va trasmessa, ma non archiviata. Scriveremo il comando:

setup y 0 AGST_ 19200 3 y 0 MARE_ 9600 1 y 0.

Ovviamente si consiglia di utilizzare la modalità interattiva, la seconda modalità è stata realizzata per il caso in cui si voglia automatizzare il setup tramite altri software di interfaccia, che pensino a creare da soli la linea di comando corretta.

MASTER2BN

Il programma *master2bn* è il responsabile della comunicazione con le schede AGDF-1 ed è il più complesso, essendo l'unico multi-processo (fig. 3). Tutte le informazioni necessarie per la comunicazione con le schede AGDF-1 gli vengono comunicate tramite il file *config.staz*, mentre per la comunicazione con l'altro software *splitter*, *master2bn* prende i parametri necessari dalla linea di comando.

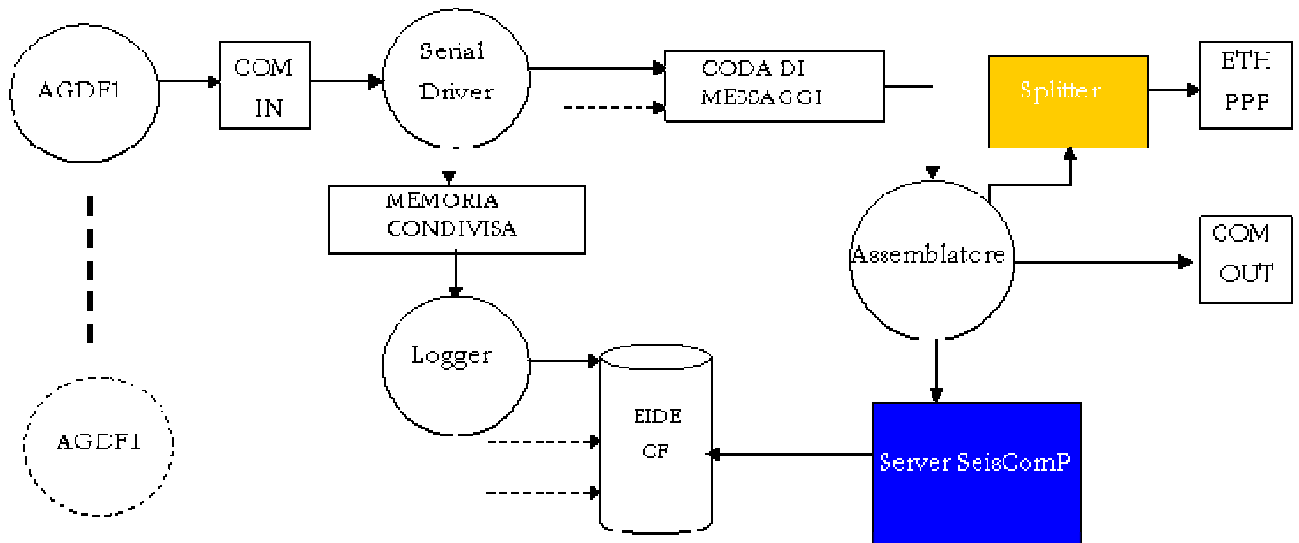


Figura 3: Diagramma a blocchi del software sviluppato per il progetto TN

La sintassi del comando è la seguente:

```
master2bn file_config [port] [interface]
```

- file_config : pathname del file di configurazione
- port: opzionale, è il numero di port TCP del socket su cui trasmetterà.
- Interface: opzionale, è l'indirizzo IP della interfaccia su cui aprirà il socket. Se omesso prenderà come default 127.0.0.1.

Esempio di linea di comando:

```
master2bn /applicativo/config.staz 999 10.100.10.38
```

Il processo principale (funzione main) ha il compito di lanciare e controllare la funzionalità dei processi figli: un processo Assemblatore e un processo Serial_Driver per ciascuna scheda AGDF-1 collegata e i cui dati si desidera trasmettere e/o archiviare.

Assemblatore

Il processo Assemblatore è stato progettato per sincronizzare le scritture dei flussi dati su una singola porta seriale di output, è suo compito estrarre i messaggi dalla coda riempita dai

Serial_Driver e scrivere i dati sulla porta di output seriale o su un socket tcp, tramite il quale comunica col programma *splitter*. Recentemente è stata studiata la possibilità di poter usufruire del collegamento seedlink utilizzando il server SeisComP. A tale scopo, Assemblatore è stato modificato per fornire i dati sismici e le informazioni necessarie (tempo, dati stazione ecc.) al server SeisComP.

SerialDriver

Ogni SerialDriver legge dalla seriale di competenza il flusso dati della scheda AGDF-1 e scrive i pacchetti in una coda di messaggi dopo averli analizzati e, eventualmente, corretti.

Al suo avvio il processo SerialDriver invia verso la scheda AGDF-1 il segnale di reset, successivamente attende la trasmissione dei pacchetti dati, si sincronizza al pacchetto e ricava dal pacchetto informazioni essenziali come nome scheda, nome dei canali e dimensione dei pacchetti stessi.

Il processo entra in un ciclo di: letture di pacchetti dalla seriale - analisi di correttezza - successiva spedizione nella coda di messaggi.

L'analisi di correttezza del pacchetto è fondamentale in tutti i casi in cui la scheda AGDF-1 è installata fisicamente distante dalla scheda TN-1, caso in cui la trasmissione su cavo seriale potrebbe risultare affetta da alcuni errori, oppure il collegamento avvenga ad esempio tramite un ponte radio, con errori ancora più frequenti.

Non viene fatto alcun controllo sulla correttezza dei dati contenuti nel pacchetto, non essendo previsto un meccanismo di verifica a livello del protocollo RSND-WF.

Nell'analisi di correttezza rientra quindi solo la correttezza formale della parte di intestazione (header), per quel che riguarda i seguenti campi:

- nome_stazione
- numero progressivo del pacchetto
- lunghezza del pacchetto
- End Of Block.

Esiste anche una fase di correzione del pacchetto, nei casi in cui l'errore sia solo nel nome della stazione o nel numero progressivo di pacchetto.

La presenza di più errori all'interno del pacchetto è considerata di maggiore severità e quindi si scarta il pacchetto arrivato e si invia al suo posto un pacchetto "dummy", corretto nell'intestazione, ma contenente nella parte dati un campione ripetuto di valore pari all'ultimo campione corretto inviato.

Al termine della verifica, il pacchetto viene inserito nella coda di messaggi servita dal processo Assemblatore. L'invio del pacchetto dummy è stato previsto per impedire al server *backnet* di

perdere la sincronizzazione al pacchetto, con l'effetto collaterale di chiusura della comunicazione con la stazione. Ovviamente il dato inviato non è corretto e visivamente si nota l'appiattimento del segnale sismico nel caso in cui i pacchetti dummy consecutivi siano numerosi. Purtroppo non ci sono alternative a questo, in quanto la scheda AGDF1 non consente la ritrasmissione del pacchetto su richiesta. Il controllo dei dati del GPS è assegnato sempre a SerialDriver che si occupa di avvisare, in caso di mancata ricezione del segnale dei satelliti GPS, il demone di sistema *syslogd*. Il demone di sistema *syslogd* è di solito configurato per il log remoto, di tutti i processi che girano sulla TN-1, su una macchina di monitoraggio remoto.

Il comportamento del programma si differenzia a questo punto se è prevista o no la memorizzazione locale dei dati.

Memorizzazione Locale

Nel caso in cui sia richiesta la memorizzazione locale dei dati, SerialDriver alloca una zona di memoria, strutturata in una coppia di memorie condivise per ciascun canale da memorizzare. Possiamo perciò descrivere il funzionamento di memorizzazione riferendoci al singolo canale.

Serial Driver copia i pacchetti consecutivamente nella area di memoria libera relativa al canale del pacchetto secondo il formato WF proprietario dell'INGV. All'esaurirsi dello spazio nel buffer zona-1, SerialDriver prosegue a scrivere nella zona-2, lanciando un altro processo concorrente (Logger) che dovrà salvare i dati su disco. Questa strategia è stata adottata per far sì di avere un ridotto numero di accessi al disco durante il funzionamento.

Logger

Il processo Logger viene utilizzato per salvare i dati presenti nelle memorie condivise nel formato WF compresso gzip.

Il funzionamento di Logger è configurabile in modo da selezionare la modalità di registrazione a bassa priorità oppure urgente.

Logger viene lanciato da SerialDriver non appena viene effettuato lo scambio tra le memorie condivise. Viene lanciata una istanza di Logger a bassa priorità per ciascuna memoria condivisa da salvare.

Ciascun canale ha pre-assegnata una tempistica diversa così i tre processi (nel caso dei tre canali) vengono lanciati scaglionati nel tempo e con priorità bassa. Questo garantisce nel caso normale di funzionamento un ridotto sovraccarico del sistema.

Logger viene invece eseguito in modalità urgente dal gestore di allarmi *catcher_SIGTERM_SD*, armato da SerialDriver.

Logger avvia inoltre la compressione dei file WF appena creati utilizzando lo script

compressor.sh, e, qualora sia necessaria, la cancellazione dei file memorizzati meno di recente utilizzando lo script *delete.sh*.

catcher_SIGTERM_SD

Questo è il gestore dell'allarme sul segnale SIGTERM per il processo SerialDriver. Nel caso venga intercettato questo allarme, il gestore lancia Logger con la massima priorità su ciascuna memoria condivisa. In tal modo è stato realizzato un meccanismo per salvare i dati in memoria sia nel caso che il programma venga volontariamente chiuso tramite questo segnale, ad esempio lanciato dallo script *agdfd* con parametro STOP (vedi oltre), sia nel caso in cui il sistema sia in fase di arresto o reboot. Il segnale SIGTERM, infatti, viene lanciato anche da init durante la fase di arresto del sistema.

Gli script : compressor.sh, delete.sh

Si è deciso che i compiti di archiviazione su disco e manutenzione dello spazio libero sul disco stesso siano affidati a degli script di shell.

In questo modo si è garantita la massima flessibilità e configurabilità del sistema riguardo la funzionalità di archiviazione.

COMPRESSOR.SH

Questo script per bash si occupa della compressione di un file wf in formato gzip. Ovviamente utilizza il programma linux gzip per compiere la compressione del file. Per ridurre il carico del sistema durante la compressione, che avviene contemporaneamente alla acquisizione ed eventualmente alla trasmissione del dato, si è scelto di non operare la massima compressione ma un livello intermedio che consente però un notevole guadagno in termini di tempi di calcolo, a scapito di una piccola perdita in spazio su disco. Lo script può essere facilmente editato per personalizzare la struttura delle directory in cui si archiviano i file.

Attualmente la struttura consiste in una directory principale /data, su cui è montato il file-system della scheda CompactFlash, all'interno di tale directory è prevista una directory per ogni giorno di archiviazione. Lo schema dei nomi di directory è AAAAMMGG. I file di dati in formato compresso sono suddivisi all'interno di queste directory secondo il giorno di inizio registrazione del file. Le operazioni svolte da *compressor.sh* vengono registrate nel file log.txt, presente nella stessa directory principale /data.

DELETE.SH

Questo script per bash si occupa della cancellazione dei file di dati, tenendo conto della percentuale di occupazione dello spazio nel filesystem dove è montata la directory /data. Esiste

cioè una soglia, configurabile nello script, oltre la quale l'occupazione del disco non potrà salire, perchè al suo raggiungimento verranno cancellate le directory dei dati meno recenti. Le directory da cancellare vengono individuate attraverso l'ordinamento lessicografico consentito dallo schema dei nomi AAAAMMGG.

PPUPS

Questo programma serve a controllare lo stato di alimentazione fornita dall'UPS progettato nel laboratorio INGV. Nel caso di una caduta della alimentazione di rete, che porta l'UPS a fornire alimentazione alla stazione GAIA e/o alla coppia Router-Modem, l'apparato avverte di questo stato portando una linea di output a una tensione corrispondente al bit logico 1. Questa linea di output è riportata sulla scheda di alimentazione MASTER della TN-1, e dirottata su un pin della porta parallela della scheda TN-1 stessa. *ppups* testa lo stato della porta parallela alla ricerca del cambiamento di stato della linea. Quando ciò si verifica, invia un messaggio al demone di sistema *syslogd*.

START.SH

Lo script bash *start.sh* serve a lanciare *master2bn* e *splitter* con i giusti parametri.

