

Rapporti tecnici INGV

Software per l'installazione e
la configurazione della stazione
sismica GAIA2

130



Direttore

Enzo Boschi

Editorial Board

Raffaele Azzaro (CT)

Sara Barsotti (PI)

Mario Castellano (NA)

Viviana Castelli (BO)

Anna Grazia Chiodetti (AC)

Rosa Anna Corsaro (CT)

Luigi Cucci (RM1)

Mauro Di Vito (NA)

Marcello Liotta (PA)

Lucia Margheriti (CNT)

Simona Masina (BO)

Nicola Pagliuca (RM1)

Salvatore Stramondo (CNT)

Andrea Tertulliani - coordinatore (RM1)

Aldo Winkler (RM2)

Gaetano Zonno (MI)

Segreteria di Redazione

Francesca Di Stefano - coordinatore

Tel. +39 06 51860068

Fax +39 06 36915617

Rossella Celi

Tel. +39 06 51860055

Fax +39 06 36915617

redazionecen@ingv.it



Rapporti tecnici INGV

SOFTWARE PER L'INSTALLAZIONE E LA CONFIGURAZIONE DELLA STAZIONE SISMICA GAI A2

Sandro Rao, Leonardo Salvaterra, Catello Acerra

INGV (Istituto Nazionale di Geofisica e Vulcanologia, Centro Nazionale Terremoti)

130

Indice

Introduzione	5
1. RedBoot.....	5
1.1 Installazione.....	5
1.2 Configurazione	5
1.3 Comandi.....	6
2. PrgArm.....	8
2.1 Comandi.....	10
3. Setup di GAIA e GAIASetup	12
3.1 Setup AGDF2	13
3.2 Protocollo SNMP e Agent TN2.....	13
3.2.1 Versioni SNMP	14
3.2.2 SNMP e UDP	15
3.2.3 Agent TN2.....	15
3.3 GAIASetup	16
3.3.1 Parametri AGDF2	19
3.3.2 Parametri TN2.....	23
Conclusioni	25
4. Appendice A: Comandi Redboot.....	26
5. Appendice B: Terminale seriale per Linux (accesso di emergenza).....	28
6. Appendice C: Script PrgArm	29
7. Appendice D: Descrizione stringa di setup dell'AGDF2.....	35
8. Appendice E: Setup di stazione	46
9. Bibliografia	48

Introduzione

L'evoluzione della stazione sismica denominata GAIA prodotta nei laboratori dell'INGV (sede di Roma), ha coinvolto sia il modulo di acquisizione a 24 bit AGDF (Acquisitore GPS DAC Feedback) sia il modulo di connettività ed elaborazione TN (Transmission Node), dal punto di vista dell'hardware e del software.

La nuova versione del modulo AGDF prevede la possibilità di aggiornamento dei parametri di acquisizione via software per cui si è reso necessario sviluppare dei programmi ad hoc in linguaggio Visual Basic™, che consentissero il dialogo tramite seriale e TCP-IP.

La versione 2 del modulo TN (TN-2) è un sistema embedded basato su architettura ARM (il TN-1 era basato sul processore AMD ELAN SC520 compatibile con l'Intel 486DX4/DX5) [Pintore e Salvaterra, 2007]. Come per la prima versione, il sistema operativo è sempre Linux, ma si è dovuto operare un notevole sforzo per la conversione degli applicativi del TN-1 per la nuova architettura. Oltre a ciò il processo di avvio prevede un ambiente dedicato denominato RedBoot™ (acronimo di “Red Hat Embedded Debug and Bootstrap”), standard di Red Hat per il debug e avvio dei sistemi embedded, che ha un suo protocollo di installazione e configurazione. Per il dialogo con l'AGDF e la sua configurazione, inoltre, sono stati sviluppati, in linguaggio C, una libreria e un server (agent) basato sul protocollo SNMP (Simple Network Management Protocol).

Lo scopo di questa pubblicazione è quello di illustrare i software sviluppati e/o utilizzati per il progetto GAIA2: i primi due capitoli descrivono il processo di installazione del TN-2 (svolto nei laboratori INGV), mentre il terzo, rivolto all'utente finale, introduce il protocollo SNMP e analizza il software, GAIASetup, sviluppato per la programmazione della GAIA2.

In questo documento verranno seguite le seguenti convenzioni tipografiche:

- *corsivo* per i nomi di programmi
- **grassetto** per la specifica di linee di comando
- Font Arial per file di configurazione di programmi.

1. RedBoot

RedBoot™ è l'acronimo di “Red Hat Embedded Debug and Bootstrap”, ed è lo standard di Red Hat per il debug e avvio dei sistemi embedded, quale è appunto il TN-2. Comprende un ambiente di avvio completo per un ampio numero di sistemi operativi, come Linux e eCos. Per far ciò, attraverso una semplice interfaccia a comandi, è dotato di varie funzionalità tra le quali quella di eseguire degli script, la possibilità di modificare l'ora di sistema, di selezionare l'immagine del sistema operativo di avvio, d'impostare l'indirizzo IP ecc.

1.1 Installazione

In genere *RedBoot* viene caricato dal settore di boot della flash del sistema o dalla ROM di boot nella fase di accensione. Il metodo per installare *RedBoot* nella memoria non-volatile cambia a seconda dell'hardware, ma in generale prevede il collegamento via seriale o con un programmatore di memorie. Per quanto riguarda il TN-2, *RedBoot* è installato tramite seriale e memorizzato nel RAM flash.

L'interfaccia a linea di comando di *RedBoot* (Command Line Interface – CLI) è normalmente disponibile su una porta seriale del sistema e, se è presente anche un'interfaccia di rete, è accessibile usando il protocollo telnet alla porta TCP/9000 (comunque configurabile).

1.2 Configurazione

Per la configurazione di rete, oltre all'indirizzo statico, *RedBoot* supporta l'indirizzo IP dinamico, utilizzando il BOOTP (un protocollo sottoinsieme del DHCP). Il comando per configurare la rete è **fconfig**. La configurazione è salvata nella memoria flash e ogni modifica diventa effettiva al riavvio del sistema.

Esempio:

```
RedBoot> fconfig -l
```

Run script at boot: false
Boot script timeout (1000ms resolution): 2
Use BOOTP for network configuration: false
Gateway IP address: 0.0.0.0
Local IP address: 10.10.10.100
Local IP address mask: 255.0.0.0
Default server IP address: 10.100.10.14
DNS server IP address: 0.0.0.0
Set eth0 network hardware address [MAC]: false
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false

In questo caso il sistema è stato configurato con un IP statico (Local IP address), il default server IP address specifica il server TFTP a cui richiedere le immagini di boot, il “DNS server IP address” messo a 0.0.0.0 disabilita la possibilità di usufruire del servizio di lookup. Se “Use BOOTP for network configuration” fosse stato “true”, l’IP sarebbe stato caricato all’avvio col protocollo BOOTP. La “GDB connection port” è quella relativa alla connessione telnet.

1.3 Comandi

I comandi di RedBoot si possono classificare in tre gruppi:

- Caricamento ed esecuzione di programmi
- Gestione delle immagini nella memoria Flash e relativa configurazione
- Comandi vari

In appendice A è riportato l’elenco di tutti i comandi di *RedBoot*. In questo lavoro analizzeremo in dettaglio solo alcuni comandi, poiché utilizzati nella installazione del TN-2.

Il primo di questi è **load**, comando per caricare in memoria , via seriale o rete, programmi o dati:

load [-v] [-d] [-r] [-m [[xmodem | ymodem] | tftp | disk]] [-h server_IP_address] [-f location] [-b location] [-c channel] [file_name]

Le connessioni possono essere attraverso la rete, con i protocolli TFTP o http, o attraverso la seriale con il protocollo X/Y modem. Inoltre i file possono essere caricati direttamente dal file system locale o da disco. Per quanto riguarda il TN-2, questo comando viene utilizzato per il caricamento da remoto del kernel Linux e del file system da installare.

Il secondo comando che analizziamo è **fis**.

Molti sistemi, per tenere traccia di come è strutturato lo spazio in flash, contengono una “Flash Image System (FIS) directory”; *RedBoot* è in grado di gestirla, appunto, con il comando **fis** che si differenzia in:

fis init

che serve per inizializzare il FIS. Normalmente viene eseguito solo una volta all’installazione di *RedBoot*. È necessario rieseguirlo quando variano le immagini o la loro grandezza memorizzate in flash.

Esempio TN-2:

```
RedBoot> fis init
```

```
About to initialize [format] FLASH image system - continue (y/n)? y
```

```
*** Initialize FLASH Image System
```


Warning: device contents not erased, some blocks may not be usable

... Erase from 0x63fc0000-0x64000000: .

... Program from 0x03fbf000-0x03fff000 at 0x63fc0000: .

fis list

che visualizza il contenuto della directory FIS.

Esempio TN-2:

```
RedBoot> fis list
```

Name	FLASH addr	Mem addr	Length	Entry point
RedBoot	0x60000000	0x60000000	0x00040000	0x00000000
RedBoot config	0x63F80000	0x63F80000	0x00001000	0x00000000
FIS directory	0x63FC0000	0x63FC0000	0x00040000	0x00000000
root_tn2.gz	0x60040000	0x00800000	0x00B00000	0x00800000
zImage	0x60B40000	0x00080000	0x00200000	0x00080000

fis create **{-b data address}** **{-l length}** **[-f flash address]** **[-e entry]** **[-r relocation address]** **[-s data length]** **[-n | name]**

che si usa per creare un'immagine nella directory FIS, prendendo i dati precedentemente memorizzati in RAM. Tipicamente segue il comando precedente **load** che, come detto, è usato a tal scopo.

Esempio TN-2:

```
RedBoot> fis create -b 0x80000 -l 0x1DFF5C zImage
```

... Erase from 0x60b40000-0x60d40000:

... Program from 0x00080000-0x0025ff5c at 0x60b40000:

... Erase from 0x63fc0000-0x64000000: .

... Program from 0x03fbf000-0x03fff000 at 0x63fc0000: .

fis load **[-b load address]** **[-c | -d | name]**

che è usato per caricare in RAM un'immagine presente in memoria flash. Una volta caricata, l'immagine può essere eseguita con il comando **go** o **exec**.

fis erase **{-f flash_address}** **{-l length}**

usato per cancellare una porzione di memoria flash:

Esempio TN-2:

```
RedBoot> fis erase -f 0x60040000 -l 0x3F40000
```

... Erase from 0x60040000-0x63f80000:.....

Il terzo comando che analizziamo è:

exec **[-w timeout]** **[-r ramdisk_address]** **[-s ramdisk_length]** **[-b load_address {-l load_length}]** **[-c kernel_command_line]** **[entry_point]**

che serve ad eseguire applicazioni come un Linux kernel. Possono essere passati dei parametri addizionali al kernel per la sua esecuzione. Le opzioni “-r” e “-s” servono per passare al kernel, le informazioni sul ramdisk. L'opzione “-c” è usata per passare al kernel i parametri di avvio.

Esempio TN-2:

```
RedBoot> exec -r 0x800000 -s 0xB00000 -c "root=/dev/ram ramdisk=34000 console=ttyAM0"
```

2. PrgArm

Per l'installazione del sistema operativo e del relativo file system nel TN-2 attraverso *RedBoot*, è necessario quindi dotare la stessa scheda (nella fig. 1 è visualizzato il box della GAIA2) di un collegamento seriale e ethernet per caricare in memoria i files per l'esecuzione di Linux. Inoltre, per automatizzare tutto il processo, viene utilizzato un personal computer con sistema operativo Windows XP™, dove gira il programma *PrgArm*, sviluppato dalla società E-Logic, in cui si trovano le funzionalità necessarie all'installazione quali terminale seriale, terminale telnet ed esecuzione di script. Durante l'installazione vengono scaricati nel TN-2 le immagini del sistema operativo tramite protocollo TFTP, per cui si rende necessario un computer con un tale server. Attualmente questo lavoro lo svolge un altro computer con sistema operativo Linux e con il quale sono state anche create le immagini del file system del TN-2. Naturalmente la stazione e i due computer devono appartenere alla stessa rete ethernet.

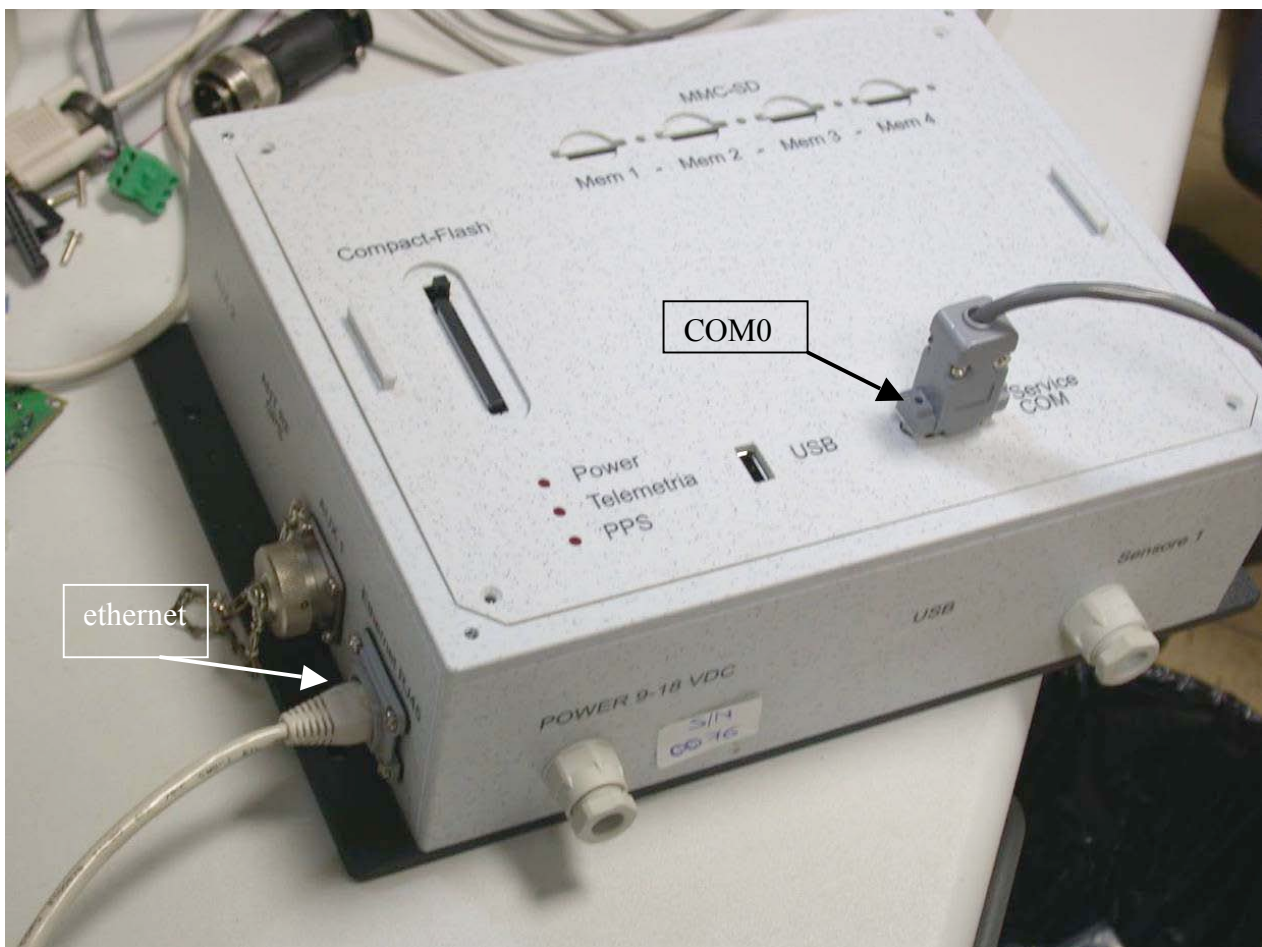
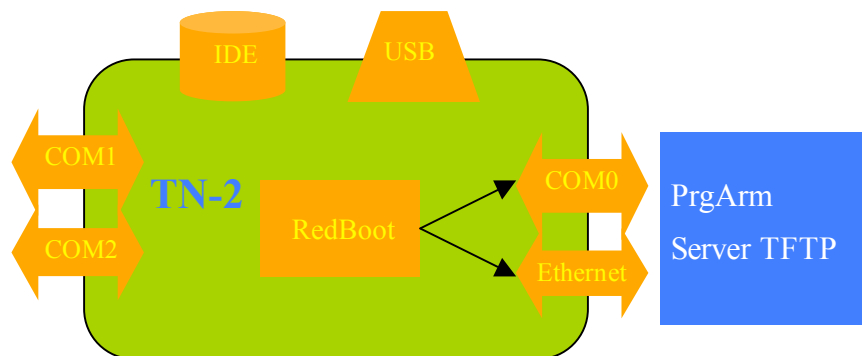


Figura 1. Connessioni per la programmazione del TN-2.