AGDFD

Lo script *agdfd* serve ad avviare e a terminare tutti i processi relativi alla gestione della scheda AGDF-1. Non fa altro che avvisare il processo *init* di salire o scendere di run-time level in modo da chiudere o riavviare tutti i processi.

SPLITTER

Il programma *splitter* consente la gestione della comunicazione con il server *backnet*. Anche in questo caso si è voluta creare una separazione tra la funzione di collegamento ed il resto degli applicativi. In questo modo tutto ciò che riguarda i problemi di collegamento remoto ha causa o effetto all'interno del software *splitter*, senza effetti collaterali propagati ad altre funzionalità. Questa separazione è valida nel normale funzionamento, e si è dimostrata valida anche nella fase di test e debugging degli applicativi. Poiché il programma di comunicazione IP è stato concepito quando il programma *backnet* era già funzionante assieme al programma *frontnet* per i collegamenti seriali CDN, ci si è adattati alle sue specifiche di comunicazione. Queste prevedevano che *backnet* si comporti da server e che i client che vogliono comunicare dati sismici richiedano l'apertura di porte TCP sul server *backnet* stesso. Inoltre non si utilizza alcun protocollo di negoziazione della comunicazione a livello applicativo, quindi il server accetta connessioni e legge il contenuto inviatogli dal client, riservandosi una analisi sintattica e semantica a livello di pacchetti WF.

Seguendo tale schema di collegamento, il software *splitter* risulta lui responsabile di mantenere la comunicazione con il server *backnet* riaprendo la connessione in caso di problemi alla stessa.

Il programma *splitter* prende in input fino a tre coppie IP-PORT, una di sorgente e due di destinazione. All'avvio apre un socket su un server sorgente e tenta di aprire e mantenere connessi altri due socket di destinazione. Prende i dati dal socket sorgente e li replica su entrambi i socket di destinazione. Le due destinazioni vengono trattate in modo simmetrico in caso di caduta della connessione. Quando una connessione remota cade, il software cerca di riaprirla, mentre continua a scrivere sull'altro socket. Se entrambe cadono cerca di riaprirle entrambe un numero limitato di volte. Se non riesce a stabilire almeno una connessione remota il programma si chiude. Lo stesso accade se cade la connessione locale con *master2bn*.

La logica in questo comportamento è quella di garantirsi a un livello più alto, ovvero nello script *start.sh*, la possibilità di un reset degli applicativi.

In dettaglio il software *splitter* è stato progettato come un automa a stati finiti, i cui stati sono lo stato delle sue connessioni e il passaggio da uno stato all'altro è governata dall'aprirsi o chiudersi delle connessioni stesse (fig. 4).

Lo stato di inizio è **L try** dal quale si passa, in caso di successo, a **LR** o **LA** (gli archi

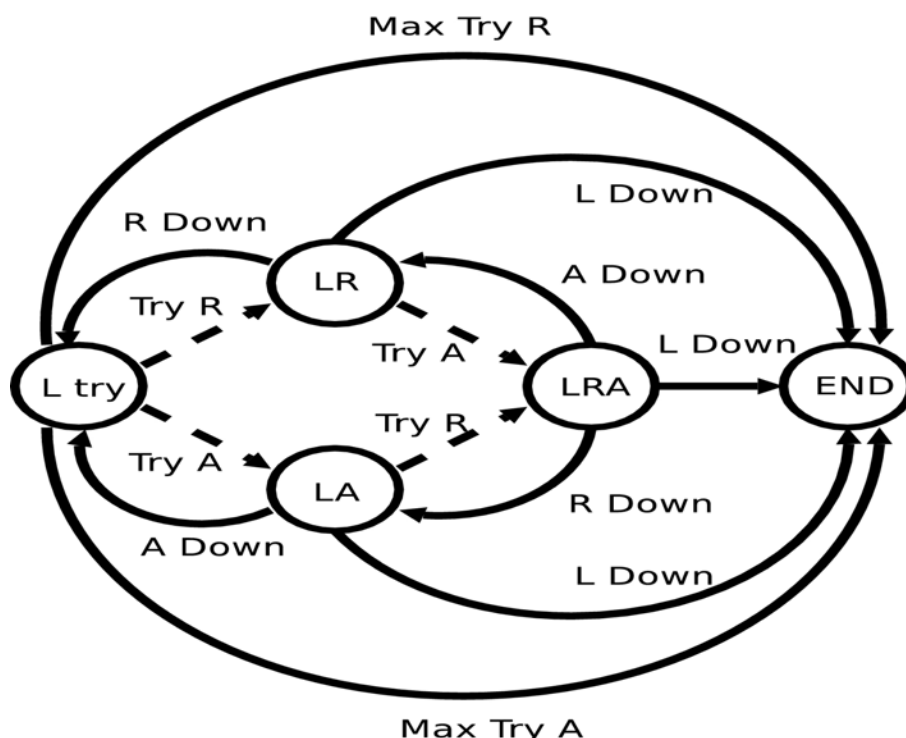


Figura 4: diagramma a stati del programma *splitter*

tratteggiati rappresentano, appunto, i passaggi di stato in seguito alla riuscita entro un numero limitato di tentativi **Max try**). Si noti che se nello stato **L try** si supera il limite di tentativi **Max try** (**R** o **A**), il programma si chiude.

Sintassi del comando *splitter-1.3*:

splitter-1.3 [OPTIONS] destination-host port source-host port [auxiliary host port]

OPTIONS

-r N, --retry=N

Numero di tentativi di riapertura dei socket destinazione sul socket close, 0 for infiniti tentativi

-p N, --pause=N

Millisecondi di pausa prima di tentare di riaprire il socket.

-l N, --local=N

Secondi di attesa prima della lettura sul socket di input

-a , --aux

Indica la presenza della destinazione ausiliaria

-V, --version

Stampa informazioni sulla versione

-h, --help

Stampa dell'help.

Installazione

1. Introduzione

In questo capitolo verranno descritti i passi per l'installazione, la configurazione e i test relativi alla scheda TN-1. Per far ciò si parte dalla descrizione della scheda di alimentazione della TN-1, si affrontano le varie configurazioni hardware descrivendo nel dettaglio i collegamenti, per finire con la "personalizzazione" software della scheda TN-1 secondo i parametri richiesti dalla stazione sismica.

2. Descrizione della scheda di alimentazione MASTER

La scheda di alimentazione (vedi fig. 5) MASTER, interamente sviluppata nei laboratori INGV, prevede la presenza di un alimentatore switching che, con i 12 V in ingresso, alimenta la TN-1 a 5 V attraverso il connettore PC104. Nella scheda è stata aggiunta tutta una serie di connettori per le

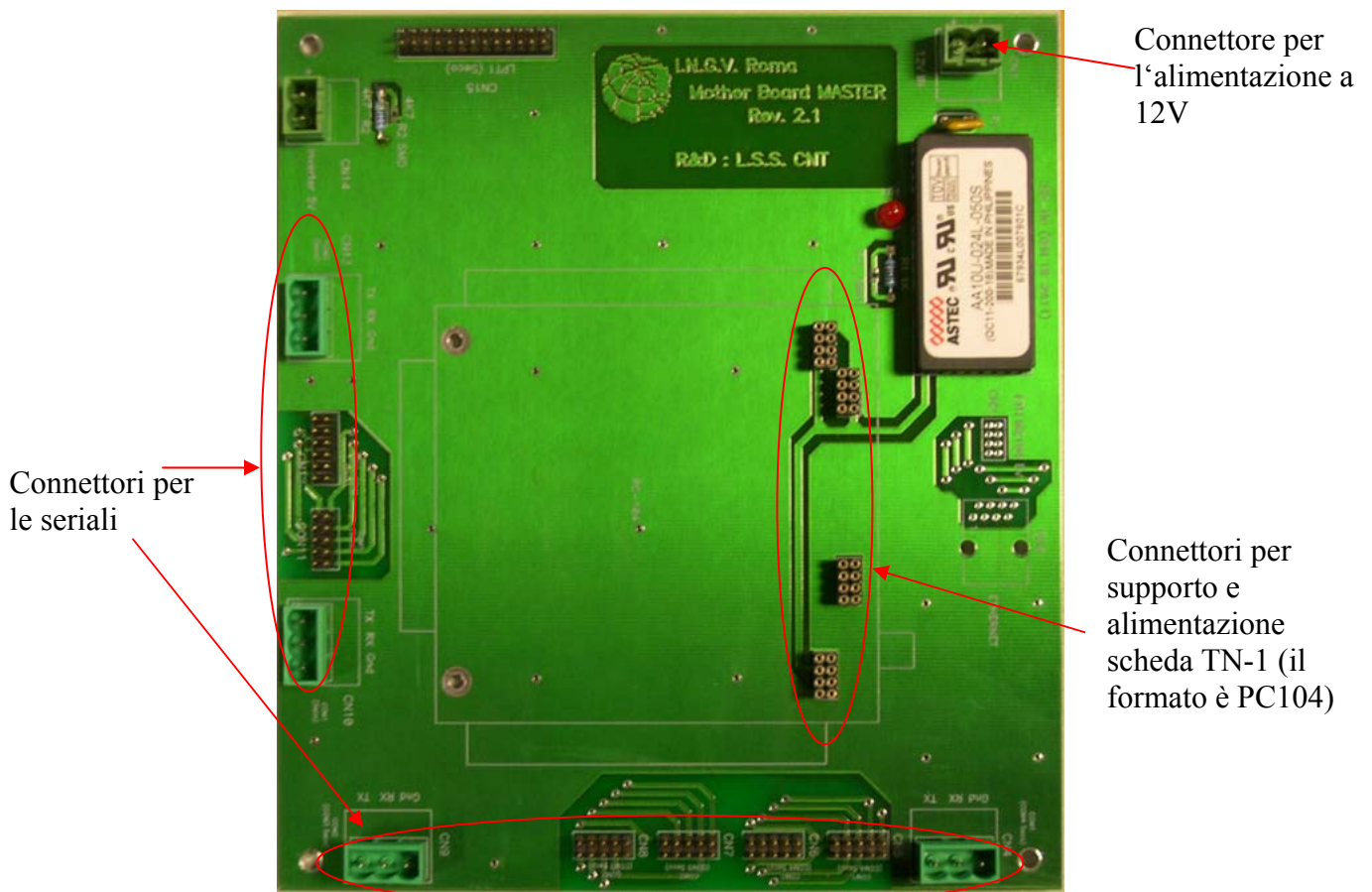


Figura 5: scheda MASTER

seriali, la parallela e per i collegamenti con la scheda di acquisizione AGDF1.

3. Procedura di assemblaggio con la scheda di alimentazione

Nella fig. 6 si può vedere la configurazione minima di funzionamento per la scheda TN-1.