Nella fig. 2 è riportata la semplice finestra del programma *PrgArm* con i messaggi che *Redboot* fornisce all'avvio. La connessione è di tipo telnet (indicazione TCP nella finestrella) all'indirizzo **10.10.10.101** come indicato nella fig. 3. Quest'ultima finestra dei settaggi è accessibile premendo il pulsante "Set".

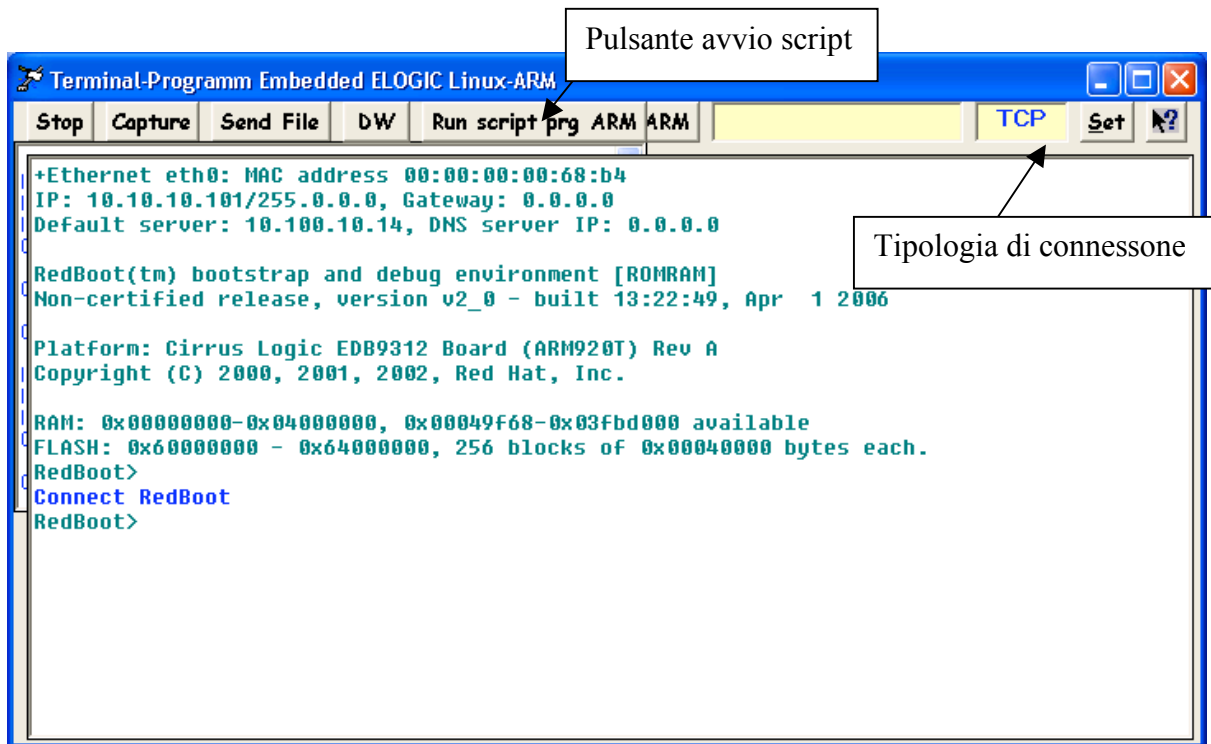


Figura 2. Screenshot di *PrgArm*.

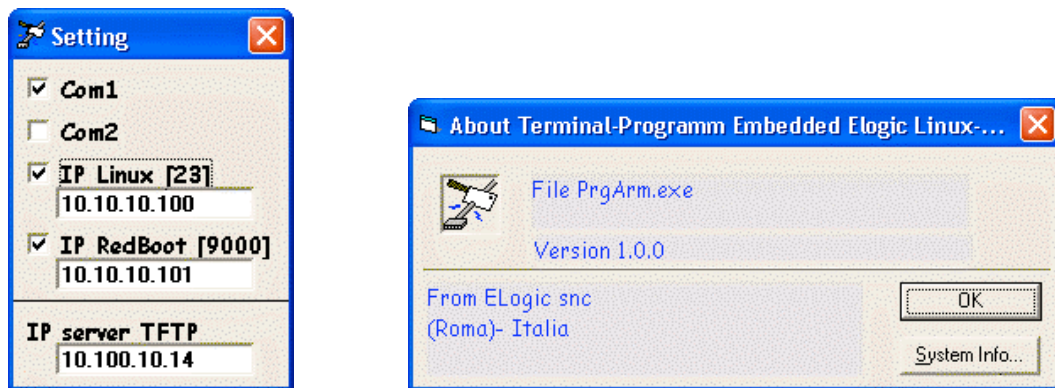


Figura 3. Settings e info sul programma *PrgArm*.

Vi sono una serie di pulsanti tra cui i principali sono: "Run script prg Arm" per avviare l'esecuzione di script, "Stop" per fermare l'esecuzione di uno script e il pulsante "DW" per aggiornare RedBoot.

L'esecuzione di script è, chiaramente, la funzionalità più importante di *PrgArm*, per cui abbiamo scritto uno script atto alla gestione di tutta l'installazione. In sequenza, le operazioni che *PrgArm* deve eseguire sono la formattazione della memoria flash del TN-2, la configurazione dei parametri di rete di *Redboot*, lo scaricamento delle due immagini (kernel e file system Linux) e loro installazione nella memoria RAM flash, la modifica dei settaggi di avvio di *Redboot* per l'esecuzione di Linux. Nell'appendice B è riportato il listato completo dello script di programmazione per il TN-2.

Per far partire lo script si preme sul pulsante di avvio script e dalla finestra di selezione file (fig. 4) si seleziona “programmaTN2V2.txt”.

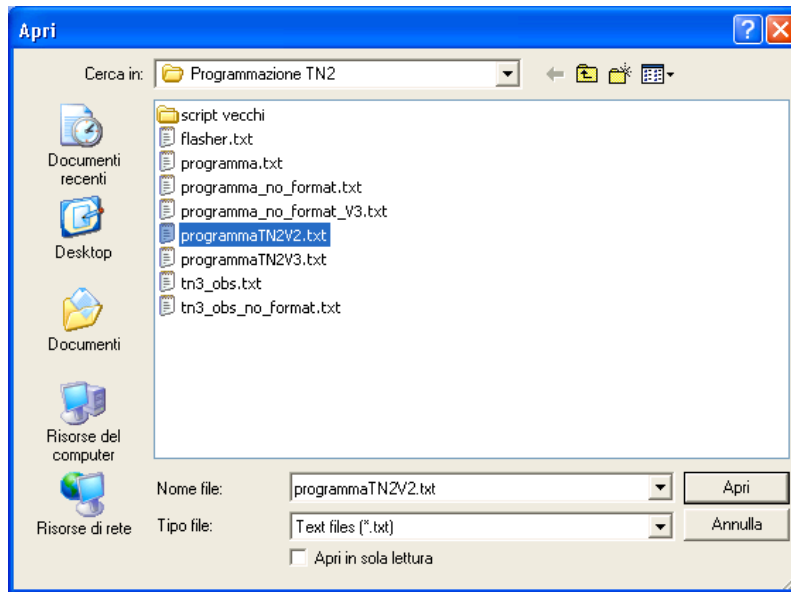


Figura 4. Finestra di dialogo per la selezione di script.

Dopo aver selezionato lo script, *RedBoot* riceverà, da *PrgArm*, i comandi da eseguire che verranno visualizzati a schermo.

2.1 Comandi

I comandi messi a disposizione da *PrgArm* per la creazione degli script sono cinque:

- **send stringa**

utilizzato per spedire a *RedBoot* i comandi. Per spedire il solo comando “invio” basta mettere stringa vuota, mentre \b equivale al backspace.

Esempio:

```
send fconfig -i
```

- **expect {**
 stringa
 timeout
}

comando per il riconoscimento e controllo dell’output di *RedBoot* per la corretta gestione del flusso dello script. In caso di non corrispondenza, dopo un tempo di timeout, viene terminata l’esecuzione dello script.

Esempio:

```
expect{  
continue (y/n)?  
timeout 2 goto expect_timeout  
}
```

- **print ‘stringa’**

utilizzato per visualizzare nella finestra di *RedBoot* informazioni sull’esecuzione dello script

- **exit**

per terminare l'esecuzione dello script.

- **#commento**

per inserire dei commenti nello script.

Infine, per l'aggiornamento di *RedBoot*, bisogna operare, ad alimentazione spenta, sul modulo TN-2 cambiando la posizione di un ponticello (cerchietto in fig. 5), mettendo in corto i piedini 1 e 2 (nel funzionamento normale il corto è fra i piedini 2 e 3).

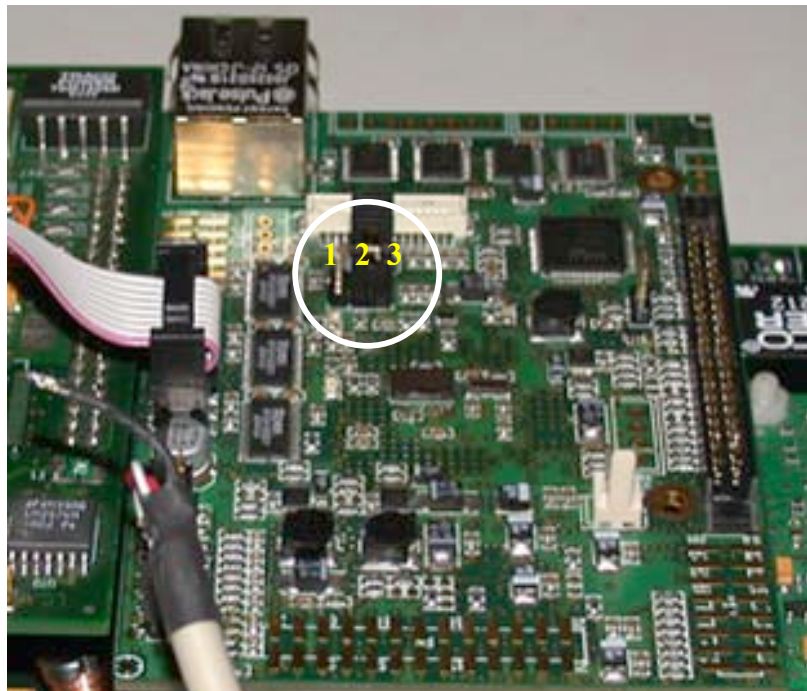


Figura 5. Modulo TN-2 con il ponticello di programmazione evidenziato.

Quindi si preme il pulsante di *PrgArm* "DW", si accende l'alimentazione e si attende che il processo abbia termine (fig. 6). Alla fine si rimette il ponticello nella posizione originale 2-3.

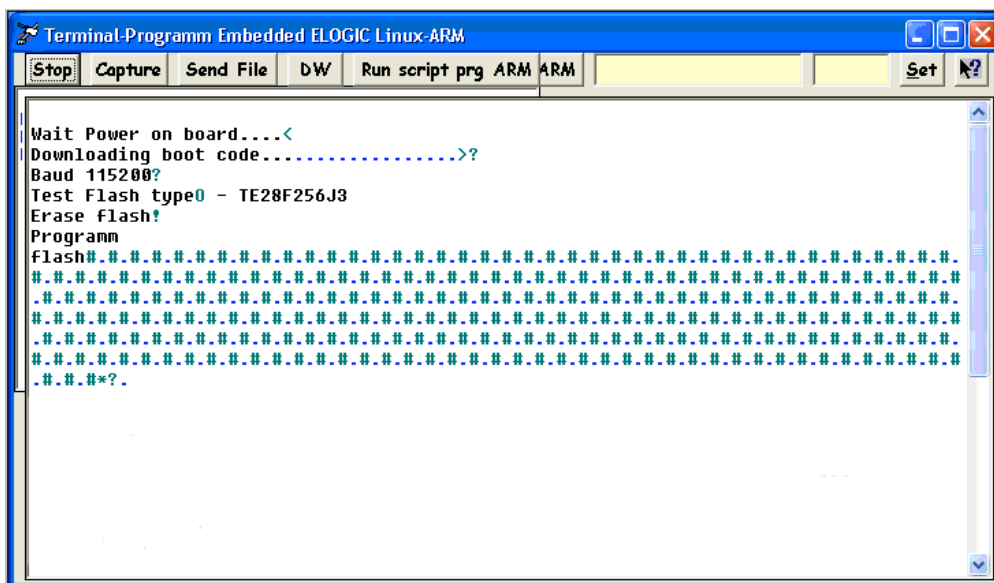


Figura 6. Screenshot di *PrgArm* con l'aggiornamento di RedBoot.

3. Setup di GAIA e GAIASetup

Terminata l'esecuzione dello script di *PrgArm*, il controllo del processo di installazione automatica passa a Linux, infatti, dopo il processo descritto nel paragrafo precedente, nella flash del TN-2 è installato il kernel 2.6.20 e un'immagine minimale ramdisk di un file system Linux. Al primo avvio, lo script di sistema *rc.sysint* esegue una serie di operazioni atte a terminare l'installazione: controllo della presenza della compact flash, di cui è dotata la GAIA2, e sua formattazione in due partizioni (delle quali una di swap); creazione delle directory per i log del sistema e il salvataggio dei dati sismici; prelevamento dal server TFTP della restante parte del file system linux (immagine di tipo "jffs2") dove risiedono tutti gli applicativi e successiva installazione nella memoria flash. Finite queste operazioni il sistema si riavvia, pronto per essere configurato.

Per proseguire bisogna illustrare come i due moduli, TN-2 e AGDF2, di cui è composta la GAIA2, comunicano. In breve la comunicazione, tra questi due moduli avviene attraverso una seriale TTL con velocità programmabile e si possono avere due stati di funzionamento: acquisizione e setup (fig. 7).

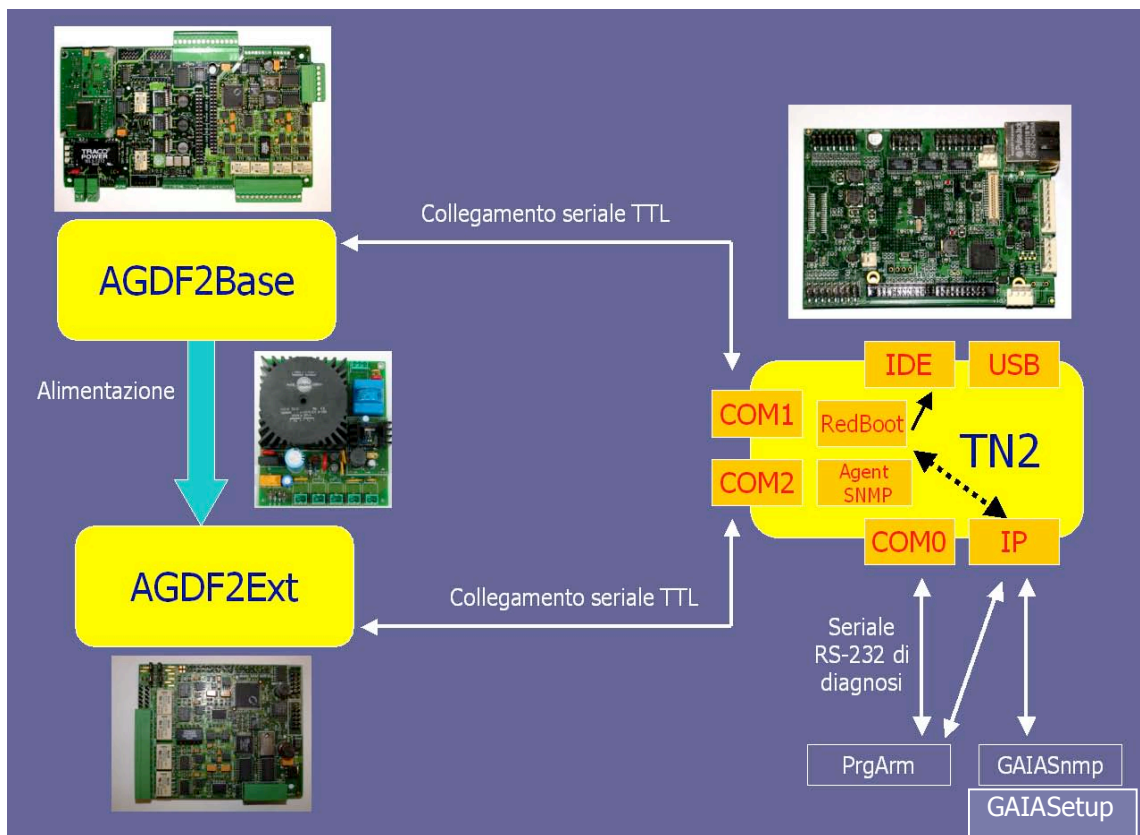


Figura 7. Schema a blocchi dei collegamenti di GAIA2.

Nel primo stato l'AGDF2 spedisce, in modo asincrono sulla seriale, i dati digitali a 24 bit dei segnali sismici, nel secondo stato spedisce una stringa con i parametri di setup pronta a ricevere un eventuale aggiornamento. Infatti, come vedremo nel prossimo paragrafo, per configurare l'AGDF2 non sono necessarie operazioni di tipo hardware (come invece accadeva per la prima versione dove bisognava saldare o dissaldare resistori per modificare il fondo scala), ma solo software attraverso l'invio di una stringa di setup con la seriale e questo compito è demandato appunto al TN-2. Per assolvere ciò, è stata sviluppata in linguaggio C una libreria che consente di creare la stringa di setup in base ai parametri modificati e di eseguire le operazioni previste dal protocollo di aggiornamento dell'AGDF2. Infine per impartire i comandi al TN-2 si è deciso di adottare il protocollo SNMP (vedi par. 3.2), per cui è stato necessario sviluppare un software che, utilizzando appunto il protocollo SNMP, fornisca all'utente un'interfaccia semplice per l'aggiornamento di tutti i parametri della GAIA2: GAIASetup.

3.1 Setup AGDF2

Il modulo AGDF2 nel funzionamento normale spedisce, via seriale, i valori campionati dei segnali sismici al suo ingresso. Durante questa attività, è possibile mandare alcuni comandi per ottenere la configurazione dell'AGDF2. Attraverso la seriale, inviando il carattere ASCII "F", la scheda ferma l'acquisizione ed entra nello stato di setup restituendo, una volta al secondo, una stringa di 1035 byte con i valori di tutti i parametri.

Il formato della stringa è il seguente:

```
SSETUP00DD01DD02DD03DD ----- FCDDFDDDFEDDFDDEOBCC
```

In cui SSETUP e EOB sono caratteri ASCII (9 byte), CC sono i due byte di checksum (C è un carattere alfanumerico ASCII), e i valori dei parametri sono rappresentati con i due byte DD (D è un carattere alfanumerico ASCII).

Il checksum viene costruito utilizzando (somma senza riporto) solo le coppie locazione – dato escludendo dal calcolo i caratteri ASCII SSETUP, EOB e CC.

Durante la trasmissione del setup è possibile inviare alla stazione AGDF2 un nuovo setup oppure è possibile tornare in modo acquisizione inviando il carattere ASCII "F". Il nuovo setup può essere sia completo (dalla locazione 00 alla locazione FF), sia parziale, cioè contenente le sole locazioni da aggiornare, rispettando sempre il formato stringa RSETUP descritto sotto. Appena terminato l'invio, la stazione AGDF2 rimanendo in modo setup, restituisce la stringa SSETUP con i campi modificati. La stringa per aggiornare il setup ha un formato simile al precedente:

```
RSETUP00DD01DD02DD03DD ----- FCDDFDDDFEDDFDDEOBCC
```

Il significato dei vari caratteri è lo stesso della stringa SSETUP. Per una descrizione dettagliata dei vari campi si rimanda all'Appendice D.

Errori

Durante l'invio del nuovo setup, potrebbero verificarsi degli errori, come ad esempio un trasferimento dati non corretto, la presenza di valori di parametri non ammessi, ecc. La locazione FF del setup è quella riservata al codice errore. In caso di errore, la stazione GAIA2 non aggiornerà la memoria permanente riservata al setup e trasmetterà ripetutamente il setup completo, non aggiornato tranne la locazione FF che conterrà il codice errore verificatosi (vedi appendice D). Attualmente l'AGDF2 opererà la procedura anzidetta solo in due casi: errore di trasferimento dati ed errore nel parametro di velocità seriale non ammessa..

Tutti gli altri codici errore sono relativi all'applicativo di acquisizione, pertanto saranno presenti dopo che è avvenuto l'avvio dell'acquisizione, in tal caso l'AGDF2 terminerà il modo acquisizione ed entrerà in modo setup aggiornando la locazione FF con il codice errore verificatosi.

3.2 Protocollo SNMP e Agent TN2

La stazione sismica GAIA2, come per la precedente versione, è utilizzata in reti di stazioni (dalla Rete Sismica Nazionale al piccolo array per esperimenti) tutte collegate in TCP/IP. Per migliorare la gestione di tali reti si è pensato, come detto, di ricorrere al protocollo internazionale *Simple Network Management Protocol* (SNMP). Introdotto nel 1988 per soddisfare la crescente esigenza di uno standard per la gestione di dispositivi IP (Internet Protocol), SNMP fornisce ai suoi utenti un semplice insieme di operazioni idonee allo scopo. Ad esempio è possibile spegnere da remoto l'interfaccia Ethernet di un router oppure tenere traccia della temperatura di uno switch e avvisare l'utente nel caso fosse troppo alta. L'uso dell'SNMP non si limita agli apparati di rete, ma può gestire sistemi Unix (come è appunto la GAIA2), Windows, stampanti, alimentatori e tutto ciò che esegue un software di gestione SNMP. Per capire come questo sia possibile, bisogna introdurre il concetto di manager e agent.

Un manager è un server in cui gira un software (detto anche Network Management Stations, NMS), in grado di monitorare tutte le risorse di rete, che è responsabile dell'interrogazione e della ricezione di allarmi dai vari agent risiedenti negli apparati di rete. In base alle informazioni ricevute o agli allarmi è in grado di compiere determinate azioni per avvisare l'utente di eventuali problemi. L'agent, è un software che risiede nella risorsa di rete da controllare e gestire, sia come programma a se stante (demone in sistemi Unix) sia incorporato nel sistema operativo (come ad esempio nei router Cisco). L'agent fornisce all'NMS tutte le

informazioni sul funzionamento del dispositivo (per esempio lo stato delle interfacce di un router) e mandare degli allarmi in caso di problemi (fig. 8). L'insieme di tutti i parametri (detti anche oggetti) controllati dall'agent costituisce un'astrazione di database detta Management Information Base (MIB).

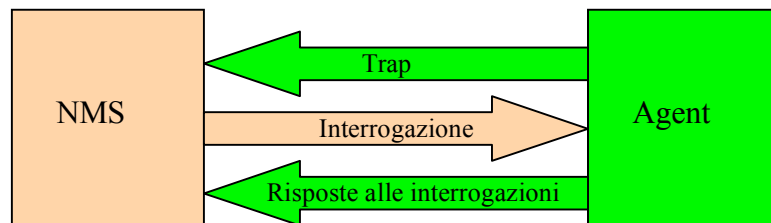


Figura 8. Rapporto tra Agent e NMS (manager).

La *Structure of Management Information* (SMI) ha il compito di definire, in modo standard, come devono essere descritti i vari oggetti all'interno di un MIB, per cui ogni apparato di rete avrà un suo MIB strutturato secondo le regole sintattiche dello SMI.

Visivamente, gli oggetti da gestire vengono organizzati secondo una struttura ad albero, in cui dal nodo radice (root) partono vari rami con i relativi nodi che rappresentano i parametri/oggetti da gestire. A questi ultimi è associato un identificativo composto da una serie di numeri separati da punti, ognuno dei quali rappresenta un nodo dell'albero. Le serie di numeri possono essere sostituite da nomi per avere un formato più leggibile. Ad esempio, nella fig. 9, l'oggetto *private* ha un OID (object identifier) 1.3.6.1.4 o iso.org.dod.internet.private.

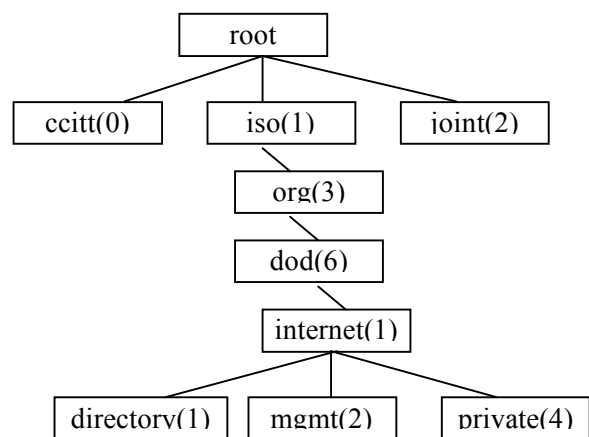


Figura 9. Esempio di MIB secondo la struttura ad albero.

Esiste una organizzazione, la Internet Assigned Numbers Authority (IANA), che gestisce e regola le concessioni di spazi all'interno dell'albero, sotto il nodo private, per privati, istituzioni, organizzazioni e compagnie. All'INGV è stato dato uno spazio privato sotto il ramo iso.org.dod.internet.private.enterprise.ingv o 1.3.6.1.4.1.21949.

3.2.1 Versioni SNMP

L'evoluzione del protocollo SNMP ha portato alla versione 3 e ha riguardato soprattutto la protezione nella comunicazione. La prima versione ha introdotto il concetto di community che definisce sia il metodo di autenticazione sia la politica di controllo degli accessi (read-only, read-write e trap). Ad ogni community viene assegnato un nominativo unico (community name) che sarà noto a tutte le stazioni (manager e agent) coinvolte in quella particolare relazione. Il community name svolge, quindi, il ruolo di parola chiave nella comunicazione tra le varie workstations, garantendo una limitata autenticazione nella richiesta di informazioni.

Allo scopo di estendere queste semplici funzionalità, nel 1993 è stata introdotta la versione successiva del protocollo SNMP. La SNMPv2 ha semplificato il trasferimento di una grande quantità di valori, ha introdotto nuovi comandi e cambiato il formato delle trap. La terza versione SNMP fu lanciata a giugno del 1998 e introdusse possibilità aggiuntive nel campo della sicurezza e dell'amministrazione. Le nuove opzioni del SNMPv3 si riferiscono particolarmente alle funzionalità di sicurezza per login e privacy e il controllo dell'accesso.

3.2.2 SNMP e UDP

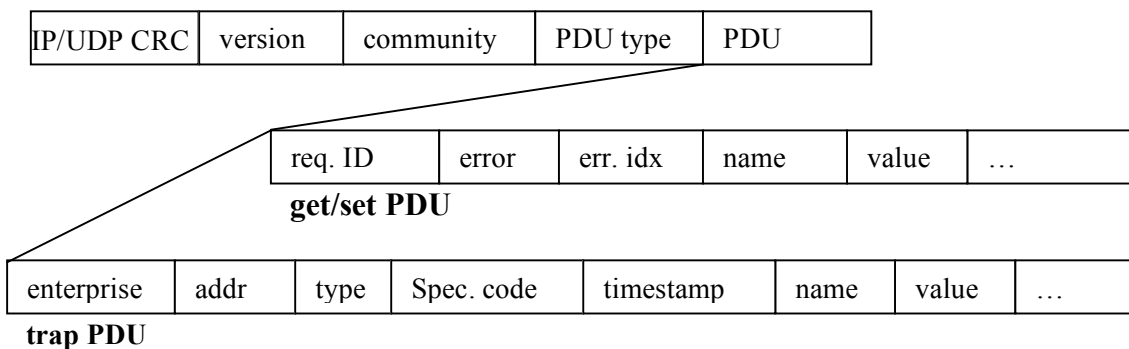
SNMP utilizza l'User Datagram Protocol (UDP), alle porte 161 per invio e ricezione richieste e 162 per le trap, come protocollo di trasporto per il passaggio dei dati tra i manager e gli agent. L'UDP non è orientato alla connessione come il TCP/IP, per cui l'eventuale ritrasmissione dei pacchetti persi è demandata ai manager e agli agent. Nella maggior parte dei casi l'inaffidabilità dell'UDP non è un problema, anzi si ha il vantaggio di minor carico della rete per la spedizione dei vari messaggi.

I messaggi scambiati tra manager e agent sono strutturati secondo il formato standard chiamato *Protocol Data Unit* (PDU).

Ogni operazione SNMPv1, di seguito elencata, ha un proprio PDU:

- *get* e *get-next* (richieste di informazioni da parte di un manager su uno o più elementi di un MIB)
- *set* (assegnazione, da parte di un manager, di valori a un elemento di un MIB)
- *get-response* (risposta di un agent ad una *get* o una *get-next*)
- *trap* (allarme asincrono inviato da un agent)

la struttura di ogni PDU è chiarita nella seguente figura



dove:

version è la versione del protocollo (0 per SNMPv1, 1 per SNMPv2 ecc.);

community è la stringa col nome della community;

PDU type indica il tipo di operazione richiesta (0=get, 1=get-next ecc.);

Request ID è l'identificativo della richiesta;

Error e **Err. Idx** sono il codice di errore eventualmente verificatosi (noError(0), tooBig(1), noSuchName(2), badValue(3), readOnly) e il puntatore alla variabile che ha causato l'errore;

enterprise è l'identificativo del costruttore;

agent addr. è l'indirizzo IP dell'agent;

trap type è l'identificativo dell'evento che ha causato la trap;

spec. code è un ulteriore codice stabilito dal costruttore;

timestamp intervallo di tempo trascorsi dall'ultima inizializzazione.

3.2.3 Agent TN2

Come detto in precedenza, per gestire la stazione GAI2 (aggiornamento del setup del modulo AGDF2 attraverso il TN-2, configurazione TN-2 ecc.), sono stati scritti in linguaggio C una libreria, un agent (basato sul pacchetto NET-SNMP) e un MIB (non trattati in questa pubblicazione) per utilizzare il protocollo SNMP. I vari parametri sono stati divisi in quattro gruppi: setup e status della AGDF2, setup e status del TN-2. I parametri di setup sono di tipo read-write, mentre i secondi sono di tipo read-only. I parametri del TN-2 vengono settati in un'unica operazione, mentre per quelli relativi all'AGDF2, dopo averli settati, bisogna anche operare una sincronizzazione tra TN-2 e AGDF2: viene in sostanza avviato il procedimento di aggiornamento descritto nel capitolo 3.1 "setup AGDF2". Allo stato attuale non sono state ancora implementate trap. La community name scelta è **tco-comm**.

3.3 GAIASetup

Il protocollo SNMP implementato nel TN-2 è il protagonista principale nella fase di configurazione della stazione sismica, ma, per essere sfruttato al massimo delle sue potenzialità e fornire all'utente una semplice interfaccia grafica si è reso necessario lo sviluppo di *GAIASetup*. Il software è stato progettato in ambiente Visual Basic™ e permette di variare sia la configurazione del TN-2 sia quella dell'AGDF2. Dopo aver lanciato il programma, appare la schermata di selezione del livello di setup e del protocollo da utilizzare (fig. 10).

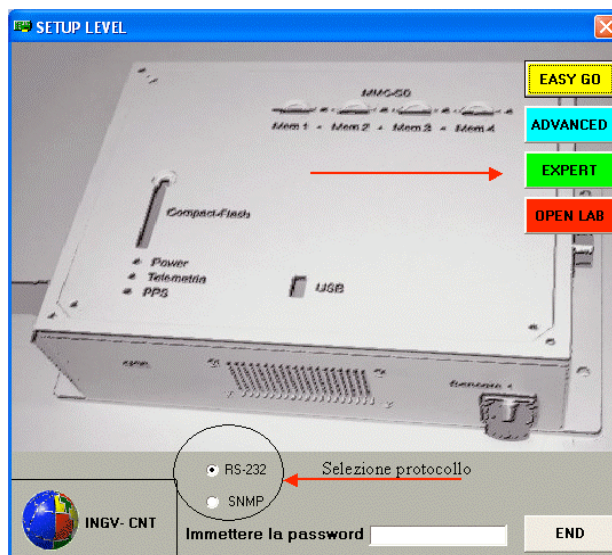


Figura 10. Screenshot di GAIASetup.

Progettato in origine solo per la comunicazione seriale, GAIASetup, mantiene la scelta del livello di setup, ma scegliendo il protocollo SNMP il livello più indicato e che analizzeremo è l'EXPERT (in fig. 11 si vede la schermata di setup vera e propria). I livelli EASY GO ed ADVANCED contengono un minor numero di campi, mentre OPEN LAB (protetto da password) consente l'accesso, solo tramite seriale, a dati riservati per interventi tecnici sull'AGDF2.