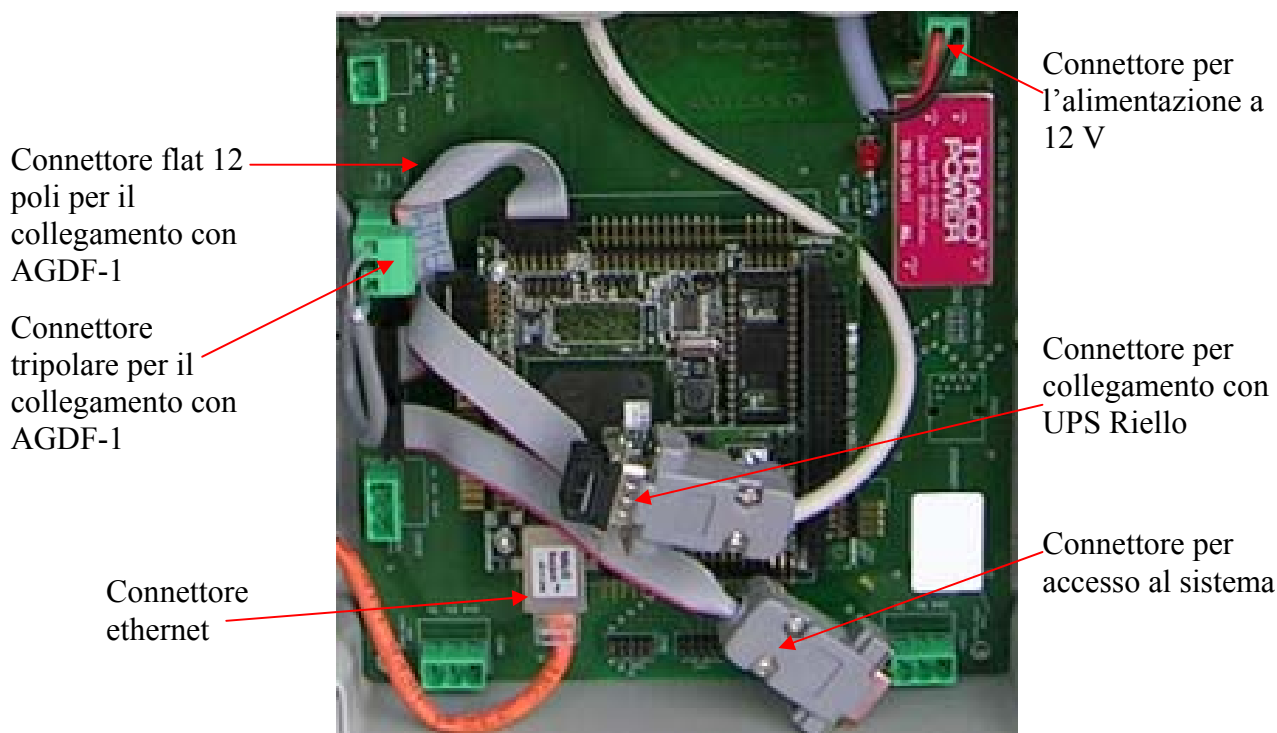


Figura 6: configurazione minima per la TN-1

Il montaggio prevede l'inserimento della scheda TN-1, con relativo DOM con sistema operativo preinstallato, nei connettori appositi e il collegamento ai connettori (ethernet, seriali e parallela) secondo la figura. Nello specifico al connettore CN12 va collegato, tramite una piattina a 12 fili, il connettore COM A della TN-1 (prima e seconda porta seriale). Al connettore CN11 va collegato, tramite piattina a 10 fili, un connettore db9 che servirà come seriale di servizio consentendo l'interazione col BIOS prima e con il sistema operativo installato, poi. Ai connettori CN5 e CN7, sempre con piattine da 10 fili, vanno collegati i connettori COM C e COM B della TN-1 (terza e quarta porta seriale). Ai connettori CN13, CN9 e CN4 vanno connesse, con cavi schermati trifilari, le eventuali schede AGDF-1 seguendo l'ordine indicato. Infine al connettore CN1 va collegata, con un cavo bipolare, l'alimentazione a 12 V.

Per il monitoraggio dell'alimentazione da parte della TN-1, esistono due possibilità: utilizzo dell'inverter INVERT1 progettato nei laboratori INGV collegando il connettore CN15 al connettore LPT della TN-1 (porta parallela); utilizzo degli UPS della Riello S.p.A. collegando il connettore COM C della TN-1 al connettore CN12 della Master. Per far ciò bisogna dotare la TN-1 di un connettore DB9 maschio con una piattina a 10 pin. Gli ups della Riello S.p.A. sono dotati di un cavo adatto al collegamento seriale con un pc, quindi è sufficiente usare quest'ultimo per connettersi alla TN-1.

4. Procedura di assemblaggio della scheda adattatore compact flash eide

Rispetto alla configurazione base, vi è la possibilità con la TN-1 dell'archiviazione su disco dei dati in arrivo dalle seriali. Per poter fare ciò si aggiunge al sistema una scheda (in fig. 7 è in formato PC104, ma non è l'unico) per l'alloggiamento di una Compact Flash (CF). In questo momento la configurazione prevede una CF da 1 GB. Il collegamento con la TN-1 è tramite cavo flat da 40 pin sul connettore IDE dato che la CF viene riconosciuta dal sistema come un vero e proprio hard disk.

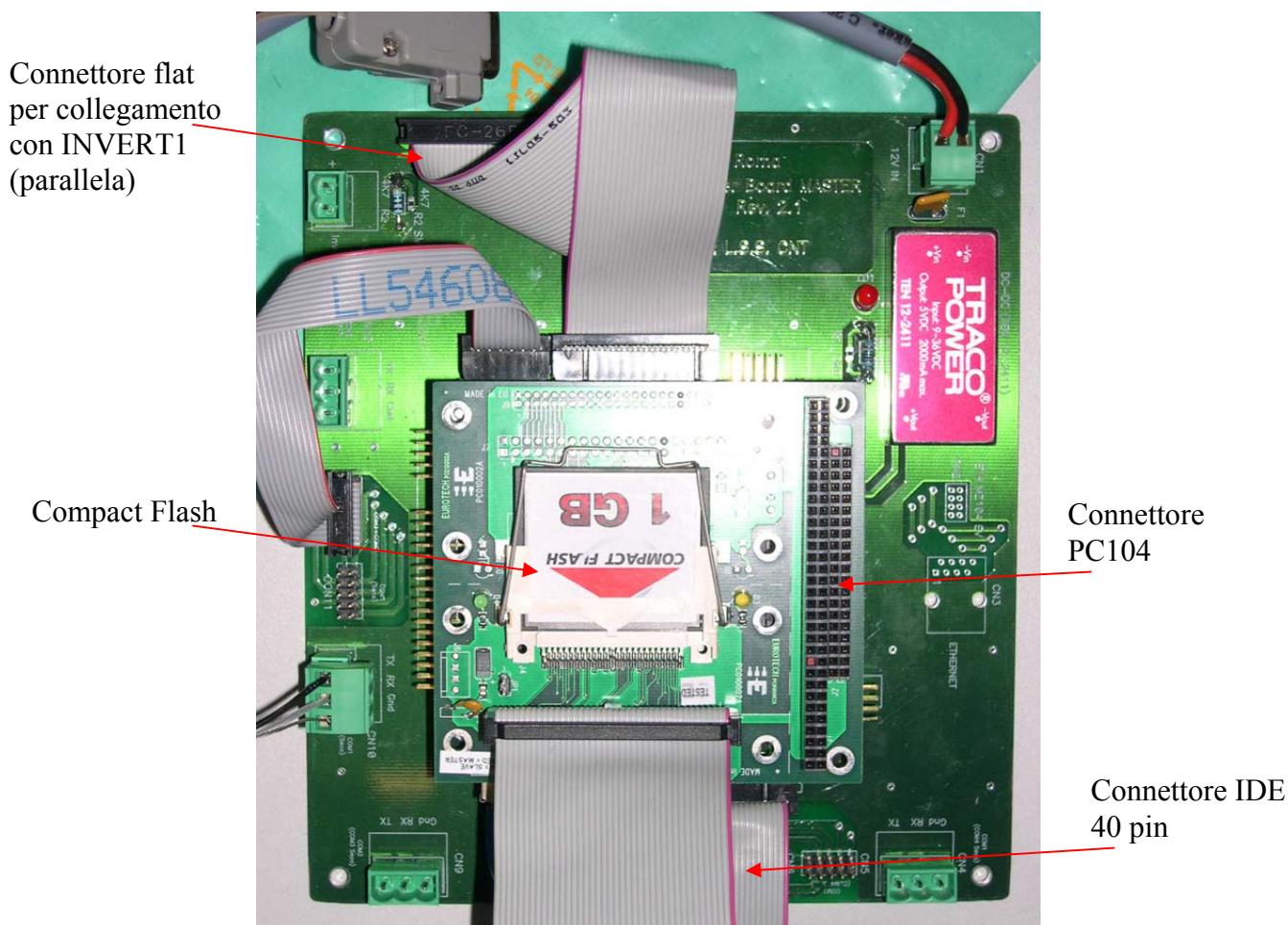


Figura 7: configurazione con Compact Flash

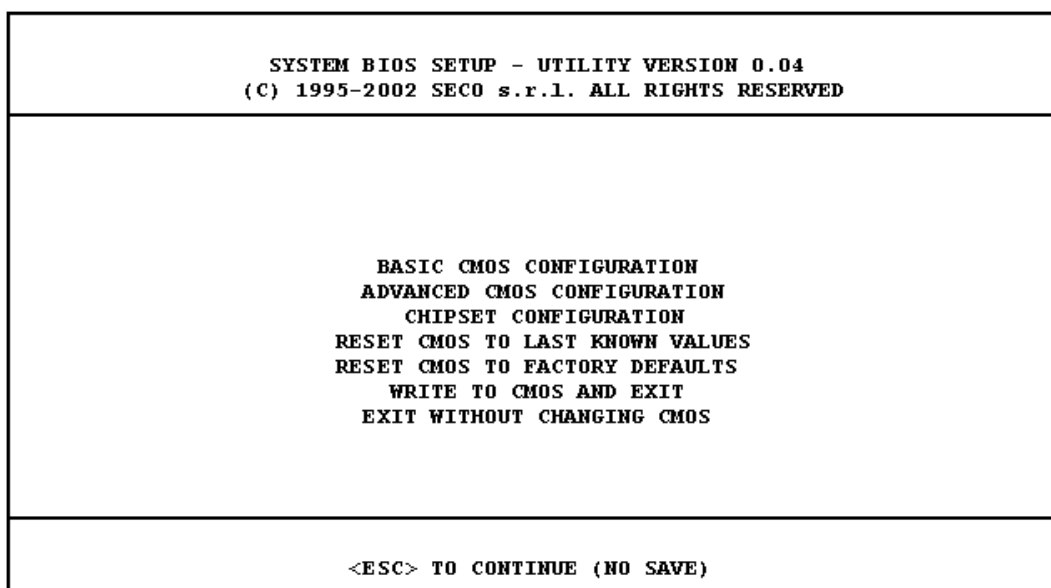
L'unico accorgimento da tenere è la verifica della configurazione di tipo Slave della CF, in genere effettuabile tramite un ponticello sulla scheda di alloggiamento.

5. Bios Setup

Per la TN-1 è necessario fare delle modifiche ai parametri di default del BIOS, nate dalla necessità di utilizzare la porta parallela come una semplice porta di I/O (nel collegamento a INVERT1) e non per stampante, e per poter accedere al BIOS attraverso la seriale (l'utilizzo normale della TN1 non prevede né monitor né tastiera).

Si accede al setup premendo il tasto (<CANC>) oppure i tasti <CTRL>+<ALT>+<ESC> all'avvio del sistema.

Apparirà il menù come il seguente:



La selezione delle varie voci avviene con l'uso dei tasti freccia **su** e **giù** ed <INVIO>, con <ESC> si torna al menù superiore.

- Selezionare **ADVANCED CMOS CONFIGURATION** e modificare l'impostazione "**Console on COM1**" su **enabled**. Questa è l'opzione che ci consente il reindirizzamento del BIOS sulla prima seriale alla velocità di 9600 b/s.
- Selezionare **CHIPSET CONFIGURATION**, modificare l'impostazione "**Internal LPT1 mode**" su **EPP** e il "**mode**" su **1.9**.
- Salvare le modifiche selezionando **WRITE TO CMOS AND EXIT**

Dopo queste modifiche la scheda video risulta superflua dato che tutto il sistema è reindirizzato anche sulla prima seriale e, quindi, tutte le operazioni che seguono possono essere effettuate connettendosi alla scheda TN-1 con un cavo seriale *null-modem* **completo** usando un emulatore di terminale seriale, come Hyperterminal®, settato a 9600 b/s 8n1.

6. Configurazioni della scheda

- *funzionamento con RUPA*

La Rete Unitaria della Pubblica Amministrazione (RUPA), dal nostro punto di vista, è essenzialmente un'estensione a livello geografico della Intranet dell'INGV. Questo significa che PathNet, la società che gestisce RUPA, fornisce ogni stazione di un Router "periferico" che è visibile attraverso un Router "centrale" presente nella sede INGV.