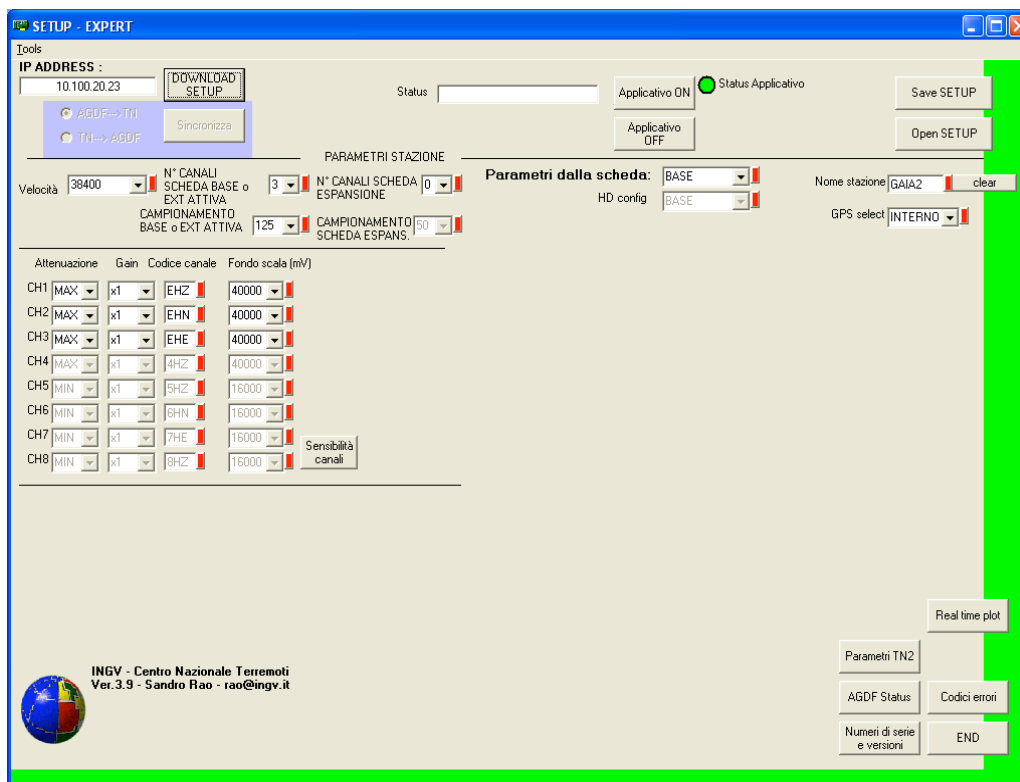


Figura 11. Schermata di setup.

Prima di analizzare l'interfaccia utente in tutte le sue funzioni, è opportuno fornire un diagramma di flusso (fig. 12) per poter meglio comprendere il funzionamento del software, tra l'altro direttamente collegato con la struttura hardware in questione.

Per prima cosa si deve immettere l'indirizzo IP della GAIA2 e far partire l'interrogazione del relativo agent SNMP (nell'appendice B è indicato il modo di recuperare l'IP se è ignoto). *GAIASetup* verifica la correttezza dell'IP immesso e dopo interroga l'agent richiedendo tutti i dati del MIB della GAIA2. Se i dati ricevuti sono corretti vengono riempiti tutti i campi dell'interfaccia, dopo di che è possibile la modifica dei vari valori a seconda delle varie esigenze di installazione. Ad ogni modifica, viene controllata la correttezza dei valori immessi. A questo punto il diagramma di flusso si divide in due rami rendendo nota l'esistenza di due procedure diverse a seconda che le modifiche riguardino il TN-2 o l'AGDF2: nel primo caso, si termina con un semplice "reboot" del TN-2 per consentirgli la lettura dei nuovi file di configurazione, mentre nel secondo, è necessaria una sincronizzazione, secondo quanto detto nel paragrafo 3.1, tra le due schede attraverso la porta seriale.

Il meccanismo di scambio dati tramite il protocollo SNMP, prevede l'utilizzo di quattro operazioni nominate *get*, *set*, *getnext* e *get-response*. L'operazione di scrittura o lettura viene effettuata attraverso una funzione scritta in linguaggio Visual Basic chiamata *GenPacket* la quale è stata strutturata per soddisfare qualsiasi tipo di richiesta all'agent SNMP, unitamente al componente Microsoft Winsock Control settato con il protocollo UDP (*sckUDPProtocol*). Quest'ultimo rende possibile la connessione verso macchine remote e lo scambio dei dati attraverso il protocollo TCP o UDP. Si possono creare applicazioni sia server sia client.

Delle quattro operazioni menzionate sopra, in realtà vengono impiegate solo le prime due e l'ultima, dal momento che la *getnext* si può ottenere dall'iterazione di una *get*, opportunamente inserita in un ciclo *for...next* controllato da un contatore di PDU ricevute.

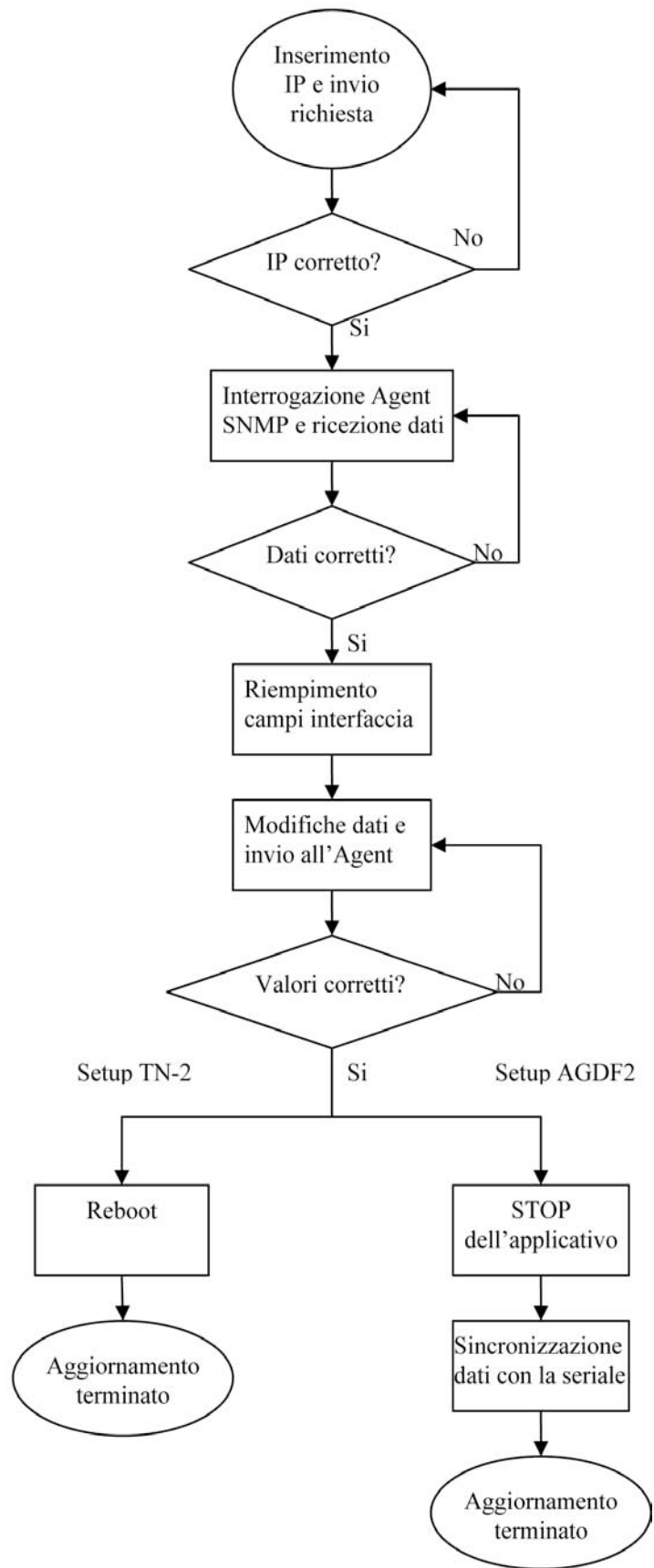


Figura 12. Diagramma di flusso di GAIA setup.

Di seguito è riportata la funzione GenPacket con i relativi argomenti e loro dichiarazioni di tipo:

Function GenPacket (SnmCommunity As String, SnmpOID As String, SnmpPacketID As Double, SnmpReqType As Double, argomento As Variant, tipo_s As Integer, lunghezza_s As Integer) As String

Dove:

SnmCommunity è la community scelta (tco-comm);

SnmOID è l'OID del dato richiesto (es. 1.3.6.1.4.1.21949.2.3.1.1.4.6);

SnmPacketID è il numero di pacchetti richiesti (nel nostro caso è sempre 1, perché si fa sempre la richiesta di un singolo pacchetto);

SnmReqType è il codice associato all'operazione richiesta (es. 160d, che corrisponde ad A0 hex, è il codice della procedura *get*);

argomento, nel caso di procedura *set*, è il valore che si vuole passare all'agent, nel caso di una *get-response* è il valore che l'agent ci restituisce, mentre nel caso di una *get* tale campo rimane vuoto;

tipo_s è il tipo di dato dell'argomento (può essere un integer, ip-address, string ecc);

lunghezza è la lunghezza in bit dell'argomento.

Per la richiesta dei dati all'agent la funzione GenPacket si configura per una procedura di *get* e, attraverso il componente WinSock, inizia a scaricare i parametri del TN-2. Essendo detta funzione, come accennato prima, inclusa in un ciclo controllato da un contatore che si incrementa ogni volta che viene scaricato un nuovo dato, l'operazione *get* in realtà viene così trasformata in una *getnext*. Tale routine è stata concepita per evitare dei blocchi riscontrati nel momento che veniva lanciata la procedura *getnext*. Alla fine della fase di download tutti i campi della finestra, sono popolati.

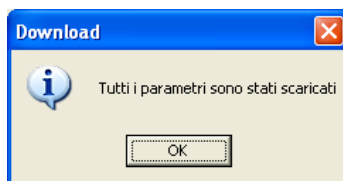
A questo punto l'utente può modificare i campi che necessitano di variazioni. Per rendere effettive le modifiche la funzione GenPacket, con il controllo WinSock, si configura per l'operazione *set* e invia la richiesta all'agent. Ricevuta la risposta dall'agent ne viene analizzata la correttezza e avvisato l'utente sul risultato.

Tornando all'analisi dell'interfaccia di GAIASetup, la prima operazione di immettere l'IP-address della stazione e poi cliccare sul pulsante "DOWNLOAD SETUP" (fig. 13), permette il download dei parametri ed il conseguente popolamento dei vari campi predisposti, che possono essere classificati in appartenenti all'acquisitore AGDF2 o alla scheda TN2.



Figura 13. Particolare della schermata di setup con l'indirizzo IP.

Il termine del download è segnalato dal seguente messaggio:



Se questa finestra non compare dopo qualche secondo e il riempimento dei campi non riesce a completarsi allora cliccare nuovamente su "DOWNLOAD SETUP".

3.3.1 Parametri AGDF2

In primo piano nella schermata di setup, sono presenti tutti i parametri relativi alla scheda AGDF2, mentre si accede per mezzo del pulsante “Parametri TN2”, alla schermata relativa a quelli del TN-2.

Per quanto riguarda l’acquisizione, i parametri modificabili sono i classici e cioè (fig. 14): numero di canali (da 1 a 4) e relativi codici, frequenza di campionamento (da 40 sps a 500 sps), fondo scala (da 250 mV a 40000 mV), nome di stazione, velocità della seriale di comunicazione con il TN-2. Inoltre c’è la selezione della modalità di comunicazione col modulo GPS (interno, esterno RS232 e RS485), il numero di canali (da 0 a 4) e la frequenza di campionamento (da 40 sps a 500 sps) dell’eventuale scheda di espansione canali.

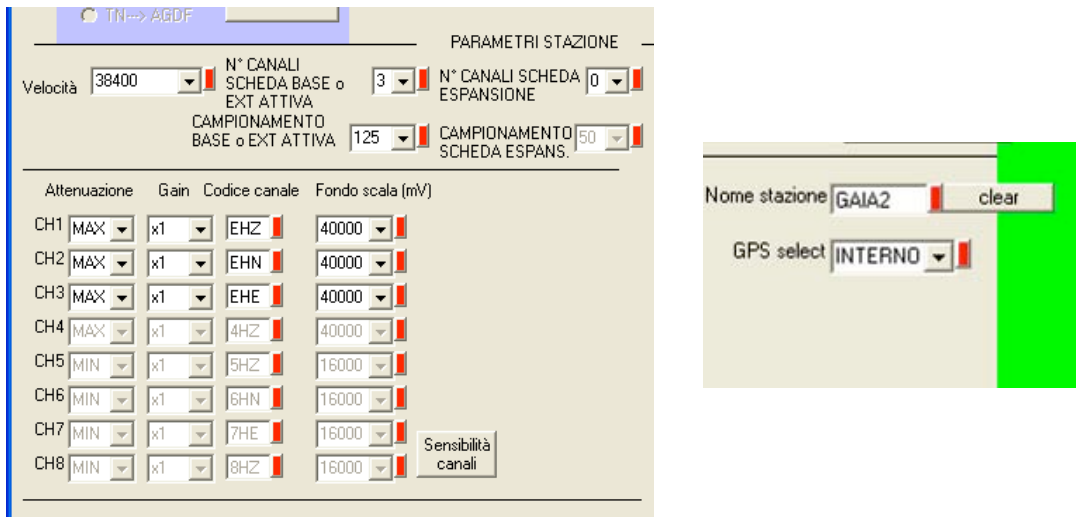


Figura 14. Particolari della schermata principale.

Nel momento in cui si seleziona la modalità per il GPS esterno appariranno sullo schermo (figg. 15 e 16) le configurazioni dei jumper che l’utente deve settare sulla schedina denominata GPS2, per il corretto funzionamento della telemetria tra GAIA2 e modulo GPS: i jumper da modificare sono JP6, JP7, JP8 e JP9.

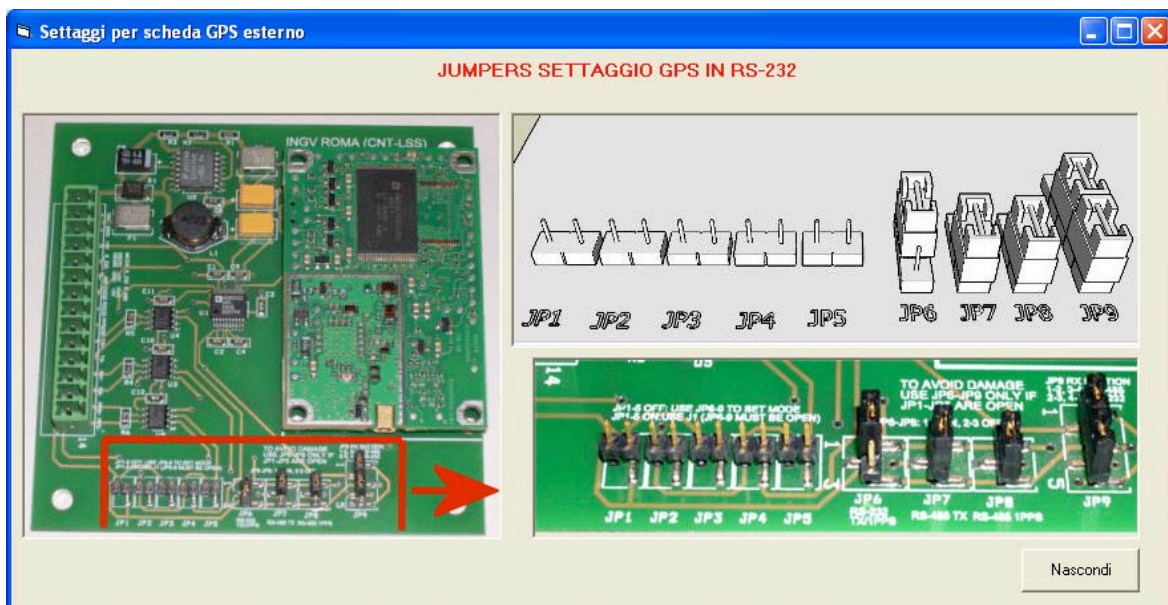


Figura 15. Configurazione GPS in RS-232.