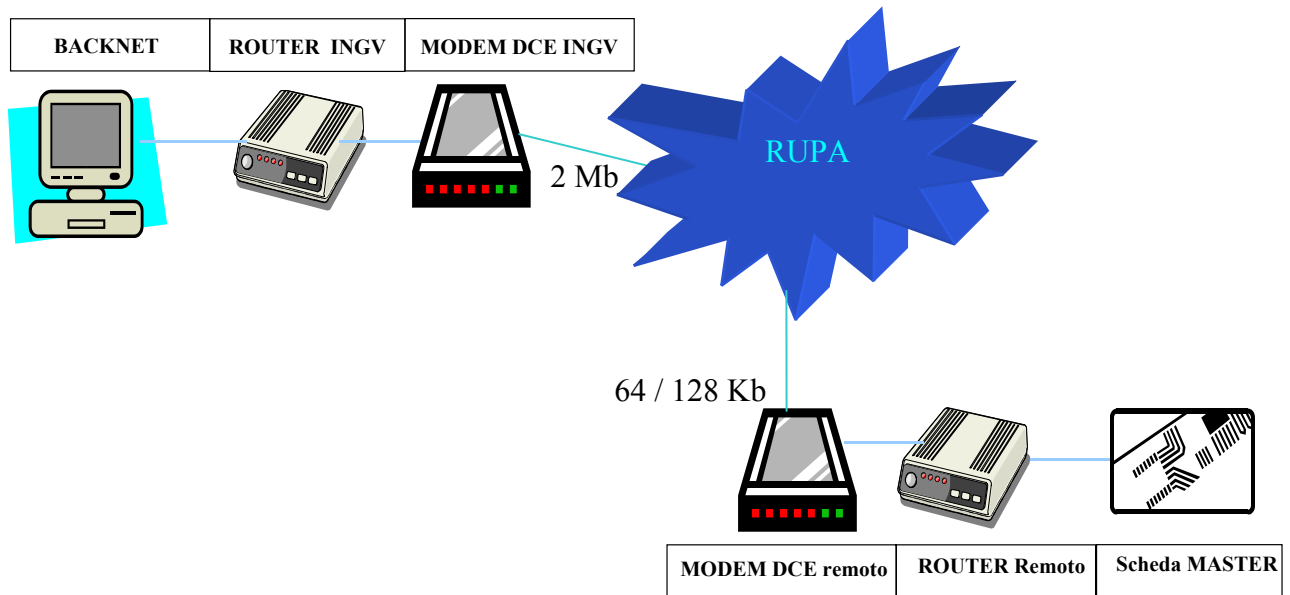


Figura 8: Schema del collegamento RUPA

Rispetto al CDN il MODEM è un apparato trasparente per l'INGV, la nostra attenzione è rivolta al Router. Il router è un apparato che consente il collegamento fra reti di tipologie differenti. Nel nostro caso adatta una rete con protocollo TCP/IP ad un CDN (tramite modem DCE plus fino a 64 kbit/sec) o X-DSL (modem DCE3 da 128 kbit/sec in su). La stazione remota è essenzialmente una piccola LAN (Local Area Network) dotata di 8 indirizzi IP (di questi quelli effettivamente utilizzabili dall'INGV sono 5): il primo indirizzo è relativo alla rete, il secondo è per il router, l'ultimo è il broadcast. Nello specchio si vede la configurazione della stazione di Montasola (RI):

Subnet (rete)	10.136.0.0
Subnet Mask	255.255.255.248
Primo IP	10.136.0.1
Ultimo IP	10.136.0.6
Broadcast	10.136.0.7
Router (gateway)	routermns.int.ingv.it IP 10.136.0.1

La stazione tipo prevede una sola scheda TN-1 collegata al Router, per cui un unico indirizzo IP utilizzato; nel caso di più schede collegate è necessario installare un hub (switch).

Per la configurazione della scheda in trasmissione IP è necessario, quindi, conoscere il nome del computer (generalmente coincidente con il nome completo della stazione), la configurazione di rete

(vedi specchietto precedente), la configurazione dell'applicativo (vedi oltre).

Eseguito lo script *tn1-config* verranno configurati i file per il corretto funzionamento di tutto il sistema. Si eseguirà pertanto, come prima cosa, la modifica del nome della stazione, seguirà la configurazione IP del TN-1 e relativi servizi, e l'avvio dell'applicativo.

A tal scopo verranno utilizzati i programmi linux “**mount**”, “**vi**”, “**init**” e INGV “**setup**”. Alla fine della testo trova posto una rapida descrizione e utilizzo dei comandi linux (tra cui anche ifconfig – programma per la configurazione rapida della rete).

- **Funzionamento in ppp**

La configurazione di questo collegamento è simile alla precedente perché il protocollo è sempre TCP-IP anche se si passa su seriale. Il collegamento PPP su seriale con un pc configurato opportunamente per questo protocollo, è simile al collegamento con il router del collegamento RUPA. L'interfaccia su cui lavora PPP necessita di un indirizzo IP, come una qualunque scheda di rete, sia lato TN-1 che lato PC. Quest'ultimo, per la TN-1, si comporta a tutti gli effetti come un router e sarà proprio il suo indirizzo IP-PPP ad essere utilizzato come gateway (vedi specchietto del funzionamento RUPA). Il file di configurazione **options** si trova nella directory **/etc/ppp** e va modificato a mano. Nella stessa directory vi è stato messo uno script con la riga di comando per lanciare il demone **pppd**:

```
pppd 192.168.0.3: 192.168.0.1 /dev/ttyS3 9600
```

dove il primo indirizzo IP è quello del PC-router, il secondo è della TN-1

- **funzionamento a registrazione locale.**

Il funzionamento a registrazione locale non prevede in genere una trasmissione, per cui in teoria non sarebbe necessario utilizzare lo script *tn1-config* e comunque la configurazione di default è completa. In pratica si può sempre utilizzare lo script per modificare il nome della stazione e fare il setup degli applicativi. La cosa importante è il parametro dell'archiviazione dei dati relativo alle singole schede collegate alla TN-1 configurato con *i-setup*.

- **funzionamento Seedlink.**

Nella scheda TN-1 è stato testato il server SeisComP (Seismological Communication Processor). A tal fine sono stati modificati, come detto, gli applicativi per poter comunicare col server, l'unica configurazione da eseguire riguarda i due file di cui si è discusso nel paragrafo per il funzionamento su RUPA. È stata testata la connettività col server e la visualizzazione in tempo reale dei segnali sismici utilizzando un'applicazione java, SeisGram2k, di Antony Lomax (fig. 4). Ulteriori test saranno relativi alla memorizzazione locale, all'accesso e al trasferimento dati in dial-up, oltre che, ad esempio, al processamento dei dati sismici in locale.

Nel dettaglio i comandi che lo script *tnl-config* esegue sono i seguenti (l'utente deve fare le modifiche del caso descritte sotto ogni comando - premere "i" per entrare in modalità "insert" e dopo le modifiche, per il salvataggio delle stesse, premere in sequenza i tasti <ESC>, due punti ":", "x" e <invio> -):

- "vi /etc/HOSTNAME", <invio>

(tale comando modifica il file HOSTNAME = nome del PC)

- "vi /etc/rc.d/rc.inet1", <invio>

(tale comando modifica il file rc.inet1 che è il file di configurazione IP di linux)

Modificare i campi evidenziati con i dati della stazione

```
# IF YOU HAVE AN ETHERNET CONNECTION, use these lines below to configure the
# eth0 interface.
```

```
# Edit these values to set up a static IP address:
```

```
IPADDR="10.100.10.39" # REPLACE with YOUR IP address!
```

```
NETMASK="255.128.0.0" # REPLACE with YOUR netmask!
```

```
NETWORK="10.30.0.0" # REPLACE with YOUR network address!
```

```
BROADCAST="10.30.255.255" # REPLACE with YOUR broadcast address, if you
# have one. If not, leave blank and edit below.
```

```
GATEWAY="" # REPLACE with YOUR gateway address!
```

(N.B. il GATEWAY, nel nostro caso, è l'indirizzo del router)

- "vi /etc/dhcpd.conf", <invio>

(eseguendo tale comando si va a modificare il file di configurazione per il server DHCP per permettere la connessione ad un portatile, con la rete configurata in modalità dinamica, attraverso il cavo incrociato ethernet).

Modificare i campi evidenziati scrivendovi la rete (subnet) e l'ultimo indirizzo IP disponibile per la stazione:

```
# Configurazione delle Subnet
```

```
#Il tempo di lease è 3 giorni
```

```
default-lease-time 259200;
```

```
#Il tempo massimo per cui si può utilizzare l'indirizzo affittato è 6 giorni
```

```
max-lease-time 518400;
```

```
#Definizione della sottorete da configurare per RUPA
```

```
subnet 10.136.0.0 netmask 255.255.255.248 {  
  range 10.136.0.6; #Indirizzo del portatile  
}
```

- “vi /etc/inittab”, <invio>

(tale comando modifica il file inittab, che descrive quali processi sono avviati all'avvio e durante le normali operazioni, per l'avvio dell'applicativo INGV)

modificare il campo evidenziato: togliere il cancelletto (#) .

```
# for 'getty_ps' you use line, linespeed and also use 'gettydefs'
```

```
c1:1235:respawn:/sbin/agetty 38400 tty1 linux
```

```
c2:1235:respawn:/sbin/agetty 38400 tty2 linux
```

```
c3:1235:respawn:/sbin/agetty 38400 tty3 linux
```

```
#script TN-1
```

```
#star:23:respawn:/applicativo/start.sh
```

```
# Serial lines
```

```
#s1:12345:respawn:/sbin/agetty 19200 ttyS0 vt100
```

```
s2:12345:respawn:/sbin/agetty 38400 ttyS1 vt100
```

- “vi /seedlink-ridotto/config/seedlink.ini”, <invio>

(tale comando modifica il file di configurazione seedlink.ini del server SeisComP)

modificare i campi evidenziati: scrivere la sigla della stazione.

```
plugin TN1 cmd="/applicativo/master2bn-seed /applicativo/config.staz 99999 &"
```

```
  timeout = 600
```

```
  start_retry = 60
```

```
  shutdown_wait = 10
```

```
station BASE description = "INGV"
```

```
  name = BASE
```

```
  network = IV
```

```
  proc = INGV_3x50
```

- “vi /seedlink-ridotto/key/key_BASE”, <invio>

(tale comando modifica il file key_BASE necessario al server SeisComP per l’acquisizione del segnale sismico)

modificare il campo evidenziato: scrivere la sigla della stazione.

KEY_VERSION='2.0'

STATION=**BASE**

DESC='INGV'

NET='IV'

SEGID='9600'

DIGITYP='ps2400_3'

PLUGIN='ps2400_3'

SPROC='yes'

- “/applicativo/i-setup”, <invio>

(seguire attentamente le indicazioni del programma di configurazione della stazione, in caso di errore abortire premendo <CTRL> + C)

Il programma *i-setup* permette all'utente di configurare la stazione interattivamente rispondendo a una serie di domande relative a: numero di schede AGDF1 collegate alla TN-1 e loro velocità di trasmissione, numero di canali delle singole schede, necessità o meno di archiviare i dati, necessità o meno di spedire i dati delle singole schede, tipo di telemetria utilizzata (ethernet e/o seriale).

Un esempio di una sessione del programma è la seguente:

*****Programma di configurazione Stazione*****

*****Ctrl+c per terminare*****

Numero schede collegate: 1

Telemetria Ethernet?(y/n): y

Baud rate output (b/s) - 0 = no telemetria seriale -: 19200

Nome 1a scheda - COM 1 -(5 caratteri, completare con _): BASE_

Baud rate 1a scheda -COM 1 - (b/s): 19200

Numero di canali 1a scheda - COM 1 -: 3

Da trasmettere (y/n) 1a scheda - COM 1 -? y

Grandezza file di log in minuti (da 0 = no archiviazione a 60): 30

Al termine della immissione dei parametri il programma presenta in output un riassunto della

configurazione appena impostata:

Nome 1a scheda - COM 1 : BASE_
Baud rate 1a scheda -COM 1 - (b/s): 19200
Numero di canali 1a scheda - COM 1 : 3
Da trasmettere 1a scheda - COM 1 - y
Grandezza file di log in minuti : 30
Da memorizzare : y
Stazione configurata.

Si ricorda che esiste anche una seconda modalità di funzionamento del programma “*setup*” che prevede l’immissione dei parametri direttamente da riga di comando e non in modalità interattiva, la cui sintassi è:

“setup eth baud_rate_out board_name baud_rate num_of_ch tx file_size ...”