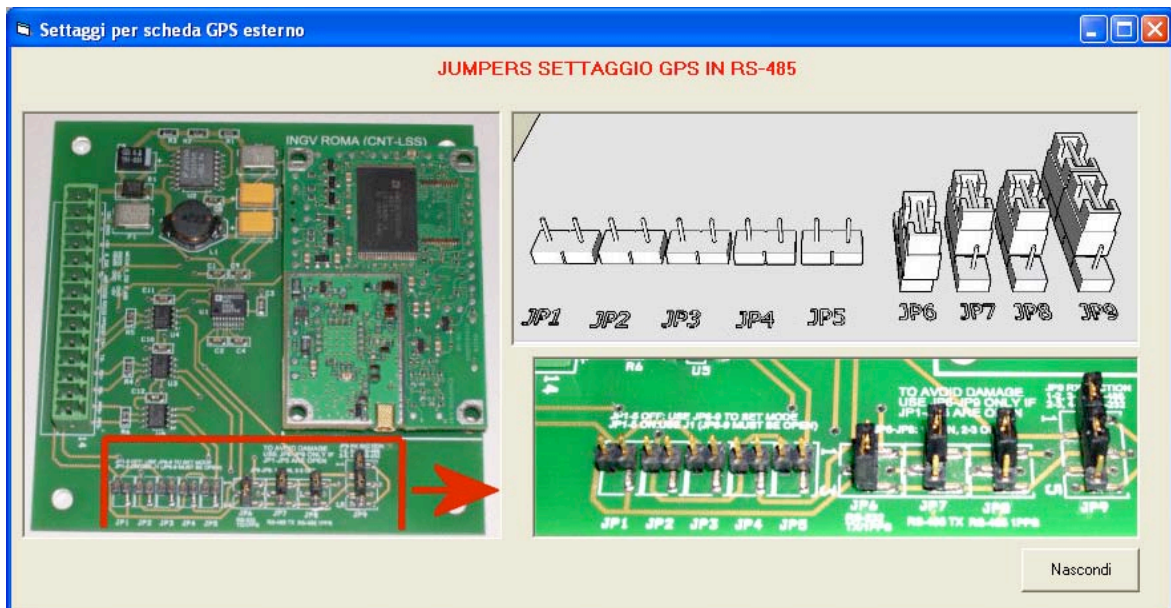


Figura 16. Configurazione GPS in RS-485.

Dal pannello di setup generale, è possibile avere informazioni sullo stato dell'AGDF premendo sul pulsante "AGDF Status" il quale aprirà la relativa finestra di controllo (fig.17): da cui è possibile avere un'indicazione sul corretto funzionamento del GPS (numero di satelliti visibili e agganciati), sul livello di tensione di alimentazione della GAIA2 e sul numero dei pacchetti acquisiti dalla scheda TN-2.

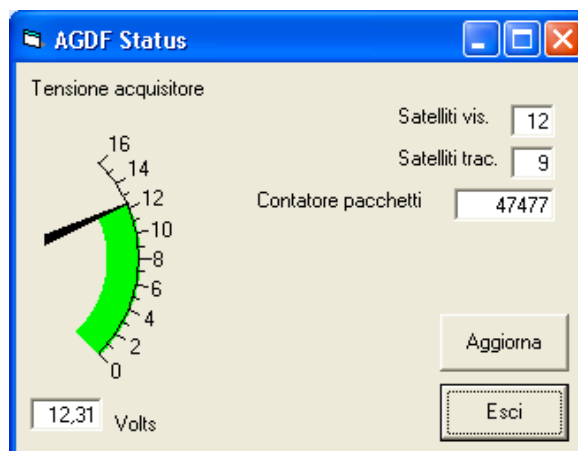
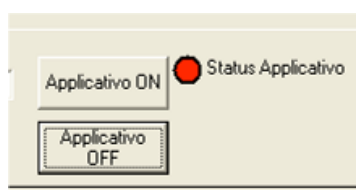


Figura 17. Finestra AGDF Status.

Per aggiornare i dati è sufficiente premere sul pulsante "Aggiorna". Effettuando questa operazione diverse volte è possibile, ad esempio, controllare che il contatore dei pacchetti venga incrementato ogni volta.

Nel caso ciò non accada l'applicativo che gestisce l'acquisizione potrebbe essere fermo ed i pacchetti non venire prodotti; ciò è deducibile dal colore del led "Status applicativo" nella schermata principale:

- verde** : applicativo in funzione
- rosso** : applicativo fermo



Se l'applicativo di acquisizione della TN2 sta girando, il led dello status risulterà verde e la zona di sincronizzazione risulterà disabilitata perché la seriale di comunicazione è occupata (fig. 18).



Figura 18. Particolare della sezione di sincronizzazione.

C'è da notare che i dati visualizzati nella schermata principale sono quelli memorizzati nel TN-2 e ogni volta che viene fatta una modifica devono essere sincronizzati con l'AGDF2: è proprio il compito della sezione di sincronizzazione.

Quest'ultima, come già accennato, è disabilitata fino a che il programma di acquisizione gira sul TN-2: è sufficiente premere il pulsante "Applicativo OFF" per attivarla.

In caso di nuova installazione o ogni volta che viene cambiata la AGDF2, bisogna sincronizzare "nel verso" AGDF → TN: la scheda TN-2 esegue delle operazioni per fermare il normale funzionamento dell'AGDF2 e farsi spedire il setup. Viceversa quando si deve aggiornare il setup dell'AGDF2, si modificano i valori dei parametri, si effettua l'operazione di *set* premendo il pulsantino rosso e aspettando la sua commutazione a verde (fig. 19) che ci informa dell'esito positivo della richiesta.



Figura 19. Esempio di cambio di nome alla stazione.

Dopo aver modificato allo stesso modo tutti i campi necessari, selezionando "TN → AGDF", si preme il pulsante "Sincronizza": il TN-2 crea un file di setup, ferma l'AGDF2 dal normale funzionamento e glielo spedisce. Dopo entrambe le sincronizzazioni il programma informa dell'avvenuta operazione con un messaggio (fig. 20). La sincronizzazione AGDF → TN si chiama **Setup Force Load**, mentre quella TN → AGDF, **Setup Apply**.

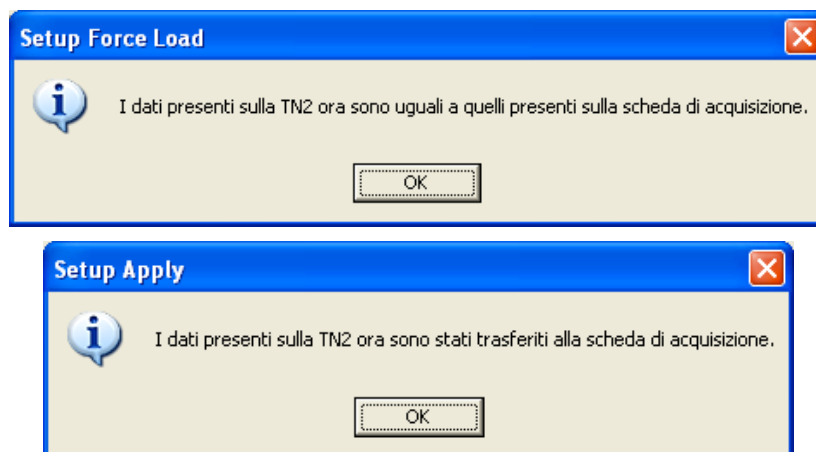


Figura 20. Finestre informative di fine sincronizzazione.

In caso di presenza del modulo di espansione denominato AGDF2ext attiva, si usano i due parametri denominati “HD config” e “Parametri della scheda” (fig. 21).

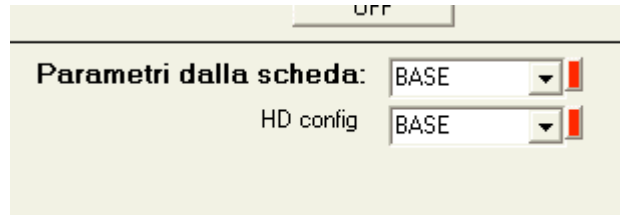


Figura 21. Particolare della sezione per la scheda di espansione.

Il primo permette di informare il TN-2 della presenza della scheda attiva e avvia una sincronizzazione con quest’ultima (infatti dopo aver premuto il pulsante rosso questo diventa verde dopo una ventina di secondi, chiaramente se è presente l’hardware – scheda espansione 4 canali attiva- fig. 22), il secondo predispose il TN-2 a fornire i valori di setup della scheda attiva quando viene premuto “DOWNLOAD SETUP”. Per modificare il campo “HD config” è necessario, come nel caso della sincronizzazione dei parametri, fermare l’applicativo (infatti se l’applicativo è attivo la finestrella “HD config” non è selezionabile), mentre sarà possibile scaricare i dati dalla BASE o dalla EXT ATTIVA in qualunque momento.



Figura 22. Messaggio di avvio sincronizzazione con l’espansione.

Il pulsante “Sensibilità canali” ci indica la sensibilità in μV su counts dei vari canali di acquisizione a seconda delle impostazioni sul fondo scala (fig. 23).

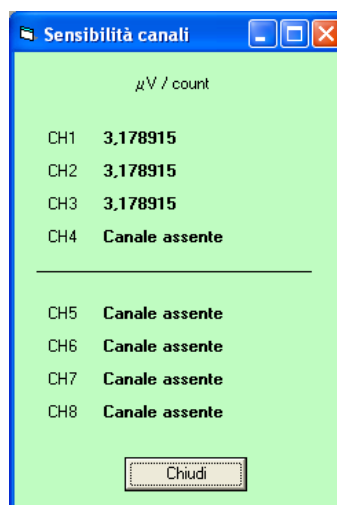


Figura 23. Sensibilità dei canali.

3.3.2 Parametri TN2

Premendo il pulsante “Parametri TN2” si accede ad una finestra (fig. 24) dove trovano posto i parametri di rete della GAIA2 e per la telemetria in protocollo INGV, con la possibilità di scegliere i dati da spedire (selezionando la relativa scheda), la selezione del supporto di memorizzazione dei dati e la sua occupazione di spazio.

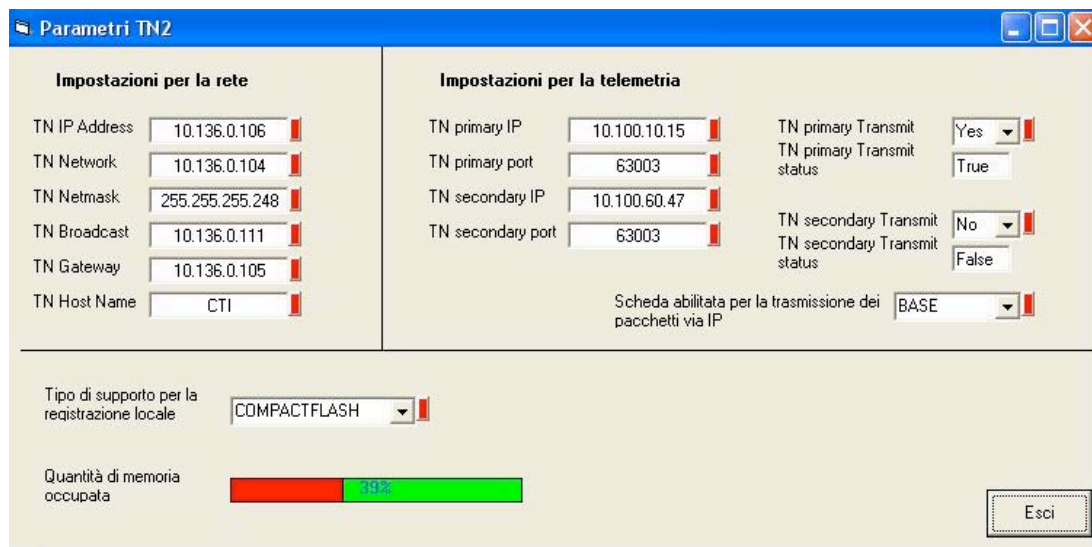


Figura 24. Parametri TN2.

Nella sezione “Impostazioni per la rete” i parametri di rete del TN-2 in dettaglio sono:

TN IP Address: indirizzo IP della stazione

TN Network: indirizzo IP della rete, in genere il primo indirizzo della sottorete assegnata dal provider

TN Netmask: netmask (RUPA: 255.255.255.248, SatLink: 255.255.255.128)

TN Broadcast: indirizzo IP di invio dei messaggi broadcast

TN Gateway: indirizzo IP del router in sito

TN Host Name: nome host della stazione

Nota: il piano di indirizzamento dei siti dovrebbe essere il più possibile uniforme, a tal scopo si raccomanda di seguire ove possibile lo schema della tabella 1, distinto in collegamento RUPA e collegamento SatLink:

Parametro	RUPA	SatLink
TN Network	address (assegnato)	address (assegnato)
TN Gateway	address+1 (assegnato)	address+1 (assegnato)
TN IP Address	address+2	address+2
TN Broadcast	address+6 (assegnato)	address+127 (assegnato)
Indirizzi disponibili	address+3, address+5	da address+3 a address+126

Tabella 1. Schema di indirizzamento per le GAIA.

Nella sezione “Impostazioni per la telemetria” i parametri in dettaglio sono:

TN primary IP: indirizzo IP della macchina primaria dell’acquisizione a cui trasmettere i dati;

TN primary Transmit: abilita o disabilita la trasmissione verso la macchina di acquisizione primaria;

TN primary port: porta della macchina remota primaria dell'acquisizione a cui trasmettere i dati;

TN primary Transmit status: stato della trasmissione verso la macchina di acquisizione primaria;

TN secondary IP: indirizzo IP della macchina secondaria d'acquisizione a cui trasmettere i dati;

TN secondary Transmit: abilita o disabilita la trasmissione verso la macchina di acquisizione secondaria;

TN secondary port: porta della macchina remota secondaria dell'acquisizione a cui trasmettere i dati;

TN secondary Transmit status: stato della trasmissione verso la macchina di acquisizione secondaria;

Nota: In genere si attiva solo la trasmissione sulla macchina primaria, si tenga presente che attivando anche la spedizione dei pacchetti sulla macchina secondaria si raddoppia il flusso dati sul canale di comunicazione con conseguente dimezzamento della banda utile utilizzabile per la singola trasmissione.

Scheda abilitata per la trasmissione dei pacchetti via IP: indica quali dati verranno spediti su IP, secondo le seguenti opzioni:

- **NESSUNA:** non trasmette nessun dato
- **BASE:** trasmette i dati provenienti dalla AGDF base e da una eventuale espansione a quattro canali passiva
- **ATTIVA EXT:** trasmette i dati provenienti dalla sola espansione a quattro canali attiva
- **ENTRAMBE:** trasmette i dati provenienti dalla AGDF base e dalla eventuale espansione a quattro canali attiva

L'ultima sezione riguarda la registrazione dei dati in locale:

Tipo di supporto per la registrazione locale: permette di scegliere il supporto di registrazione locale tra **COMPACTFLASH** e **USBDISK**;

Quantità di memoria occupata: indicatore della memoria occupata sul supporto di registrazione locale.

Per modificare i parametri della scheda TN-2 è sufficiente impostare il nuovo valore, nella relativa sezione, e premere il pulsantino rosso senza ulteriori operazioni.

Tornando alla schermata principale, il pulsante "Real Time Plot" ci fornisce una visualizzazione in tempo reale del segnale sismico sfruttando la pagina web della stazione e il software SeisGram2K di A. Lomax usato come applet java. Sono presenti inoltre i pulsanti per salvare e richiamare il setup completo su un file di testo (vedi appendice E). Vi è infine un menù a tendina (fig. 25) che da la possibilità di avviare una connessione SSH con la scheda TN2 (Prompt TN2), di riavviarla (Reboot), un tool per creare un file con le coordinate delle stazioni compatibile con il TOM TOM™ (TomTom's POI), un tool per il taglio temporale di segnali miniSeed (WINSeed), un tool per il dialogo con l'agent SNMP installato sulla GAIA2 ed infine la storia delle versioni del programma (Version History).

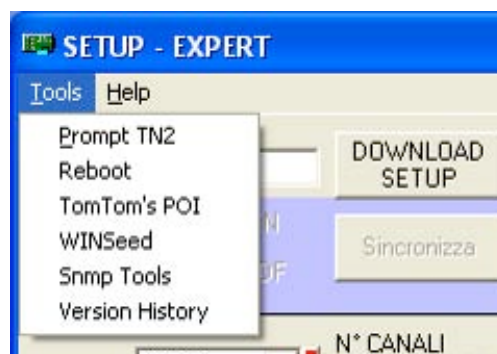


Figura 25. Particolare del menù Tools.

Conclusioni

Il presente lavoro ha fatto emergere il ruolo centrale del modulo TN-2 nelle varie operazioni di configurazione dell'acquisitore AGDF2 e nelle diverse fasi di trasmissione del dato sismico. Si è visto come il protagonista sia il potente protocollo SNMP che, unitamente al programma GAIASetup, permette all'utente di operare con la stazione sismica e di variarne i parametri di configurazione secondo le proprie esigenze.

La produzione di GAIA2 ha superato ad oggi le 300 unità suddivise in Rete Sismica Nazionale, progetti scientifici (Alta Val Tiberina), di monitoraggio e protezione civile (Rete Marche), collaborazioni e convenzioni con università (Parma, Modena, Cosenza ecc.), osservatori (Monteporzio, Varese, Ximeniano ecc.), enti privati (ENEL) ecc. Proprio l'interazione con soggetti esterni all'INGV, ha permesso lo sviluppo e il miglioramento di GAIASetup rendendolo sempre più completo e, soprattutto, semplice da utilizzare.