Test

1. Introduzione

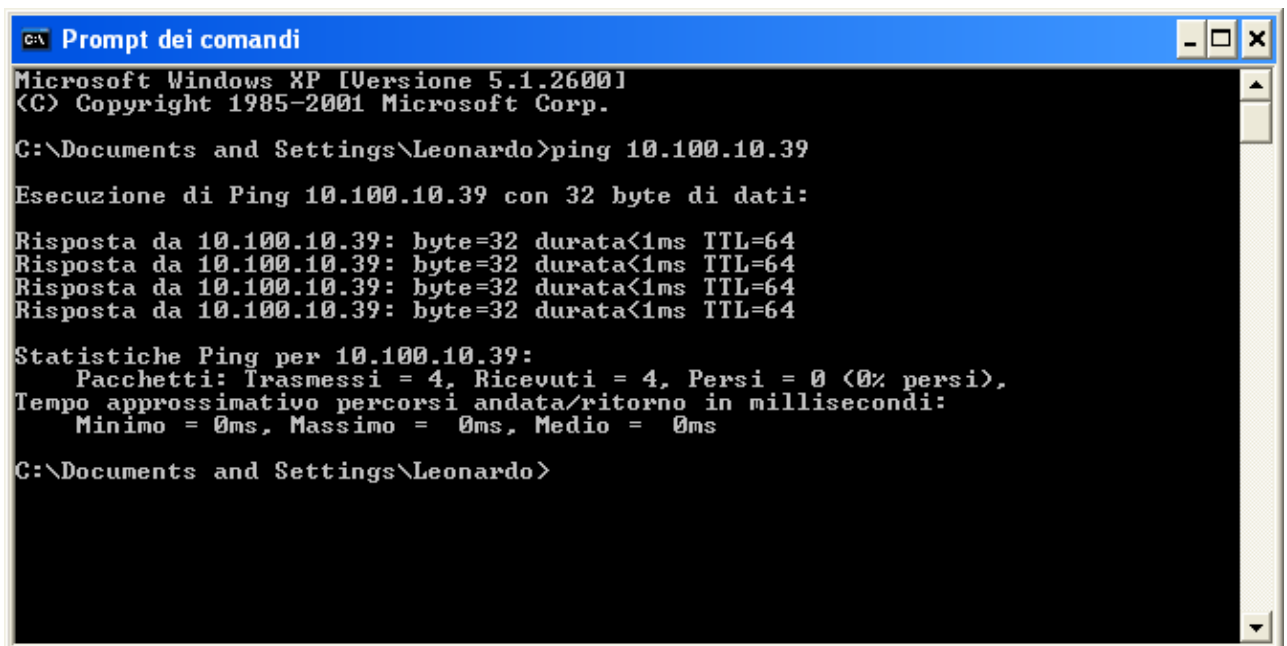
Questo capitolo verterà sui vari test che si possono eseguire sulla TN-1. Dapprima i test di funzionamento dei server che mette a disposizione il sistema operativo Linux, seguiti da quelli relativi agli applicativi INGV, per i quali è necessario l'utilizzo di una stazione GAIA completa, cioè TN-1 connessa ad una AGDF-1 dotata di modulo GPS e sensore sismico, e ad un UPS Riello.

2. Test delle connessioni di rete

- **Raggiungibilità: ping**

La configurazione di default della scheda di rete utilizza il modulo 8139too per la scheda RealTek RTL8139. L'indirizzo IP di default è il 10.100.10.39 con netmask 255.128.0.0. Gli utenti abilitati all'accesso sono root e turnista.

Collegando la TN-1 ad un computer client direttamente con un cavo Ethernet incrociato è possibile testare contemporaneamente la connessione attraverso l'esecuzione di un ping, ed il server DHCP. A tal proposito è necessario un pc client in grado di richiedere tale servizio, altrimenti il client deve avere un indirizzo IP della stessa rete del TN-1 (10.X.X.X). Avviato il sistema ed atteso il prompt dei comandi, eseguire il comando **ping 10.100.10.39** dal computer client (fig).



```
c:\ Prompt dei comandi
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Leonardo>ping 10.100.10.39

Esecuzione di Ping 10.100.10.39 con 32 byte di dati:

Risposta da 10.100.10.39: byte=32 durata<1ms TTL=64
Risposta da 10.100.10.39: byte=32 durata<1ms TTL=64
Risposta da 10.100.10.39: byte=32 durata<1ms TTL=64
Risposta da 10.100.10.39: byte=32 durata<1ms TTL=64

Statistiche Ping per 10.100.10.39:
    Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 0ms, Massimo = 0ms, Medio = 0ms

C:\Documents and Settings\Leonardo>
```

Figura 9: screenshot dell'esecuzione del ping

- **Connessione e accesso al sistema: telnet**

Non appena il collegamento di rete è attivo, è possibile l'accesso al sistema con il computer

client attraverso l'esecuzione del comando **telnet 10.100.10.39**. Il logon di linux (fig) apparirà sul client e sarà sufficiente mettere userid e password.

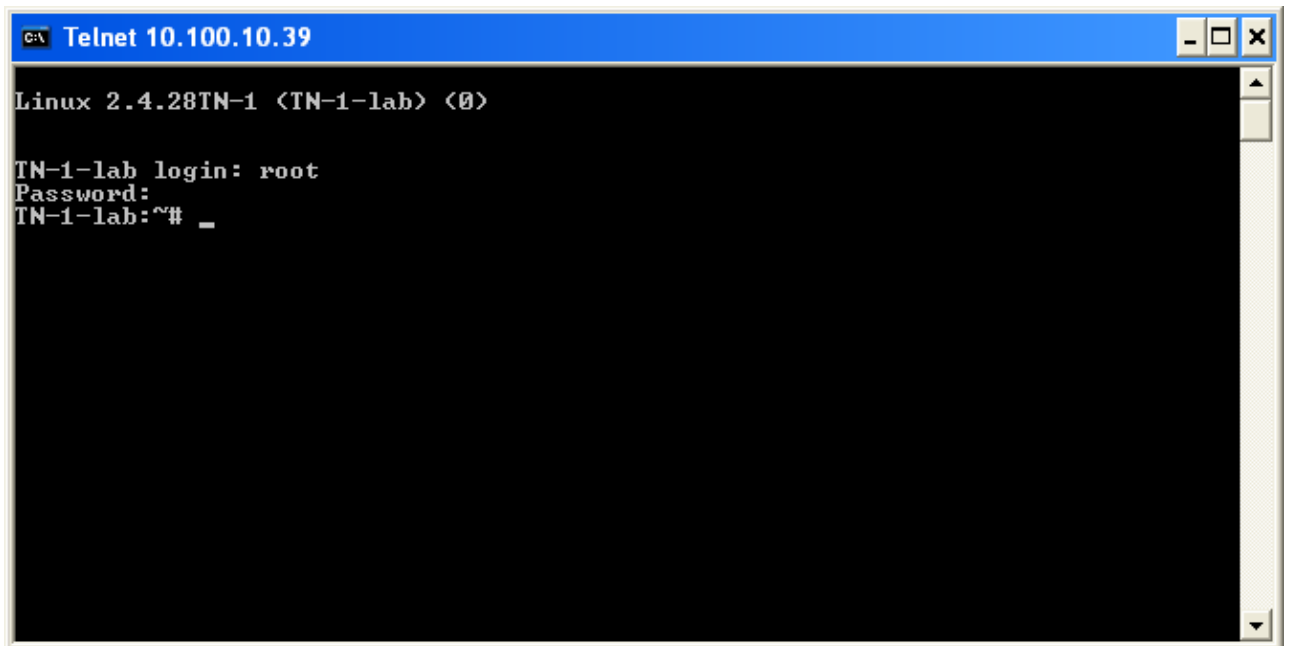


Figura 10: screenshot dell'esecuzione del telnet

- **Connessione e accesso al sistema: SSH**

Secure Shell è un login remoto sicuro, dal funzionamento simile al telnet, ma con tutte le informazioni spedite criptate. Per il test, il computer client necessita di un programma SSH client, ad esempio SSHWin-2.4.0.exe per Windows e ssh-2.4.0.tar.gz per Unix disponibili all'indirizzo:

<http://commerce.ssh.com/sshws/index.html> e <http://www.openssh.org>.

Sia da Windows sia da Unix è sufficiente lanciare il client ssh con l'indirizzo IP 10.100.10.39 e accedere con userid e password.

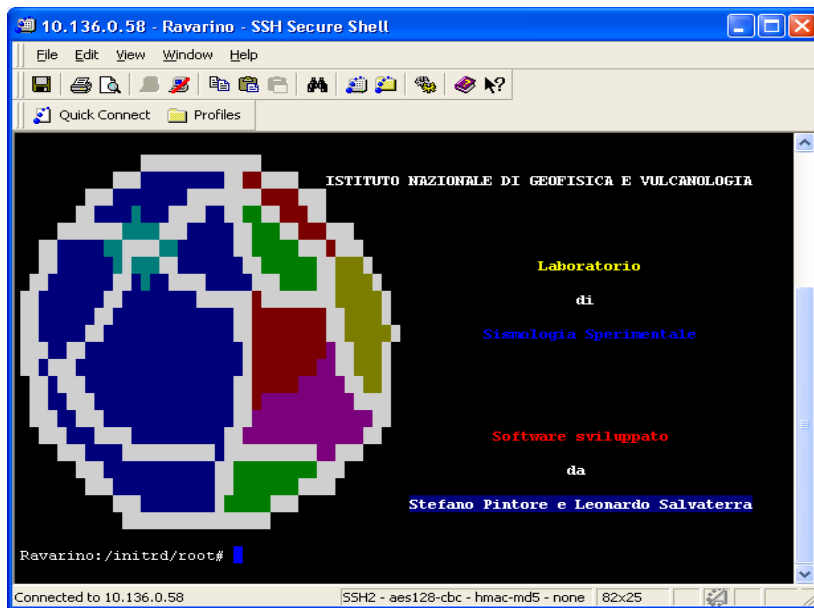


Figura 11: screenshot dell'esecuzione di SSH

- **Connessione e accesso al sistema: SFTP**

Il Secure File Transfer (SFTP) permette il trasferimento di file tra il sistema TN-1 e un computer client come il protocollo FTP, ma in modalità sicura con tutte le informazioni spedite sulla rete criptate. Per il test è necessario un programma SFTP sul computer client. I programmi SSHWin-2.4.0.exe e ssh-2.4.0.tar.gz citati sopra ne sono provvisti.

L'accesso è simile a quello con ssh e, con SSHWin (fig), si può utilizzare la possibilità del drag and drop per facilitare il trasferimento dei file.

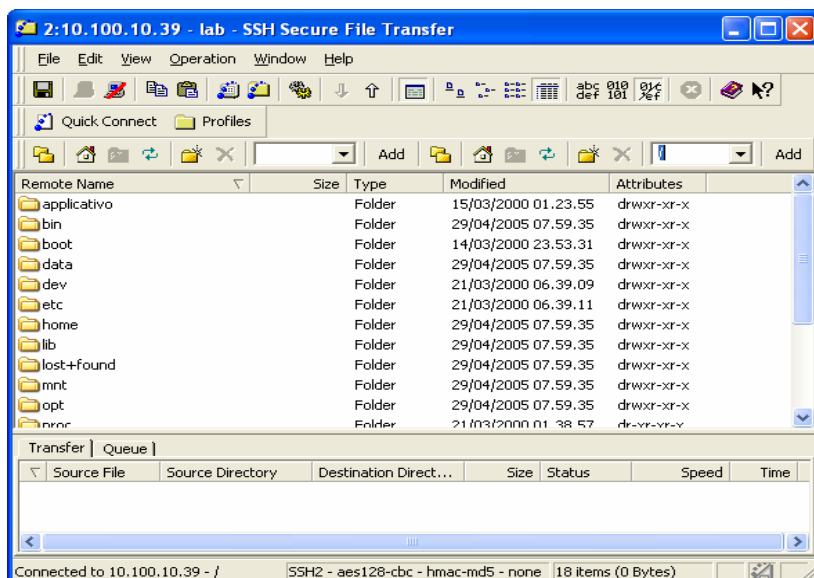


Figura 12: screenshot dell'esecuzione di SFTP

- **Connessione al sistema: HTTP**

Per effettuare questo test è sufficiente immettere in un browser web, del pc client connesso alla TN-1 con il cavo ethernet, l'indirizzo IP 10.100.10.39. Si ottiene una schermata come in figura.

L'homepage visualizzata è molto semplice, ma ha un link (Segnali sismici in tempo reale) che permette il collegamento al server SeisComP attraverso un'applicazione java (SeisGram2k di Antony Lomax) risiedente nella TN-1 e che viene scaricata sul pc client, per la visualizzazione in tempo reale dei segnali sismici.