A pag. 28 abbiamo visto come sia possibile con GAIASetup avviare una console per collegarsi al Linux della TN-2 e permettere all'utente più esperto di lavorare direttamente all'interno del suo sistema operativo. La futura evoluzione dei programmi potrebbe integrare nel protocollo SNMP alcuni comandi che al momento possono essere eseguiti solo nella suddetta modalità console, per poi essere lanciati attraverso opportuni pulsanti dal programma GAIASetup. Operando in questa direzione si offrirà all'utente uno strumento sempre più facile da utilizzare con il massimo controllo di ogni suo parametro senza la necessità di particolari ed avanzate conoscenze in campo di sistemi operativi o di networking.

4. Appendice A: Comandi Redboot

Scrivendo **help** a riga di comando, *RedBoot* restituisce la lista di tutti i comandi con la loro sintassi e una piccola spiegazione:

```
RedBoot> help
Manage aliases kept in FLASH memory
alias name [value]
Manage machine caches
cache [ON | OFF]
Display/switch console channel
channel [-1]<channel number>]
Compute a 32bit checksum [POSIX algorithm] for a range of memory
cksum -b <location> -l <length>
Display (hex dump) a range of memory
dump -b <location> [-l <length>] [-s] [-1|2|4]
Execute an image - with MMU off
exec [-w timeout] [-b <load addr> [-l <length>]]
[-r <ramdisk addr> [-s <ramdisk length>]]
[-c "kernel command line"] [<entry_point>]
Manage FLASH images
fis {cmds}
Manage configuration kept in FLASH memory
fconfig [-i] [-l] [-n] [-f] [-d] | [-d] nickname [value]
Execute code at a location
go [-w <timeout>] [entry]
Help about help?
help [<topic>]
Set/change IP addresses
ip_address [-l <local_ip_address>] [-h <server_address>]
Load a file
load [-r] [-v] [-h <host>] [-m <varies>] [-c <channel_number>]
[-b <base_address>] <file_name>
Compare two blocks of memory
mcmp -s <location> -d <location> -l <length> [-1|-2|-4]
Fill a block of memory with a pattern
mfill -b <location> -l <length> -p <pattern> [-1|-2|-4]
Network connectivity test
ping [-v] [-n <count>] [-l <length>] [-t <timeout>] [-r <rate>]
[-i <IP_addr>] -h <IP_addr>
Reset the system
reset
```

Display RedBoot version information

version

Display (hex dump) a range of memory

x -b <location> [-l <length>] [-s] [-1|2|4]

Per ulteriori approfondimenti si rimanda alla guida utente di RedBoot.

5. Appendice B: Terminale seriale per Linux (accesso di emergenza)

Per l'accesso d'emergenza a Linux, il TN-2 è dotato di un accesso al sistema dalla seriale di servizio.

I parametri della seriale da impostare in un programma di terminale seriale (in fig 25 si vede la schermata di TeraTerm) devono essere i seguenti:

Baud rate: 57600 bit/s

bit di dati: 8

parità: N

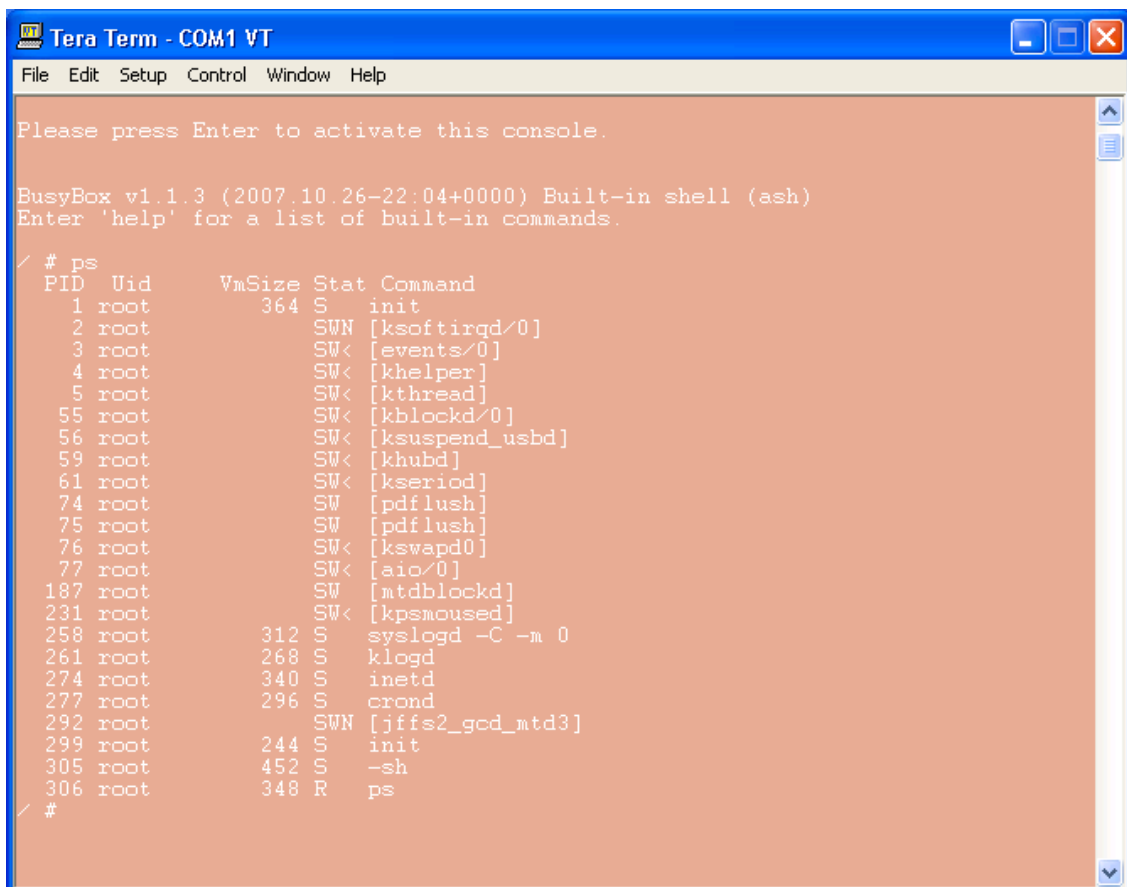
bit di stop: 1

flow control: nessuno

Per accedere a Linux è sufficiente premere il pulsante "invio" e sullo schermo del terminale apparirà il prompt, a questo punto si possono utilizzare tutti i comandi di Linux:

- **ifconfig** "Fornisce l'indirizzo IP della stazione"
- **ping** "Verso router locale" e "Verso server remoto (Roma)"
- **cat /dev/ttyS1** "Controllo del flusso dati dalla seriale collegata al modulo AGDFBase"
- **ps** "Elenco dei processi attivi" (fig. 25)
- **exit** "Uscita dalla console di Linux"

....



```
Tera Term - COM1 VT
File Edit Setup Control Window Help

Please press Enter to activate this console.

BusyBox v1.1.3 (2007.10.26-22:04+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

# ps
PID Uid      VmSize Stat Command
  1 root        364 S   init
  2 root           SWN [ksoftirqd/0]
  3 root           SW< [events/0]
  4 root           SW< [khelper]
  5 root           SW< [kthread]
 55 root           SW< [kblockd/0]
 56 root           SW< [ksuspend_usbd]
 59 root           SW< [khubd]
 61 root           SW< [kseriod]
 74 root           SW  [pdflush]
 75 root           SW  [pdflush]
 76 root           SW< [kswapd0]
 77 root           SW< [aio/0]
187 root           SW  [mtblockd]
231 root           SW< [kpsmoused]
258 root          312 S   syslogd -C -m 0
261 root          268 S   klogd
274 root          340 S   inetd
277 root          296 S   crond
292 root           SWN [jffs2_gcd_mtd3]
299 root          244 S   init
305 root          452 S   -sh
306 root          348 R   ps
#
```

Figura 26. Elenco processi dopo l'esecuzione del comando ps.

6. Appendice C: Script PrgArm

Di seguito trova spazio l'intero script per l'installazione dei firmware del TN-2, con il programma PrgArm. In sequenza esegue la formattazione della memoria flash della TN2, configura i parametri di rete di Redboot, scarica le due immagini (kernel e file system Linux) e le installa nella flash, configura lo script di avvio di Redboot per l'avvio di Linux.

```
# 30/10/2007
print 'delete'

send fconfig -i
#Initialize non-volatile configuration - continue (y/n)?
expect {
  continue (y/n)?
  timeout 2 goto expect_timeout
}
send y
#Run script at boot: false
expect {
  false
  timeout 2 goto expect_timeout
}
send
#Use BOOTP for network configuration: false
expect {
  Use
  timeout 2 goto expect_timeout
}
send \b\b\b\b\b\b
send false
#Gateway IP address:
expect {
  Gateway
  timeout 2 goto expect_timeout
}
send
#Local IP address:
expect {
  Local
  timeout 2 goto expect_timeout
}
send
#Local IP address mask:
expect {
  Local
  timeout 2 goto expect_timeout
}
send
#Default server IP address:
expect {
  Default
  timeout 2 goto expect_timeout
}
send
#DNS server IP address:
expect {
  DNS
  timeout 2 goto expect_timeout
}
send
#Network hardware address [MAC]:
```

```

expect {
[MAC]:
timeout 2 goto expect_timeout
}
send
#GDB connection port: 9000
expect {
GDB
timeout 2 goto expect_timeout
}
send
#Force console for special debug messages: false
expect {
Force
timeout 2 goto expect_timeout
}
send
#Network debug at boot time: false
expect {
Network
timeout 2 goto expect_timeout
}
send
#Update RedBoot non-volatile configuration - continue (y/n)?
expect {
Update
timeout 5 goto expect_timeout
}
send y
expect {
RedBoot>
timeout 5 goto expect_timeout
}
send reset
expect {
RedBoot>
timeout 700 goto expect_timeout
}
print 'erase flash'
send fis erase -f 0x60040000 -l 0x3F40000
expect {
RedBoot>
timeout 700 goto expect_timeout
}
send
expect {
RedBoot>
timeout 700 goto expect_timeout
}
send fis init
expect {
continue (y/n)?
timeout 600 goto expect_timeout
}
send y
expect {
RedBoot>
timeout 600 goto expect_timeout
}
send fconfig -i
#Initialize non-volatile configuration - continue (y/n)?

```



```

sleep 1
expect {
  continue (y/n)?
  timeout 20 goto expect_timeout
}
send y
#Run script at boot: false
expect {
  boot:
  timeout 2 goto expect_timeout
}
sleep 2
send \b\b\b\b\b\b
send false
#Use BOOTP for network configuration: true
expect {
  configuration:
  timeout 2 goto expect_timeout
}
send \b\b\b\b\b\b
send false
#Gateway IP address:
expect {
  address:
  timeout 2 goto expect_timeout
}
send
#Local IP address:
expect {
  Local
  timeout 2 goto expect_timeout
}
send \b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b
send 10.10.10.101
#Local IP address mask:
expect {
  mask:
  timeout 2 goto expect_timeout
}
sleep 1
send \b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b
send 255.0.0.0

#Default server IP address:
expect {
  address:
  timeout 2 goto expect_timeout
}
#send IPSERVER
send \b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b
send 10.100.10.14
#DNS server IP address:
expect {
  address:
  timeout 2 goto expect_timeout
}
send
#Network hardware address [MAC]:
expect {
  [MAC]:
  timeout 4 goto expect_timeout
}

```

```

}
send \b\b\b\b\b\b
send true
#eth0 network hardware address [MAC]: 0x00:0x00:0x00:0x00:0x68:0xB4
expect {
[MAC]:
timeout 4 goto expect_timeout
}
send
#GDB connection port: 9000
expect {
GDB
timeout 4 goto expect_timeout
}
send
#Force console for special debug messages: false
expect {
messages:
timeout 2 goto expect_timeout
}
send
#Network debug at boot time: false
expect {
time:
timeout 2 goto expect_timeout
}
send
#Update RedBoot non-volatile configuration - continue (y/n)?
expect {
continue (y/n)?
timeout 5 goto expect_timeout
}
send y
expect {
RedBoot>
timeout 5 goto expect_timeout
}
send reset
expect {
RedBoot>
timeout 180 goto expect_timeout
}
print 'load root_tn2'
send load -v -r -b 0x800000 root_tn2-1.1.gz
expect {
RedBoot>
timeout 600 goto expect_timeout
}
print 'create'
send fis create -b 0x800000 -l 0xAF1BD6 root_tn2.gz
expect {
RedBoot>
timeout 600 goto expect_timeout
}
print 'load zImage'
send load -v -r -b 0x80000 zImage
expect {
RedBoot>
timeout 600 goto expect_timeout
}
send fis create -b 0x80000 -l 0x124450 zImage

```

```

expect {
RedBoot>
timeout 600 goto expect_timeout
}
print 'End of load zImage'
send fconfig -i
#Initialize non-volatile configuration - continue (y/n)?
expect {
continue (y/n)?
timeout 2 goto expect_timeout
}
send y
#Run script at boot: false
expect {
false
timeout 2 goto expect_timeout
}
send \b\b\b\b\b\b
send true
expect {
>>
timeout 2 goto expect_timeout
}
send fis load root_tn2.gz
expect {
>>
timeout 2 goto expect_timeout
}
send fis load zImage
expect {
>>
timeout 4 goto expect_timeout
}
send exec -r 0x800000 -s 0xB00000 -c "root=/dev/ram ramdisk=34000 console=ttyAM0"
expect {
>>
timeout 2 goto expect_timeout
}
send
#Boot script timeout (1000ms resolution): 0
expect {
resolution
timeout 2 goto expect_timeout
}
send \b
send 2
#Use BOOTP for network configuration: false
expect {
Use
timeout 2 goto expect_timeout
}
send \b\b\b\b\b\b
send false
#Gateway IP address:
expect {
Gateway
timeout 2 goto expect_timeout
}
send
#Local IP address:
expect {

```

```

Local
timeout 2 goto expect_timeout
}
send
#Local IP address mask:
expect {
Local
timeout 2 goto expect_timeout
}
send
#Default server IP address:
expect {
Default
timeout 2 goto expect_timeout
}
send
#DNS server IP address:
expect {
DNS
timeout 2 goto expect_timeout
}
send
#Network hardware address [MAC]:
expect {
[MAC]:
timeout 2 goto expect_timeout
}
send
#GDB connection port: 9000
expect {
GDB
timeout 2 goto expect_timeout
}
send
#Force console for special debug messages: false
expect {
Force
timeout 2 goto expect_timeout
}
send
#Network debug at boot time: false
expect {
Network
timeout 2 goto expect_timeout
}
send
#Update RedBoot non-volatile configuration - continue (y/n)?
expect {
Update
timeout 5 goto expect_timeout
}
send y
expect {
RedBoot>
timeout 5 goto expect_timeout
}
print 'End of configuration'
send reset
exit

```

7. Appendice D: Descrizione stringa di setup dell'AGDF2

In questa appendice si riportano le locazioni della stringa di setup con i valori dei dati e le descrizioni dei parametri di stazione.

Loc	Def	Dato	Descrizione
00H	41H	Carattere ascii es. A = 41H , B = 42H	Nome stazione carattere 1
01H	47H	Carattere ascii es. A = 41H , B = 42H	Nome stazione carattere 2
02H	44H	Carattere ascii es. A = 41H , B = 42H	Nome stazione carattere 3
03H	46H	Carattere ascii es. A = 41H , B = 42H	Nome stazione carattere 4
04H	20H	Carattere ascii es. A = 41H , B = 42H	Nome stazione carattere 5
05H	31H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 1 ch1
06H	48H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 2 ch1
07H	5AH	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 3 ch1
08H	32H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 1 ch2
09H	48H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 2 ch2
0AH	5AH	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 3 ch2
0BH	33H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 1 ch3
0CH	48H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 2 ch3
0DH	5AH	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 3 ch3
0EH	34H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 1 ch4
0FH	48H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 2 ch4
10H	5AH	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 3 ch4
11H	35H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 1 ch5
12H	48H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 2 ch5
13H	5AH	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 3 ch5
14H	36H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 1 ch6
15H	48H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 2 ch6
16H	5AH	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 3 ch6
17H	37H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 1 ch7
18H	48H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 2 ch7
19H	5AH	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 3 ch7
1AH	38H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 1 ch8
1BH	48H	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 2 ch8
1CH	5AH	Carattere ascii es. A = 41H , B = 42H	Codice canale carattere 3 ch8
1DH	00H	Da 00H a FFH	Fondo scala in mV MSB ch1
1EH	00H	Da 00H a FFH	Fondo scala in mV LSB ch1
1FH	00H	Da 00H a FFH	Fondo scala in mV MSB ch2
20H	00H	Da 00H a FFH	Fondo scala in mV LSB ch2
21H	00H	Da 00H a FFH	Fondo scala in mV MSB ch3
22H	00H	Da 00H a FFH	Fondo scala in mV LSB ch3
23H	00H	Da 00H a FFH	Fondo scala in mV MSB ch4
24H	00H	Da 00H a FFH	Fondo scala in mV LSB ch4
25H	00H	Da 00H a FFH	Fondo scala in mV MSB ch5
26H	00H	Da 00H a FFH	Fondo scala in mV LSB ch5
27H	00H	Da 00H a FFH	Fondo scala in mV MSB ch6
28H	00H	Da 00H a FFH	Fondo scala in mV LSB ch6
29H	00H	Da 00H a FFH	Fondo scala in mV MSB ch7
2AH	00H	Da 00H a FFH	Fondo scala in mV LSB ch7
2BH	00H	Da 00H a FFH	Fondo scala in mV MSB ch8
2CH	00H	Da 00H a FFH	Fondo scala in mV LSB ch8
2DH	00H	Reserved	
2EH	00H	Reserved	
2FH	00H	Reserved	
30H	00H	Reserved	
31H	00H	Reserved	
32H	41H	Carattere ascii es. A = 41H , B = 42H	1Versione setup - Configurazione Hardware
33H	47H	Carattere ascii es. A = 41H , B = 42H	2Versione setup - Configurazione di acquisizione
34H	44H	Carattere ascii es. A = 41H , B = 42H	3Versione setup - Configurazione di acquisizione

35H 46H	Carattere ascii es. A = 41H , B = 42H	4	Versione setup - Configurazione di acquisizione
36H 46H	Carattere ascii es. A = 41H , B = 42H	5	Versione setup - Configurazione di acquisizione
37H 31H	Carattere ascii es. A = 41H , B = 42H	6	Versione setup - Progressivo s/n MSB
38H 32H	Carattere ascii es. A = 41H , B = 42H	7	Versione setup - Progressivo s/n CSB
39H 34H	Carattere ascii es. A = 41H , B = 42H	8	Versione setup - Progressivo s/n LSB
3AH - 51H	Reserved		23 byte riservati per usi futuri
52H 00H	0000 Bit 23:20 0000 Bit 19:16		Filter coefficient selection ch1 – ch4 MSB
53H 00H	IIR2 0000 3Hz@2000SPS Bit 15:12 0001 3Hz@1000SPS Bit 15:12 0010 3Hz@500SPS Bit 15:12 0011 3Hz@333SPS Bit 15:12 0100 3Hz@250SPS Bit 15:12 IIR1 0000 3Hz@2000SPS Bit 11:8 0001 3Hz@1000SPS Bit 11:8 0010 3Hz@500SPS Bit 11:8 0011 3Hz@333SPS Bit 11:8 0100 3Hz@250SPS Bit 11:8		Filter coefficient selection ch1 – ch4 CSB IIR1 & IIR2 Coefficients
54H 00H	FIR2 0000 Linear Phase Bit 7:4 0001 Minimum Phase Bit 7:4 FIR1 0000 Linear Phase Bit 3:0 0001 Minimum Phase Bit 3:0		Filter coefficient selection ch1 – ch4 LSB FIR1 & FIR2 Coefficients
55H 00H	0000 Bit 23:20 0000 Bit 19:16		Filter coefficient selection ch5 – ch8 MSB
56H 00H	IIR2 0000 3Hz@2000SPS Bit 15:12 0001 3Hz@1000SPS Bit 15:12 0010 3Hz@500SPS Bit 15:12 0011 3Hz@333SPS Bit 15:12 0100 3Hz@250SPS Bit 15:12 IIR1 0000 3Hz@2000SPS Bit 11:8 0001 3Hz@1000SPS Bit 11:8 0010 3Hz@500SPS Bit 11:8 0011 3Hz@333SPS Bit 11:8 0100 3Hz@250SPS Bit 11:8		Filter coefficient selection ch5 – ch8 CSB IIR1 & IIR2 Coefficients
57H 00H	FIR2 0000 Linear Phase Bit 7:4 0001 Minimum Phase Bit 7:4 FIR1 0000 Linear Phase Bit 3:0 0001 Minimum Phase Bit 3:0		Filter coefficient selection ch5 – ch8 LSB FIR1 & FIR2 Coefficients
58h 00H	000 reserved Bit 23:21 00000 EXP Bit 20:16		Filter configuration register ch1 – ch4 MSB OFFSET calibration exponent
59H 02H	0 reserved Bit 15 Run OFFSET calibration 1 Enable Bit 14 0 Disable Bit 14 Use OFFSET correction 1 Enable Bit 13 0 Disable Bit 13 Use GAIN correction 1 Enable Bit 12 0 Disable Bit 12 0 reserved Bit 11 Output filter stage select		Filter configuration register ch1 – ch4 CSB Run OFFSET calibration & Use OFFSET correction & Use GAIN correction & Output filter stage select

	111 reserved Bit 10:8 110 reserved Bit 10:8 101 IIR 3 rd Order Bit 10:8 100 IIR 2 nd Order Bit 10:8 011 IIR 1 st Order Bit 10:8 010 FIR2 Output Bit 10:8 001 FIR1 Output Bit 10:8 000 SINC Output Bit 10:8	
5AH E3H	Decimation selection 0111 reserved Bit 7:4 0110 reserved Bit 7:4 0101 reserved Bit 7:4 0100 500 SPS Bit 7:4 0011 reserved Bit 7:4 0010 250 SPS Bit 7:4 0001 200 SPS Bit 7:4 0000 125 SPS Bit 7:4 1111 100 SPS Bit 7:4 1110 50 SPS Bit 7:4 1101 40 SPS Bit 7:4 1100 reserved Bit 7:4 1011 reserved Bit 7:4 1010 reserved Bit 7:4 1001 reserved Bit 7:4 1000 reserved Bit 7:4 0 reserved Bit 3 Channel Enable 111 reserved Bit 2:0 110 reserved Bit 2:0 101 reserved Bit 2:0 100 4 Channel (1,2,3,4) Bit 2:0 011 3 Channel (1,2,3) Bit 2:0 010 2 Channel (1,2) Bit 2:0 001 1 Channel (1 only) Bit 2:0 000 0 Channel Bit 2:0	Filter configuration register ch1 – ch4 LSB Decimation selection & Channel Enable
5BH 00H	000 reserved Bit 23:21 00000 EXP Bit 20:16	Filter configuration register ch5 – ch8 MSB OFFSET calibration exponent
5CH 02H	0 reserved Bit 15 Run OFFSET calibration 1 Enable Bit 14 0 Disable Bit 14 Use OFFSET correction 1 Enable Bit 13 0 Disable Bit 13 Use GAIN correction 1 Enable Bit 12 0 Disable Bit 12 0 reserved Bit 11 Output filter stage select 111 reserved Bit 10:8 110 reserved Bit 10:8 101 IIR 3 rd Order Bit 10:8 100 IIR 2 nd Order Bit 10:8 011 IIR 1 st Order Bit 10:8 010 FIR2 Output Bit 10:8 001 FIR1 Output Bit 10:8 000 SINC Output Bit 10:8	Filter configuration register ch5 – ch8 CSB Run OFFSET calibration & Use OFFSET correction & Use GAIN correction & Output filter stage select
5DH E0H	Decimation selection 0111 reserved Bit 7:4 0110 reserved Bit 7:4 0101 reserved Bit 7:4	Filter configuration register ch5 – ch8 LSB Decimation selection & Channel Enable

	0100 500 SPS Bit 7:4 0011 reserved Bit 7:4 0010 250 SPS Bit 7:4 0001 200 SPS Bit 7:4 0000 125 SPS Bit 7:4 1111 100 SPS Bit 7:4 1110 50 SPS Bit 7:4 1101 40 SPS Bit 7:4 1100 reserved Bit 7:4 1011 reserved Bit 7:4 1010 reserved Bit 7:4 1001 reserved Bit 7:4 1000 reserved Bit 7:4 0 reserved Bit 3 Channel Enable 111 reserved Bit 2:0 110 reserved Bit 2:0 101 reserved Bit 2:0 100 4 Channel (1,2,3,4) Bit 2:0 011 3 Channel (1,2,3) Bit 2:0 010 2 Channel (1,2) Bit 2:0 001 1 Channel (1 only) Bit 2:0 000 0 Channel Bit 2:0	
5EH 07H	00000 reserved Bit 23:19 Digital Filter Frequency select 111 16.384 MHz Bit 18:16 110 8.192 MHz Bit 18:16 101 4.096 MHz Bit 18:16 100 2.048 MHz Bit 18:16 011 1.024 MHz Bit 18:16 010 512 KHz Bit 18:16 001 256 KHz Bit 18:16 000 32 KHz Bit 18:16	Hardware configuration register ch1 – ch4 MSB Digital Filter Frequency select
5FH 04H	00000 reserved Bit 15:11 MCLK frequency select 111 reserved Bit 10:8 110 reserved Bit 10:8 101 4.096 MHz Bit 10:8 100 2.048 MHz Bit 10:8 011 1.024 MHz Bit 10:8 010 512 KHz Bit 10:8 001 reserved Bit 10:8 000 reserved Bit 10:8	Hardware configuration register ch1 – ch4 CSB MCLK frequency select
60H 11H	00 reserved Bit 7:6 MCLK/2 output enable 1 Enabled Bit 5 0 Disabled Bit 5 MCLK output enable 1 Enabled Bit 4 0 Disabled Bit 4 MDATA input frequency select 1 256 KHz Bit 3 0 512 KHz Bit 3 BOOT source indicator 1 Booted from EEPROM Bit 2 0 Booted from micro Bit 2 MSYNC enable 1 MSYNC generated Bit 1 0 MSYNC remains low Bit 1	Hardware configuration register ch1 – ch4 LSB MCLK , MCLK/2 , MDATA , BOOT , MSYNC
61H 07H	00000 reserved Bit 23:19 Digital Filter Frequency select	Hardware configuration register ch5 – ch8 MSB Digital Filter Frequency select

	111 16.384 MHz Bit 18:16 110 8.192 MHz Bit 18:16 101 4.096 MHz Bit 18:16 100 2.048 MHz Bit 18:16 011 1.024 MHz Bit 18:16 010 512 KHz Bit 18:16 001 256 KHz Bit 18:16 000 32 KHz Bit 18:16	
62H 04H	00000 reserved Bit 15:11 MCLK frequency select 111 reserved Bit 10:8 110 reserved Bit 10:8 101 4.096 MHz Bit 10:8 100 2.048 MHz Bit 10:8 011 1.024 MHz Bit 10:8 010 512 KHz Bit 10:8 001 reserved Bit 10:8 000 reserved Bit 10:8	Hardware configuration register ch5 – ch8 CSB MCLK frequency select
63H 11H	00 reserved Bit 7:6 MCLK/2 output enable 1 Enabled Bit 5 0 Disabled Bit 5 MCLK output enable 1 Enabled Bit 4 0 Disabled Bit 4 MDATA input frequency select 1 256 KHz Bit 3 0 512 KHz Bit 3 BOOT source indicator 1 Booted from EEPROM Bit 2 0 Booted from micro Bit 2 MSYNC enable 1 MSYNC generated Bit 1 0 MSYNC remains low Bit 1	Hardware configuration register ch5 – ch8 LSB MCLK , MCLK/2 , MDATA , BOOT , MSYNC
64H 00H	00H – FFH	Offset Correction MSB ch1
65H 00H	00H – FFH	Offset Correction CSB ch1
66H 00H	00H – FFH	Offset Correction LSB ch1
67H 00H	00H – FFH	Offset Correction MSB ch2
68H 00H	00H – FFH	Offset Correction CSB ch2
69H 00H	00H – FFH	Offset Correction LSB ch2
6AH 00H	00H – FFH	Offset Correction MSB ch3
6BH 00H	00H – FFH	Offset Correction CSB ch3
6CH 00H	00H – FFH	Offset Correction LSB ch3
6DH 00H	00H – FFH	Offset Correction MSB ch4
6EH 00H	00H – FFH	Offset Correction CSB ch4
6FH 00H	00H – FFH	Offset Correction LSB ch4
70H 00H	00H – FFH	Offset Correction MSB ch5
71H 00H	00H – FFH	Offset Correction CSB ch5
72H 00H	00H – FFH	Offset Correction LSB ch5
73H 00H	00H – FFH	Offset Correction MSB ch6
74H 00H	00H – FFH	Offset Correction CSB ch6
75H 00H	00H – FFH	Offset Correction LSB ch6
76H 00H	00H – FFH	Offset Correction MSB ch7
77H 00H	00H – FFH	Offset Correction CSB ch7
78H 00H	00H – FFH	Offset Correction LSB ch7
79H 00H	00H – FFH	Offset Correction MSB ch8
7AH 00H	00H – FFH	Offset Correction CSB ch8
7BH 00H	00H – FFH	Offset Correction LSB ch8
7CH 00H	00H – FFH	Gain Correction MSB ch1
7DH 00H	00H – FFH	Gain Correction CSB ch1

7EH 00H	00H – FFH		Gain Correction LSB ch1
7FH 00H	00H – FFH		Gain Correction MSB ch2
80H 00H	00H – FFH		Gain Correction CSB ch2
81H 00H	00H – FFH		Gain Correction LSB ch2
82H 00H	00H – FFH		Gain Correction MSB ch3
83H 00H	00H – FFH		Gain Correction CSB ch3
84H 00H	00H – FFH		Gain Correction LSB ch3
85H 00H	00H – FFH		Gain Correction MSB ch4
86H 00H	00H – FFH		Gain Correction CSB ch4
87H 00H	00H – FFH		Gain Correction LSB ch4
88H 00H	00H – FFH		Gain Correction MSB ch5
89H 00H	00H – FFH		Gain Correction CSB ch5
8AH 00H	00H – FFH		Gain Correction LSB ch5
8BH 00H	00H – FFH		Gain Correction MSB ch6
8CH 00H	00H – FFH		Gain Correction CSB ch6
8DH 00H	00H – FFH		Gain Correction LSB ch6
8EH 00H	00H – FFH		Gain Correction MSB ch7
8FH 00H	00H – FFH		Gain Correction CSB ch7
90H 00H	00H – FFH		Gain Correction LSB ch7
91H 00H	00H – FFH		Gain Correction MSB ch8
92H 00H	00H – FFH		Gain Correction CSB ch8
93H 00H	00H – FFH		Gain Correction LSB ch8
94H AAH	10 Ch4 attenuator MAX Bit 7:6 01 Ch4 attenuator MIN Bit 7:6 10 Ch3 attenuator MAX Bit 5:4 01 Ch3 attenuator MIN Bit 5:4 10 Ch2 attenuator MAX Bit 3:2 01 Ch2 attenuator MIN Bit 3:2 10 Ch1 attenuator MAX Bit 1:0 01 Ch1 attenuator MIN Bit 1:0		GPIO attenuator ch1 – ch4
95H 55H	10 Ch8 attenuator MAX Bit 7:6 01 Ch8 attenuator MIN Bit 7:6 10 Ch7 attenuator MAX Bit 5:4 01 Ch7 attenuator MIN Bit 5:4 10 Ch6 attenuator MAX Bit 3:2 01 Ch6 attenuator MIN Bit 3:2 10 Ch5 attenuator MAX Bit 1:0 01 Ch5 attenuator MIN Bit 1:0		GPIO attenuator ch5 – ch8
96H AAH	11 Ch4 INA+INB Bit 7:6 10 Ch4 INA only Bit 7:6 01 Ch4 INB only Bit 7:6 00 Ch4 GND terminator Bit 7:6 11 Ch3 INA+INB Bit 5:4 10 Ch3 INA only Bit 5:4 01 Ch3 INB only Bit 5:4 00 Ch3 GND terminator Bit 5:4 11 Ch2 INA+INB Bit 3:2 10 Ch2 INA only Bit 3:2 01 Ch2 INB only Bit 3:2 00 Ch2 GND terminator Bit 3:2 11 Ch1 INA+INB Bit 1:0 10 Ch1 INA only Bit 1:0 01 Ch1 INB only Bit 1:0 00 Ch1 GND terminator Bit 1:0		GPIO mux ch1 – ch4
97H AAH	11 Ch8 INA+INB Bit 7:6 10 Ch8 INA only Bit 7:6 01 Ch8 INB only Bit 7:6 00 Ch8 GND terminator Bit 7:6 11 Ch7 INA+INB Bit 5:4 10 Ch7 INA only Bit 5:4		GPIO mux ch5 – ch8