Figura 14: : screenshot dell'homepage della TN-1

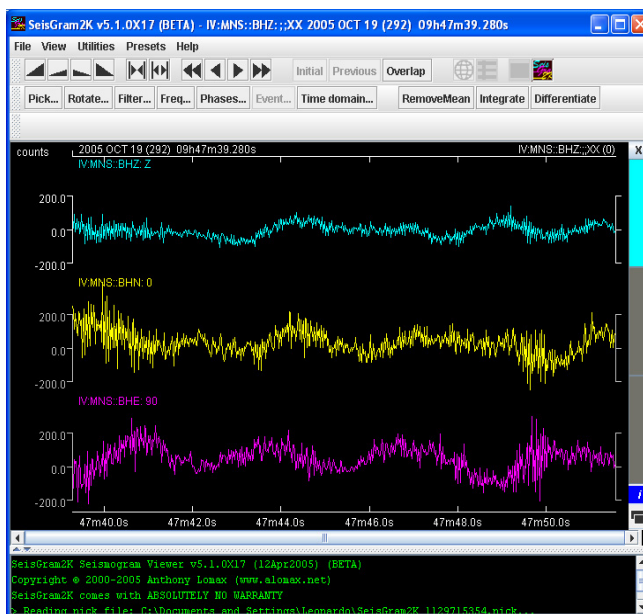


Figura 13: screenshot dell'esecuzione di SeisGram2k

- **Connessione al sistema: NFS**

La directory /DATA può essere montata via NFS usando il comando (UNIX):

mount 10.100.10.39 ://DATA .

Ovviamente il client computer deve avere il client NFS correttamente configurato per eseguire questo test.

Il file di configurazione di default sulla TN-1 è /etc/exports. Per modifiche alle impostazioni del servizio bisogna riferirsi a questo file.

- **Connessione al sistema: SMB**

Il sistema è anche accessibile con Windows Explorer di un computer client dotato di S.O. Windows, attraverso il protocollo usato per le Reti Microsoft. Il sistema (Samba) chiede la prima

volta, per l'accesso, utente e password: l'unico attualmente abilitato è l'utente turnista. Si accede alla directory /DATA, mappata, laddove è installata la compact flash, su /dev/hdb1.

La configurazione del server Samba utilizza i file /etc/smb.conf, /etc/smbusers, /etc/smbpasswd.

3. Test della connessione seriale

Linux consente l'accesso al sistema anche attraverso una seriale grazie alla redirectione della console (si ricorda che anche il BIOS può essere reindirizzato). Per il test della seriale di servizio, quindi, è necessario un computer client, connesso alla TN-1 con un cavo *null-modem completo*, dotato di un terminale seriale (ad esempio Hiperterminal® di Windows, Minicom di Unix) e accedere al sistema con utente root e password standard (si ricorda la connessione a 9600 b/s 8N1).

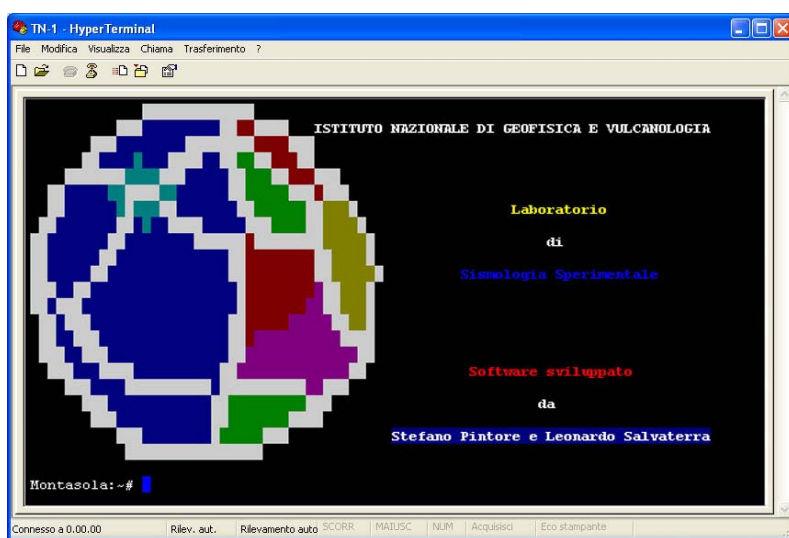


Figura 15: screenshot del sistema con terminale seriale

4. Test della stazione

Questi test servono per la verifica della corretta programmazione della TN-1 e dell'hardware in vista dell'installazione in stazione.

Dopo aver acceduto al sistema (da terminale seriale o via SSH), si verifica il corretto collegamento con la scheda AGDF1 eseguendo il comando "**cat /dev/ttyS1**". A video devono apparire i pacchetti sismici spediti dalla AGDF1. Se vi sono connesse più schede AGDF1, ripetere il comando precedente usando ttyS2 e ttyS3.

N.B. Spesso il comando precedente "corrompe" la visualizzazione dei caratteri, in tal caso è necessario riavviare il terminale e riaccedere al sistema.

Per il controllo della configurazione dell'applicativo si eseguono i comandi: "**init q**" (comando per far eseguire l'applicativo); "**ps ax | grep applicativo**" (tale comando visualizza l'elenco dei programmi che stanno "girando" sul sistema). Dovrebbe apparire una schermata simile:

```

15378 ?    S    0:00 /applicativo/master2bn-seed-com1 applicativo/config.staz 99999
15382 ?    S    0:00 /applicativo/splitter-1.3 10.30.5.20 63003 127.0.0.1 99999
15383 ?    S    0:00 /applicativo/master2bn-seed-com1 applicativo/config.staz 99999
15384 ?    S    0:00 /applicativo/master2bn-seed-com1 applicativo/config.staz 99999

```

Nota: nelle righe devono comparire un numero di “master2bn-seed-com1” (programma chiamato precedentemente master2bn) pari a 2 più il numero di AGDF1 collegate (nell’esempio di sopra compaiono 3 righe con “master2bn-seed-com1”, ciò significa che è collegata un AGDF1 sola). Nel caso ciò non succeda bisogna rieseguire il setup della scheda.

Controllo del collegamento del TN-1 con l’UPS-RIELLO:

“/opt/upsmon/upsview“ (tale comando esegue uno script per il controllo dell’ups con possibilità di test dello stesso)

A video apparirà una schermata iniziale che comunicherà la necessità di selezione dell’ups:

UPSAgent funzionante : Si

Free Linux version !

```

-----
UPSView      UPSMON Monitor      Version 4.2 (007)
-----

```

UPS non selezionato!

----- MENU (tasto+INVIO) -----

S-seleziona UPS 1-5-viste (B,S,D,S,A) 0-esci dal programma

E-vedi registraz. D-vedi datalog altri tasti aggiorna

Funzione-F

Seleziona la funzione :

Scegliere l’opzione S, quindi 1 (UPS controllato dall’Upsagent locale), ancora 1 (primo UPS dell’elenco) e si avrà una schermata simile in cui, evidenziate, compaiono le informazioni sullo stato del collegamento con l’ups, della batteria del carico, ecc. :

```

-----
UPSView      UPSMON Monitor      Version 4.2 (007)
-----

```

Nome UPS: MGR Tipo: GPSE11201RU

Connessione: Seriale Tipo.com: GP SER

Stato: [0000] On Line

[Da rete] [Batteria OK] [Carico OK] [Normale]

----- MENU (tasto+INVIO) -----

S-seleziona UPS 1-5-viste (B,S,D,S,A) 0-esci dal programma

E-vedi registraz. D-vedi datalog altri tasti aggiorna

Funzione-F

Seleziona la funzione :

Scegliendo le viste 2 o 3 si hanno ulteriori informazioni, con F si possono far eseguire dei test all'UPS (in particolare quello della batteria), con 0 si esce dal programma.

5. Aggiornamento del firmware dell'AGDF1

Il sistema operativo della TN-1 è dotato di Minicom, un terminale seriale, che consente di collegarsi alle schede AGDF1 connesse per effettuare controlli, ma soprattutto l'aggiornamento firmware. È una caratteristica utile perché può essere sfruttata anche da remoto senza dover essere in stazione. Dopo aver lanciato il programma premendo <CTRL> + A e poi Z si accede all'help come in figura 16.

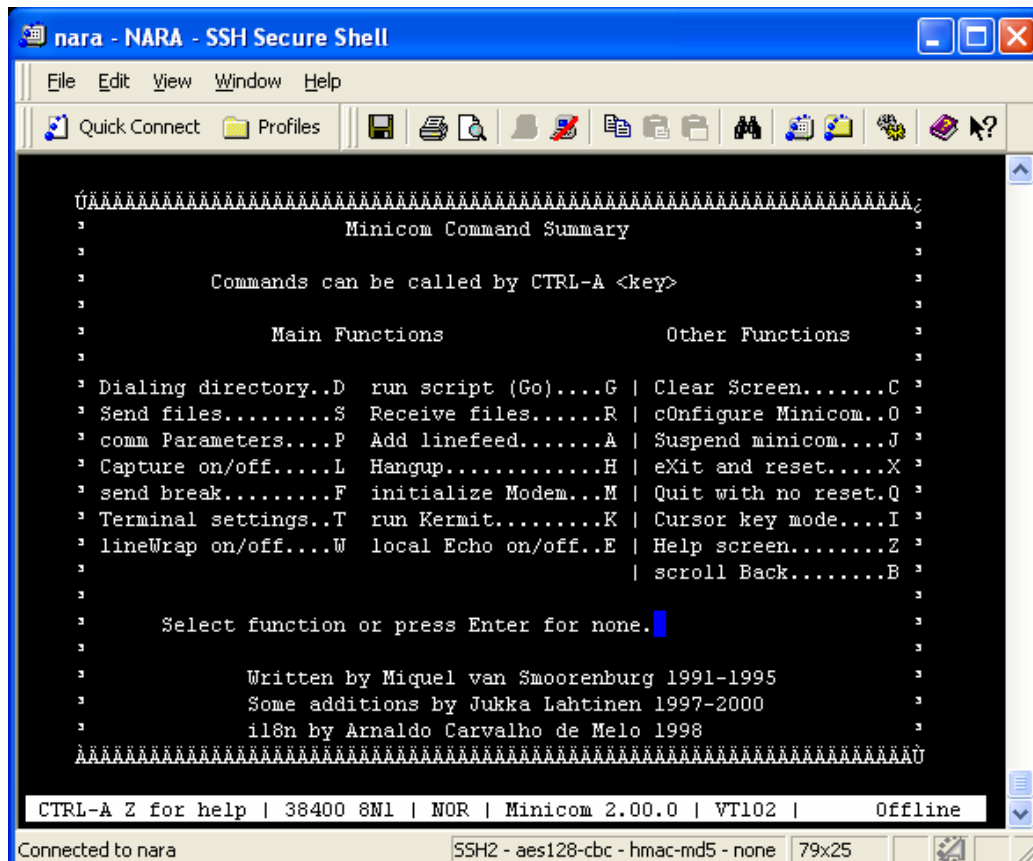


Figura 16: screenshot dell'help di minicom

A questo punto per connettersi all'AGDF1 bisogna impostare i settaggi della seriale su cui lavorerà **minicom** col tasto O. Apparirà il sottomenù "configuration" e da questo bisogna selezionare "Serial port setup". È sufficiente impostare la seriale (tasto A) e la sua velocità (tasto E), che nel caso di una sola AGDF1 connessa sono, rispettivamente, /dev/ttyS1 e 19200 8N1.

Per migliorare la visualizzazione dell'output dell'AGDF1 bisogna premere <CTRL> + A e poi W per consentire il ritorno a capo alla fine della riga. Per l'aggiornamento del firmware, dopo aver predisposto l'AGDF1 per la ricezione del file esadecimale, si preme <CTRL> + A e poi S (Send Files), dal sottomenù Upload si seleziona ASCII. Si aprirà una schermata con il contenuto delle varie directory. Bisognerà selezionare la directory dove è stato precedentemente messo il file firmware per l'AGDF1, selezionarlo con la barra spaziatrice e infine <INVIO>. Atteso qualche minuto per il trasferimento del file l'AGDF1 dovrebbe rispondere positivamente. In caso contrario bisogna ripetere la procedura.

Come si sarà notato, per spedire i comandi a **minicom** bisogna sempre premere prima <CTRL> + A.

Detailed system tree

1. Introduzione

Questo capitolo è una sorta di appendice, vi è riportato infatti tutto il file system creato e risiedente nel DOM. Per ogni directory c'è l'elenco dei file in essa contenuti e una piccola descrizione degli stessi.

/applicativo:

Readme	Readme file for programs info
agdfd	Script for starting/stopping programs
compressor.sh	Script for compressing data file
config.staz	Configuration file for master2bn created by setup program
delete.sh	Script for deleting older data file
i-setup -> setup	Link for interactive setup
master2bn	Acquisition program
master2bn-seed	Acquisition program with communication with a seedlink server
setup	Acquisition Setup program
splitter-1.3	Communication program
start.sh	Script for autostarting programs
tn1-config	Script for TN-1 configuration

/boot :

System.map-2.4.28TN-1	Symbol table for the kernel
boot.b	Boot sector used by lilo
map	Disk map used by lilo
vmlinuz-2.4.28TN-1	Linux kernel used by the system

/bin :

bash	Bourne Again shell
cat	Concatenates links files
chgrp	Changes the group ownership of a file
chmod	Changes the permissions of a file
chown	Changes the user ownership of a file

cp	Copies files from one place to another
cut	Prints selected parts of lines from files
dd	Copies and converts from one input to an output
df	Displays the mounted filesystem
dircolors	Commands to set ls-color environment variables
dmesg	Examine or control the kernel ring buffer
du	Displays amount of disk used by files
echo	Displays a line of text
free	Displays the amount of memory
gunzip	Decompresses files compressed by gzip
gzip	Compresses files
hostname	Sets or prints the name of the current host
ipcs	Provides information on the IPC facilities
ipcrm	removes IPC objects and associated data structures from the system
kill	Sends signals to processes
killall	Sends a signal to all processes
ln	Creates symbolic links
login	Establishes a new session with the system
ls	Lists directory contain
mkdir	Creates directory
mkfifo	Creates FIFO
mknod	Create device entry
more	Filter for paging
mount	Mounts a filesystem
mv	Move files from one place to another
nice	scheduling priority
ping	Tests IP connection
ps	Displays running tasks
rm	Removes files or directories
rmdir	Removes directories
setserial	set and/or report serial port configuration information
setterm	Sets terminal attributes
sh	Link to linux bash
stty	Changes and prints terminal lines settings

sed	Stream editor
su	Run a shell with substitute user and group ID
sync	Forces changed blocks to disk, update the SB
tar	Archiving program
tcsh	Remote shell
telnet	Remote login
touch	Update the access and modification times of FILE to the current time
umount	Unmounts a mounted filesystem
uname	Prints system information

/sbin :

agetty	Opens a tty port and prompt for login
badblocks	Search bad blocks on a device
clock	Displays date and clock
debugfs	Interactive file system debugger
depmod	Handles dependency descriptions for modules
dosfsck	Verifies and repair a MS-DOS file systems
dumpe2fs	Prints the super block and blocks group information for a filesystem present on a device
e2fsck	Checks and repairs a Linux filesystem
e2label	Display or change the filesystem label
fdisk	Disk partition table manipulator
fsck	Checks and repairs a specified filesystem
fsck.ext2	Checks and repairs a Linux filesystem
halt	Tells the kernel to halt the system
hdparm	Provides a command line interface to various hard disk ioctls
ifconfig	Configures the IP address of Ethernet card
init	Initial process, run /etc/inittab config
initlog	Logs messages and events to the system logger
insmod	Loads a module in the running kernel
isapnp	Programs a pnp device from a config file
kerneld	Automount needed modules
killall5	Sends signal to all processes except its own
ldconfig	Load the dynamic link libraries

lilo	Install the Linux boot loader
lsmod	Shows information about loaded modules
mkdosfs	Creates a MSDOS filesystem
mke2fs	Creates a Linux filesystem EXT2
mkfs	Creates a specified filesystem
mkswap	Sets up a Linux swap area on a device
modinfo	Displays information from specified module
modprobe	Automatically load the relevant module
mount	Link to /bin/mount
pnpdump	Search for pnp devices
rdev	Queries and sets root,swap,video,ram disk size
reboot	Reboots the computer. = shutdown .r now
rmod	Unloads module from the running kernel
route	Displays the network routing
rpc.portmap	SUN RPC portmapper
runlevel	Display the previous and the present runlevel
shutdown	Brings the system down in a secure way
ln	Symbolically links
swapoff	Disables a swap devices
swapon	Enables a swap device
telinit	Sends appropriate signals to init
tune2fs	Adjusts tunable filesystem parameters
umount	Link to /bin/umount
update	Flushes the filesystem buffers

/usr/bin :

chattr	Changes the file attributes
clear	Clears your screen
elvis	VI like text editor
find	Search for files matching certain criteria
getconf	Prints the value of a POSIX or X/Open path or system configuration variable
getent	Get entries from administrative database
grep	Searches expression patterns in files

iconv	Converts text from one encoding to another encoding
id	Prints real and effective UID and GID of a user
ldd	Print shared library dependencies
lddlibc4*	
loadkeys	Load keyboard translation tables
locale	Prints information about the current locale environment
localedef	Compile locale definition files
lsattr	Lists the file attributes on a second extended file system
passwd	Changes password for a user
rpcgen	Remote Procedure Call (RPC) protocol compiler
scp	Secure copy
sleep	Delays for a specified amount of time
smbmount	Mount Windows shared drive
smbpasswd	Change password for Samba user
smbstatus	Displays information on Samba connection
smbumount	Unmount mounted Windows shared drive
sprof	Read and display shared object profiling data
tput	Initializes or resets the terminal
tzselect	Select a time zone
uuidgen	Command-line utility to create a new UUID value
vi	Link on elvis text editor

/usr/sbin :

adduser	Prompt utility to add user on the system
chat	Automated conversational script with a modem
chroot	Changes to another root filesystem
dhcpcd	Web server daemon from Apache
groupadd	Creates a new group account
groupdel	Deletes all entries that refer to a group
groupmod	Modifies a group account
grpck	Verifies the integrity of the system group
grpconv	Convert normal group files to shadow group
grpunconv	Convert shadow group files to normal group
icmpinfo	Displays ICMP messages

iconvconfig	Create fastloading iconv module configuration file
in.xxx	Protocol xxx started by inetd
in.comsat	
in.fingerd	
in.identd	
in.ntalkd -> in.talkd	
in.pop3d	
in.rexecd	
in.rlogind	
in.rshd	
in.talkd	
in.telnetd	
in.telnetnoopd	
in.tftpd	
in.timed	
in.writed	
inetd	Starts protocols configured in /etc/inetd.conf
klogd	Intercepts and logs kernel messages
lpd	Line printer daemon
mklost+found	Create a lost+found directory
netresolv	Program needed by netwatch
netwatch	Displays the network traffic
nmbd	Server for netbios over IP protocol
pmap_dump	Dumps the RCP protocol mapping
pmap_set	Sets the RCP protocol mapping
pppd	Point to Point Protocol daemon
pppdump	Convert PPP record file to readable format
pppstats	Print PPP statistics
routed	Maintain current routing tables
routed-query	Tool used by program routed
routed-trace	Tool used by program routed
rcp.mountd	Daemon used by NFS protocol
rpc.nfsd	NFS server daemon
rpcinfo	Displays info on RPC protocols

smbd	Provides files & printers sharing to Windows client
snmpd	Daemon to respond to SNMP request packets
snmptrapd	Receive and log SNMP trap messages
syslogd	Provides a log-in used by many modern programs
tcpd	Daemon for TCP protocol
tcpdchk	TCP wrapper rules check
tcpdmatch	Filters TCP incoming request
useradd	Creates a new user on the system
userdel	Deletes all entries that refer to a user
usermod	Modifies a user account
utmpd	Daemon for utmp protocol
watchdog	Daemon that checks if your system is still working.
wu.ftp	File transfer protocol daemon

/etc :

DIR_COLORS	Color mapping for ls-color
HOSTNAME	Hostname of the system
MACHINE.SID	File containing the system ID
NETWORKING	File to choose if you need the network or not
csh.cshrc	Config for csh
csh.login	Default login for csh and tcsh
dhcpd.conf	dhcpd configuration file
exports	Share definition for NFS
fstab	Filesystem mounted at the startup
ftppass	Definitions for ftp access
ftpconversions	Definitions for access compressed files on ftp
ftpgroups	Groups not allowed to use ftp
ftpusers	Users not allowed to use ftp
group	System group definitions
host.conf	IP search order definition
hosts	IP names definitions
hosts.allow	IP allowed to use the inet services
hosts.deny	IP not allowed to use the inet services
hosts.equiv	IP names equivalent table

hosts.lpd	IP allowed to use the remote printer service
inetd.conf	Configuration file for inetd daemon
inittab	INIT process configuration file
isanel.conf	PNP configuration file for NE2000 card
issue	Welcome message
ld.so.cache	Configuration file for ldconfig
ld.so.conf	Path for libraries
lilo.conf	Configuration file for lilo and doc-lilo
localtime	Convert a time value to a broken-down local time
login.access	Login access control table
login.defs	Login configuration control definitions
minirc.dfl	Default configuration file for Minicom
modules.conf	Can be used to pass parameters to the modules
motd	Linux kernel version
mtab	Filesystem currently mounted
netgroup	Define group for network users
networks	define netname-to-address mappings
nsswitch.conf	System Databases and Name Service Switch configuration file
ntp.conf	Configuration file to synchronize local time
passwd	System users definition
profile	Configuration file executed for all users login
protocols	protocols available for TCP/IP
resolv.conf	DNS definitions
rpc	Current RPC mapping
securetty	Define on which terminal root are allowed
services	TCP and UDP ports mapping
shadow	System users encrypted password
shells	Possible unix shell
smb.conf	Configuration file for Samba
smbpasswd	Encrypted password of Samba users
smbusers	System users equivalence of Samba users
syslog.conf	Configuration file for syslogd daemon
termcap	Configuration file for some terminal
watchdog.conf	Configuration file for watchdog

/etc/ppp/

Contains script and configuration files for PPP.

rc.d/

rc.0	Executed when the system is halting
rc.4	Executed when the system start in graphical mode
rc.6	Executed when the system is rebooting
rc.K	Executed when the system goes in administrative mode
rc.M	Executed when the system goes in multi-users mode
rc.S	Executed when the system goes in single-user mode
rc.cdrom	Use to mount cdrom at the startup
rc.font.sample	Set the default screen font
rc.httpd	Start the apache web server
rc.ibcs2	Start the Intel binary compatibility emulation
rc.inet1	Configure network devices with IP address
rc.inet2	Start the network protocols
rc.keymap	Load the default keyboard map
rc.local	Executed at the end of boot for custom configuration
rc.modules	Load extra-modules to Linux kernel
rc.netdevice	Configure the networks devices
rc.pcmcia	Start the PCMCIA service
rc.samba	Start the Samba service
rc.serial	Initialize the system serial ports
rc.sysvinit	provides basic compatibility with SystemV

/etc/rc.d/init.d/

snmpd	snmpd agent management script
watchdog	watchdog management script

/etc/snmp/

snmpd.conf	snmpd configuration file
------------	--------------------------

/dev :

console	Main terminal of the system on normal nod
fd	Main terminal directly mapped from /proc
fd	Floppy A device
fla	DiskOnChip device
fla1 → fla4	DiskOnChip partitions
hda	IDE primary master device
hda1 → hda8	IDE primary master partitions
hdb	IDE primary slave device
hdb1 → hdb8	IDE primary slave partitions
hdc	IDE secondary master device
hdc1 → hdc4	IDE secondary master partitions
hdd	IDE secondary slave device
hdd1 → hdd4	IDE secondary slave partitions
initrd	Special ram device for initrd
kmem	Special ram device for kernel
loop0 → loop3	Loop devices for mounting files
lp0 → lp2	Printer port devices
null	Null device
ptmx	Device needed for telnet
ram0 → ram4	RAM disk devices
stderr	Standard error terminal
stdin	Standard input terminal
stdout	Standard output terminal
systty	Link to main console
tty0 → tty9	Auxiliary console terminal for multi-user mode
ttyS0 → ttyS6	Serial devices
urandom	Device used by the SSH protocol
watchdog	Watchdog device
zero	Device zero

/proc :

Contains the hardware mapping of the system detected by the Linux kernel at the startup.

Some text files contain the device detected on the PCI bus, the used interrupts, the used port

addresses, the memory mapping and much more.

/mnt :

Contains the reserved mounting point to mount some devices on the system like hard drive, cdrom, floppy, RAM disk, network link. /lib :

Contains all dynamic linked libraries used by the system.