	01 Ch7 INB only Bit 5:4 00 Ch7 GND terminator Bit 5:4 11 Ch6 INA+INB Bit 3:2 10 Ch6 INA only Bit 3:2 01 Ch6 INB only Bit 3:2 00 Ch6 GND terminator Bit 3:2 11 Ch5 INA+INB Bit 1:0 10 Ch5 INA only Bit 1:0 01 Ch5 INB only Bit 1:0 00 Ch5 GND terminator Bit 1:0	
98H 00H	Ch2 power down 1 Enable Bit 7 0 Disable Bit 7 Ch1 power down 1 Enable Bit 6 0 Disable Bit 6 Ch2 gain 110 x64 gain Bit 5:3 101 x32 gain Bit 5:3 100 x16 gain Bit 5:3 011 x8 gain Bit 5:3 010 x4 gain Bit 5:3 001 x2 gain Bit 5:3 000 x1 gain Bit 5:3 Ch1 gain 110 x64 gain Bit 2:0 101 x32 gain Bit 2:0 100 x16 gain Bit 2:0 011 x8 gain Bit 2:0 010 x4 gain Bit 2:0 001 x2 gain Bit 2:0 000 x1 gain Bit 2:0	GPIO gain & power down ch1 – ch2
99H 00H	Ch4 power down 1 Enable Bit 7 0 Disable Bit 7 Ch3 power down 1 Enable Bit 6 0 Disable Bit 6 Ch4 gain 110 x64 gain Bit 5:3 101 x32 gain Bit 5:3 100 x16 gain Bit 5:3 011 x8 gain Bit 5:3 010 x4 gain Bit 5:3 001 x2 gain Bit 5:3 000 x1 gain Bit 5:3 Ch3 gain 110 x64 gain Bit 2:0 101 x32 gain Bit 2:0 100 x16 gain Bit 2:0 011 x8 gain Bit 2:0 010 x4 gain Bit 2:0 001 x2 gain Bit 2:0 000 x1 gain Bit 2:0	GPIO gain & power down ch3 – ch4
9AH 00H	Ch6 power down 1 Enable Bit 7 0 Disable Bit 7 Ch5 power down 1 Enable Bit 6 0 Disable Bit 6 Ch6 gain	GPIO gain & power down ch5 – ch6

	110 x64 gain Bit 5:3 101 x32 gain Bit 5:3 100 x16 gain Bit 5:3 011 x8 gain Bit 5:3 010 x4 gain Bit 5:3 001 x2 gain Bit 5:3 000 x1 gain Bit 5:3 Ch5 gain 110 x64 gain Bit 2:0 101 x32 gain Bit 2:0 100 x16 gain Bit 2:0 011 x8 gain Bit 2:0 010 x4 gain Bit 2:0 001 x2 gain Bit 2:0 000 x1 gain Bit 2:0	
9BH 00H	Ch8 power down 1 Enable Bit 7 0 Disable Bit 7 Ch7 power down 1 Enable Bit 6 0 Disable Bit 6 Ch8 gain 110 x64 gain Bit 5:3 101 x32 gain Bit 5:3 100 x16 gain Bit 5:3 011 x8 gain Bit 5:3 010 x4 gain Bit 5:3 001 x2 gain Bit 5:3 000 x1 gain Bit 5:3 Ch7 gain 110 x64 gain Bit 2:0 101 x32 gain Bit 2:0 100 x16 gain Bit 2:0 011 x8 gain Bit 2:0 010 x4 gain Bit 2:0 001 x2 gain Bit 2:0 000 x1 gain Bit 2:0	GPIO gain & power down ch7 – ch8
9CH 52H	01010010 Int. GPS Bit 7:0 00110101 Ext. GPS 232 Bit 7:0 01001101 Ext. GPS 485 Bit 7:0	GPIO gps internal – external 232/485 select
9DH 05H	00000101 telemetry 232 Bit 7:0 00001010 telemetry 485 Bit 7:0	GPIO telemetry 232/485 select
9EH FFH	XXdddddd Bit 5:0 d=0 open drain (in) d=1 push pull (out) LEGGERE NOTA 1	Configura la porta controllo sensori Sono utilizzabili solo i bit da 0 a 5 mentre i bit 6 e 7 sono riservati pertanto qualsiasi stato logico ad essi impostato viene ignorato.
9FH CFH	XXdddddd Bit 5:0 d=0 start up livello logico basso d=1 start up livello logico alto LEGGERE NOTA 1	Inizializza lo stato logico della porta controllo sensori. Sono utilizzabili solo i bit da 0 a 5 mentre i bit 6 e 7 sono riservati pertanto qualsiasi stato logico ad essi impostato viene ignorato.
A0H 00H	00000000 sensore sconosciuto 00000001 Lennartz 5 Sec. 00000010 Trillium 40 Sec. 00000011 Trillium 120 Sec. 00000100 Trillium 240 Sec. 00000101 Guralp CMG-3T 00000110 Guralp CMG-40 00000111 STS2 120 Sec. 00001000 Episensor	Tipo di sensore installato
A1H 00H	00H Base da sola oppure con External passiva	Codice identificativo della scheda. Il valore del dato viene riportato nella locazione header 37H di ogni

	01H External attiva	pacchetto di acquisizione.
*****	*****	*****
B0H 00H		
*****	40 caratteri di commento	Da B0 a D7 40 caratteri ci commento
*****	40 caratteri di commento	Da B0 a D7 40 caratteri ci commento
D7H 00H		
*****	*****	*****
*****	*****	*****
*****	*****	*****
*****	*****	*****
*****	*****	*****
*****	*****	*****
*****	*****	*****
*****	*****	*****
*****	*****	*****
*****	*****	*****
E8H FFH	00H Disattiva scrittura telemetria FFH Attiva scrittura telemetria	Attiva / Disattiva la scrittura su telemetria contemporanea alla scrittura su SD
E9H 00H	0000 card non formattata Bit 7:4 0001 formattata vuota Bit 7:4 0010 formattata non piena Bit 7:4 0011 formattata piena Bit 7:4 0100 errore di lettura Bit 7:4 0000 card non ammessa Bit 3:0 0001 capacità 128Mb Bit 3:0 0010 capacità 256Mb Bit 3:0 0011 capacità 512Mb Bit 3:0 0100 capacità 1Gb Bit 3:0 0101 reserved Bit 3:0 0110 reserved Bit 3:0 0111 reserved Bit 3:0	Info Card SD 1 Le informazioni relative ai 4 byte successivi a questo sono descritte dopo questa tabella set-up
EAH 00H	00H – FFH add. MSB	Puntatore successivo settore da scrivere
EBH 00H	00H – FFH add. CMSB	Puntatore successivo settore da scrivere
ECH 00H	00H – FFH add. LMSB	Puntatore successivo settore da scrivere
EDH 00H	00H – FFH add. LSB	Puntatore successivo settore da scrivere
EEH 00H	0000 card non formattata Bit 7:4 0001 formattata vuota Bit 7:4 0010 formattata non piena Bit 7:4 0011 formattata piena Bit 7:4 0100 errore di lettura Bit 7:4 0000 card non ammessa Bit 3:0 0001 capacità 128Mb Bit 3:0 0010 capacità 256Mb Bit 3:0 0011 capacità 512Mb Bit 3:0 0100 capacità 1Gb Bit 3:0 0101 reserved Bit 3:0 0110 reserved Bit 3:0 0111 reserved Bit 3:0	Info Card SD 2 Le informazioni relative ai 4 byte successivi a questo sono descritte dopo questa tabella set-up
EFH 00H	00H – FFH add. MSB	Puntatore successivo settore da scrivere
FOH 00H	00H – FFH add. CMSB	Puntatore successivo settore da scrivere
F1H 00H	00H – FFH add. LMSB	Puntatore successivo settore da scrivere
F2H 00H	00H – FFH add. LSB	Puntatore successivo settore da scrivere
F3H 00H	0000 card non formattata Bit 7:4 0001 formattata vuota Bit 7:4 0010 formattata non piena Bit 7:4 0011 formattata piena Bit 7:4 0100 errore di lettura Bit 7:4	Info Card SD 3 Le informazioni relative ai 4 byte successivi a questo sono descritte dopo questa tabella set-up

	0000 card non ammessa Bit 3:0 0001 capacità 128Mb Bit 3:0 0010 capacità 256Mb Bit 3:0 0011 capacità 512Mb Bit 3:0 0100 capacità 1Gb Bit 3:0 0101 reserved Bit 3:0 0110 reserved Bit 3:0 0111 reserved Bit 3:0	
F4H 00H	00H – FFH add. MSB	Puntatore successivo settore da scrivere
F5H 00H	00H – FFH add. CMSB	Puntatore successivo settore da scrivere
F6H 00H	00H – FFH add. LMSB	Puntatore successivo settore da scrivere
F7H 00H	00H – FFH add. LSB	Puntatore successivo settore da scrivere
F8H 00H	0000 card non formattata Bit 7:4 0001 formattata vuota Bit 7:4 0010 formattata non piena Bit 7:4 0011 formattata piena Bit 7:4 0100 errore di lettura Bit 7:4 0000 card non ammessa Bit 3:0 0001 capacità 128Mb Bit 3:0 0010 capacità 256Mb Bit 3:0 0011 capacità 512Mb Bit 3:0 0100 capacità 1Gb Bit 3:0 0101 reserved Bit 3:0 0110 reserved Bit 3:0 0111 reserved Bit 3:0	Info Card SD 4 Le informazioni relative ai 4 byte successivi a questo sono descritte dopo questa tabella set-up
F9H 00H	00H – FFH add. MSB	Puntatore successivo settore da scrivere
FAH 00H	00H – FFH add. CMSB	Puntatore successivo settore da scrivere
FBH 00H	00H – FFH add. LMSB	Puntatore successivo settore da scrivere
FCH 00H	00H – FFH add. LSB	Puntatore successivo settore da scrivere
FDH 0FH	Card SD 1 1 Forz. Formattazione Enable Bit 7 0 Forz. Formattazione Disable Bit 7 Card SD 2 1 Forz. Formattazione Enable Bit 6 0 Forz. Formattazione Disable Bit 6 Card SD 3 1 Forz. Formattazione Enable Bit 5 0 Forz. Formattazione Disable Bit 5 Card SD 4 1 Forz. Formattazione Enable Bit 4 0 Forz. Formattazione Disable Bit 4 Card SD 1 1 Scrittura pacchetti Enable Bit 3 0 Scrittura pacchetti Disable Bit 3 Card SD 2 1 Scrittura pacchetti Enable Bit 2 0 Scrittura pacchetti Disable Bit 2 Card SD 3 1 Scrittura pacchetti Enable Bit 1 0 Scrittura pacchetti Disable Bit 1 Card SD 4 1 Scrittura pacchetti Enable Bit 0 0 Scrittura pacchetti Disable Bit 0	Card SD da scrivere e selezione della forzatura di formattazione
FEH EEH	95H = 19200 CBH = 38400 DDH = 57600 EEH = 115200	Velocità seriale
FFH 00H	00H = no error 01H = velocità seriale insufficiente	La quantità totale di dati è eccessiva

02H = velocità seriale non ammessa	Ammesse 19200b/s,38400b/s,57600b/s,115200b/s
03H = word rate fir1/2 non ammessa	Word Rate fir1 >= Word Rate fir2
10H = hardware filter SPI fir1	Errore di scrittura su fir1
11H = parametri filter fir1	Parametro non permesso su fir1
12H = hardware coefficient SPI fir1	Errore di scrittura su fir1
13H = parametri coefficient fir1	Parametro non permesso su fir1
14H = hardware mclk SPI fir1	Errore di scrittura su fir1
15H = parametri mclk fir1	Parametro non permesso su fir1
16H = offset fir1	Parametro non permesso su fir1
17H = gain fir1	Parametro non permesso su fir1
18H = hardware attenuator spi fir1	Errore di scrittura su fir1
19H = parametri attenuator fir1	Parametro non permesso su fir1
1AH =hardware mux gain power-d fir1	Errore di scrittura su fir1
1FH = comando start fir1	Errore di scrittura su fir1
20H = hardware filter SPI fir2	Errore di scrittura su fir2
21H = parametri filter fir2	Parametro non permesso su fir2
22H = hardware coefficient SPI fir2	Errore di scrittura su fir2
23H = parametri coefficient fir2	Parametro non permesso su fir2
24H = hardware mclk SPI fir2	Errore di scrittura su fir2
25H = parametri mclk fir2	Parametro non permesso su fir2
26H = offset fir2	Parametro non permesso su fir2
27H = gain fir2	Parametro non permesso su fir2
28H = hardware attenuator spi fir2	Errore di scrittura su fir2
29H = parametri attenuator fir2	Parametro non permesso su fir2
2AH =hardware mux gain power-d fir2	Errore di scrittura su fir2
2FH = comando start fir2	Errore di scrittura su fir2
30H = lettura fir1	Errore di lettura su fir1
31H = lettura fir2	Errore di lettura su fir2
32H = hardware GPS select	Errore di scrittura su fir1
33H = parametri GPS select	Parametro non permesso su fir1
34H = hardware telemetria select	Errore di scrittura su fir1
35H = parametri telemetria select	Parametro non permesso su fir1
46H = carattere F stop applicativo	Uscita dall'applicativo con successo
FFH = trasferimento dati	Errore generale durante trasferimento dati setup

8. Appendice E: Setup di stazione

Di seguito è riportato il file di testo auto esplicativo di un setup di GAIASetup:

```
IP ADDRESS
10.100.0.84
VELOCITA' SERIALE
115200
N° CANALI SCHEDA BASE o EXT ATTIVA
4
CAMPIONAMENTO BASE o EXT ATTIVA
200
N° CANALI SCHEDA EXT PASSIVA
0
CAMPIONAMENTO EXT PASSIVA
50
NOME STAZIONE
ROM9
TIPO GPS
INTERNO
CH1 - ATTENUAZIONE
MAX
CH1 - GAIN
x2
CH1 - CODICE CANALE
HNZ
CH1 - FONDO SCALA
20000
CH2 - ATTENUAZIONE
MAX
CH2 - GAIN
x2
CH2 - CODICE CANALE
HNN
CH2 - FONDO SCALA
20000
CH3 - ATTENUAZIONE
MAX
CH3 - GAIN
x2
CH3 - CODICE CANALE
HNE
CH3 - FONDO SCALA
20000
CH4 - ATTENUAZIONE
MAX
CH4 - GAIN
x8
CH4 - CODICE CANALE
HNU
CH4 - FONDO SCALA
5000
CH5 - ATTENUAZIONE
MIN
CH5 - GAIN
x1
CH5 - CODICE CANALE
5HZ
CH5 - FONDO SCALA
16000
CH6 - ATTENUAZIONE
```


MIN
CH6 - GAIN
x1
CH6 - CODICE CANALE
6HN
CH6 - FONDO SCALA
16000
CH7 - ATTENUAZIONE
MIN
CH7 - GAIN
x1
CH7 - CODICE CANALE
7HE
CH7 - FONDO SCALA
16000
CH8 - ATTENUAZIONE
MIN
CH8 - GAIN
x1
CH8 - CODICE CANALE
8HZ
CH8 - FONDO SCALA
16000
PARAMETRI DALLA SCHEDA
EXT ATTIVA
CONFIGURAZIONE HARDWARE
EXT ATTIVA
TN IP ADDRESS
10.100.0.84
TN NETWORK
10.100.10.0
TN NETMASK
255.128.0.0
TN BROADCAST
10.100.255.255
TN GATEWAY
10.0.0.253
TN HOSTNAME
ROM9
TN PRIMARY IP
10.100.10.15
TN PRIMARY PORT
63003
TN SECONDARY IP
10.100.60.47
TN SECONDARY PORT
63003
TN PRIMARY TRANSMIT
Yes
TN SECONDARY TRANSMIT
No
SCHEDA ABILITATA PER LA TRASMISSIONE DEI PACCHETTI VIA IP
BASE
TIPO DI SUPPORTO REGISTRAZIONE LOCALE
COMPACTFLASH

9. Bibliografia

Dallaway, J. (eCosCentric), Garnett, N. (eCosCentric), Larmour, J. (eCosCentric), Lunn, A. (Ascom), Thomas, G. (MLB Associates), Veer, B. (eCosCentric), (1998, 1999, 2000, 2001, 2002, 2003, 2004) Red Hat, Inc. RedBoot User's Guide

Vincent Himpe, (2005) Visual Basic for Electronics Engineering Applications - Second Edition, 569 pp.

Mauro, D., Schmidt, K., (2005) Essential SNMP, 2nd Edition, 460 pp.

Pintore, S. e Salvaterra, L. (2007). Il Progetto TN-1. Rapporti Tecnici INGV, n° 40, 38 pp.

Ringraziamenti

Si ringrazia il dott. S. Pintore per l'assistenza sul protocollo SNMP.

Coordinamento editoriale e impaginazione

Centro Editoriale Nazionale | INGV

Progetto grafico e redazionale

Laboratorio Grafica e Immagini | INGV Roma

© 2010 INGV Istituto Nazionale di Geofisica e Vulcanologia

Via di Vigna Murata, 605

00143 Roma

Tel. +39 06518601 Fax +39 065041181

<http://www.ingv.it>



Istituto Nazionale di Geofisica e Vulcanologia