/lib/modules/2.2.19/...:

Contains all modules usable by the system kernel.

/usr/lib :

Contains dynamic linked libraries used by Samba and SSH.

/usr/local... :

Contains all binary files and configuration files needed for the SSH protocol.

/usr/share/terminfo/...:

Contains configuration files for the VT100 and the Linux terminal.

/root :

Contains profile and configuration files for the user root.

/seedlink-ridotto/...

Contains all binary files and configuration files needed for the Seedlink server.

/home/...:

Contains profile and configuration files for the systems users.

/tmp :

Directory used to store temporary files.

/var :

Directory used to stored lock files, log files, messages and spool. /var/lib/apache :

Contains all binary files, configuration files and data files needed by the web server apache.

/lost+found :

Directory used to store the not recovered data after an .fsck. on a EXT2/3 filesystem.

Appendice: descrizione comandi linux usati.

VI

VI è l'editor "universale" di UNIX, per editare un file la sintassi è:

vi filename (dove filename è il nome del file ASCII che si vuole creare o modificare)

Segue ora una breve descrizione di alcuni comandi divisi per gruppi:

Comandi di Paging

- <CTRL/F> Avanzamento di una pagina
- <CTRL/B> Retrocessione di una pagina
- <CTRL/D> Avanzamento di metà schermo.
- <CTRL/U> Retrocessione di metà schermo

Comandi di posizionamento del cursore

Freccie muove il cursore nella direzione della freccia. In alternativa si usano i tasti h(sn) j(su) k(giù) l(dx).

- <INVIO-RETURN> Muove il cursore all'inizio della riga successive
- 0 Muove il cursore all'inizio della riga corrente
- \$ Muove il cursore alla fine della riga corrente
- <SPACE> Muove il cursore avanti di un carattere
- /pattern Sposta il cursore alla prima occorrenza in avanti di "pattern"
- ?pattern Sposta il cursore alla prima occorrenza indietro di "pattern"
- n Ripete l'ultima ricerca fatta con / o ? Repeats last / or ? pattern search.
- :n Moves to line number n

Comandi di inserimento testo

- i Modalità inserimento testo prima del cursore. Termina con <ESC>
- o Apre una nuova linea sotto la linea corrente per inserimento testo. Termina con <ESC>
- <ESC> Ferma l'inserimento di testo.

Comandi di cancellazione testo.

- dw Cancella la parola corrente
- x Cancella il carattere corrente
- dd Cancella la linea corrente
- P Rimette il testo precedentemente cancellato

Uscita da vi

:wq (equivalente :x) Scrive i cambiamenti e esce

:q Uscita senza aver fatto cambiamenti

:q! Forza l'uscita senza scrivere i cambiamenti

Per ulteriori informazioni sul programma si rimanda al manuale.

MOUNT

Tutti i file accessibili in un sistema Unix, sono disposti secondo un grande albero con radice /. Il comando **mount** serve per collegare i file system presenti nei vari dispositivi nell'albero dei file. Viceversa il comando **umount** stacca i file system.

La sintassi standard di **mount** è:

mount -t type device dir

Questo comando dice al sistema operativo (kernel) di attaccare il file system trovato sul dispositivo (di tipo *type*) sulla directory *dir*. Il precedente contenuto, proprietario di *dir* diventa invisibile e rimane tale fino a che questo file system rimane montato.

Sia **mount** sia **umount** mantengono una lista dei file system montati nel file */etc/mstab*. Se **mount** viene chiamato senza argomenti, viene visualizzata questa lista.

All'avvio del sistema il kernel monta tutti i file system elencati nel file di configurazione *fstab*. Normalmente il comando *mount* lo può eseguire solo il superuser, a meno di indicazioni in *fstab* (opzione **user**). Se un file system è presente in *fstab*, per montarlo è sufficiente indicare a *mount* solo il dispositivo o il punto di mount.

Ad es. la linea

/dev/cdrom /cd iso9660 ro,user,noauto,unhide

Permette a tutti il montaggio del file system di tipo ISO9660 trovato nel CDROM usando semplicemente il comando

mount /dev/cdrom o mount /cd

Solo l'utente che ha montato un file system lo può smontare, per permetterlo a tutti bisogna usare l'opzione *users* in *fstab*.

Per maggiori dettagli si rimanda al manuale di *fstab*.

Alcune opzioni disponibili per **mount** sono:

-V Scrive la versione del programma

-h Scrive l'help

-v Modo Verbose

-a Monta tutti i file system Mount scritti in *fstab*.

- n Monta senza aggiornare */etc/mstab*.
- r Monta il file system in modalità sola lettura (equivale a **-o ro**).
- w Monta il file system in lettura/scrittura (è il default).

Equivale a **-o rw**.

-t *vfstype*

L'argomento *vfstype* è usato per indicare il tipo di file system. Quelli attualmente supportati sono: *adfs, affs, autofs, coda, coherent, cramfs, devpts, efs,*

ext, ext2, ext3, hfs, hpfs, iso9660, jfs, minix, msdos, ncpfs, nfs, ntfs, proc, qnx4, ramfs, reiserfs, romfs, smbfs, sysv, tmpfs, udf, ufs, umsdos, vfat, xenix, xfs, xiafs.

-o Alcune opzioni richiedono il flag **-o**, separate dalla virgola:

async Tutti gli accessi al file system devono essere fatti in modo asincrono

atime (default) Aggiorna l'ora di accesso ad ogni accesso

auto come **-a**

defaults usa le opzioni di default: **rw, suid, dev, exec, auto, nouser,** and **async**.

dev Traduce caratteri o blocchi sul file system di dispositivi speciali

exec Permette l'esecuzione di file binari

nodev il contrario di **dev**

noexec Nega l'esecuzione di file binari. Questa opzione è utile quando nel file system sono presenti file per differenti architetture hardware dalla propria

nosuid Non permette ai bit set-user-identifier o set-group-identifier di aver effetto

nouser (default) ad utenti ordinary (non root) di montare file system

remount Tenta di rimontare un file system già montato. Comunemente è usata per modificare le caratteristiche di un file system, specialmente per farlo in sola scrittura. Non cambia il punto di mount

ro monta il file system read-only.

rw monta il file system read-write.

suid Contrario di **nosuid**

sync Tutti gli accessi al file system devono essere fatti in modo sincrono

dirsync L'aggiornamento di **directory** nel file system deve essere fatto in maniera sincrona.

Questa opzione condiziona il funzionamento delle chiamate di sistema: *creat, link, unlink, symlink, mkdir, rmdir, mknod* e *rename*.

user Consente ad un utente ordinario di montare i file system, secondo quanto specificato in *fstab*.

users consente a tutti gli utenti di montare e smontare file system.

Per ulteriori informazioni si rimanda al manuale.

INIT: inizializzazione dei processi

La sintassi di **init** è:

```
/sbin/init [ -a ] [ -s ] [ -b ] [ -z xxx ] [ 0123456Ss ]
```

Init è il genitore di tutti i processi. Il suo ruolo principale è quello di lanciare i processi di sistema secondo quanto scritto negli script elencati nel file **/etc/inittab**. In questo file, di solito, ci sono le chiamate al programma **getty** per permettere il login degli utenti.

RUNLEVELS

Un *runlevel* è una configurazione del sistema che consente ad un determinato gruppo di processi di girare. Questa associazione (processi – runlevel) è definita sempre nel file **/etc/inittab**. **Init** può essere in un runlevel compreso fra **0** e **6** oltre a **S** e **s**. I runlevels **0**, **1**, and **6** sono riservati: runlevel **0** è per l'halt del sistema, il **6** per il reboot e l'**1** per il system down in modalità single user.

Cambio RUNLEVELS

Dopo aver lanciato tutti i processi, **init** aspetta che un processo creato termini, un segnale di powerfail, o fino a che non deve cambiare il runlevel del sistema. Quando una di queste tre condizioni accadono, riesamina **/etc/inittab**, seguendone i comandi. Si possono fare modifiche a questo file in qualunque momento, ma la loro effettiva esecuzione non avviene eccetto che non accada una delle tre precedenti condizioni o si forzi **init** con il comando **init q**.

Al momento del cambio del runlevel **init** fornisce il segnale SIGTERM a tutti i processi non richiesti dal nuovo runlevel. Attende 5 secondi prima di terminare forzatamente i processi con il segnale SIGKILL.

IFCONFIG - configurazione di un'interfaccia di rete

SINTASSI

```
ifconfig [interfaccia]
```

```
ifconfig interfaccia [atype] opzioni | indirizzo ...
```

DESCRIZIONE

Ifconfig è usato per configurare le interfacce di rete residenti nel kernel. È usato al boot per configurare molte di queste ad uno stato funzionale (running state). Dopo di che, è necessario solamente quando si fa il debug o quando è necessaria una regolazione del sistema.

Se non è dato nessun argomento, **ifconfig** mostra solamente lo stato delle interfacce attualmente definite. Se è dato solo l'argomento interfaccia, mostra lo stato solo dell'interfaccia specificata.

Altrimenti suppone che ci sia qualcosa da configurare.

OPZIONI

interfaccia

Il nome dell'interfaccia di rete. Di solito è un nome come eth0, sl3 o qualcosa così: il nome di un device driver seguito da un numero di unità.

up

Questo flag causa l'attivazione dell'interfaccia. E' specificato implicitamente se viene dato un nuovo indirizzo all'interfaccia (si veda sotto).

down

Questo flag causa la disattivazione del driver per questa interfaccia, ed è utile quando le cose iniziano ad andare male.

`[-]arp` Abilita o disabilita l'uso del protocollo ARP su questa interfaccia. Se è presente il segno meno (-), il flag è disattivato.

`netmask addr`

Imposta l'IP network mask per questa interfaccia. Il valore di default è quello solito delle network mask di classe A, B e C (come dedotto dall'indirizzo IP dell'interfaccia), ma può essere impostato ad un valore qualsiasi per l'uso delle sottoreti.

`[-]broadcast [addr]`

Se è dato anche un indirizzo imposta l'indirizzo broadcast di protocollo (protocol broadcast address) per questa interfaccia. Altrimenti imposta solo il flag `IFF_BROADCAST` dell'interfaccia.

Se la parola chiave è preceduta da un segno meno (-), allora il flag è reimpostato.

`[-]pointpoint [addr]`

Questa parola chiave abilita la modalità point-to-point di un'interfaccia, intendendo che è un collegamento diretto tra due macchine con nessun'altro in ascolto. Se è dato anche un indirizzo, imposta l'indirizzo di protocollo (protocol address) dell'altro capo del collegamento, proprio

come fa l'obsoleta parola chiave `dstaddr`. Altrimenti imposta solamente il flag `IFF_POINTOPOINT` dell'interfaccia. Se la parola chiave è preceduta da un segno meno (-), allora il

flag è reimpostato.

hw

Imposta l'indirizzo hardware di questa interfaccia, se il device driver supporta questa operazione. La parola chiave deve essere seguita dal nome della classe hardware e l'equivalente stampabile in ASCII dell'indirizzo hardware. Le classi hardware attualmente supportate comprendono: ether (Ethernet), ax25 (AMPRAX.25), ARCnet e netrom (AMPR NET/ROM).

multicast

Imposta il flag multicast sull'interfaccia. Ciò di solito non dovrebbe essere necessario in quanto i driver impostano correttamente da soli il flag.

indirizzo

Il nome dell'host o l'indirizzo IP (un nome di host sarà risolto in un indirizzo IP) di quell'interfaccia. Questo parametro è richiesto, sebbene la sintassi attualmente non lo richieda.

INDICE

Sistema Transmission Node 1	2
1. Introduzione	2
2. Generalità	2
3. Descrizione hardware della scheda M543	4
4. Informazioni sul Kernel	5
5. Installazione del Disk On Module	5
6. L'architettura software	7
Installazione	16
1. Introduzione	16
2. Descrizione della scheda di alimentazione MASTER	16
3. Procedura di assemblaggio con la scheda di alimentazione	17
4. Procedura di assemblaggio della scheda adattatore compact flash eide	18
5. Bios Setup	19
6. Configurazioni della scheda	20
Test	26
1. Introduzione	26
2. Test delle connessioni di rete	26
3. Test della connessione seriale	30
4. Test della stazione	30
5. Aggiornamento del firmware dell'AGDF1	32
Detailed system tree	34
1. Introduzione	34
Appendice: descrizione comandi linux usati.	46