

Rapporti tecnici

INGV

**Sistemi di telecontrollo su rete
ethernet e GSM per stazioni sismiche**

89



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

SISTEMI DI TELECONTROLLO SU RETE ETHERNET E GSM PER STAZIONI SISMICHE

Ciro Buonocunto, Giovanni Scarpato, Giuseppe Borriello

¹INGV (Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Napoli - Osservatorio Vesuviano)

89

Indice

	Introduzione	5
1.	Sistema di telecontrollo su rete ethernet	5
1.1	Firmware di gestione	8
1.2	Interfaccia Utente	9
2.	Sistema di telecontrollo su rete GSM	10
2.1	Descrizione del sistema hardware	11
2.2	Utilizzo operativo	14
2.2.1	Monitoraggio della tensione di alimentazione	14
2.2.2	Reset dei sistemi	14
3.	Bibliografia	15
	Ringraziamenti	15
	Appendice 1 – Interfaccia grafica. Esempi di codice	16
	Appendice 2 – Caratteristiche tecniche del modulo Rabbit	18

Introduzione

La gestione e la manutenzione di sistemi di monitoraggio remoti richiede notevole dispendio di risorse umane.

La complessità dei sistemi utilizzati necessita di metodi di protezione hardware e software (*watchdog*) atti a limitare interferenze esterne quali sovratensioni sulle linee di segnale e di alimentazione o blocchi indesiderati.

Può talvolta accadere che per cause impreviste, quali ad esempio scariche atmosferiche a breve distanza, i sistemi elettronici si pongano in una condizione di blocco, pur senza subire danni strutturali significativi.

Tale situazione è spesso risolvibile con una procedura di spegnimento e successivo riavvio dei sistemi. Si comprende come tale semplice procedura richieda l'impiego di risorse nel caso di apparecchiature poste a grande distanza o in siti difficilmente raggiungibili in avverse condizioni meteorologiche, come i sistemi di monitoraggio della Rete sismica di Stromboli e alcune installazioni della Rete Sismica Vesuvio.

Per minimizzare la necessità di tali interventi di manutenzione è stato realizzato un sistema di controllo remoto, che consente operazioni di comando e verifica a distanza. È possibile controllare qualsiasi tipo di dispositivo tramite rete ethernet.

In alternativa a tale supporto è stato sviluppato un telecontrollo che utilizza la rete cellulare GSM.

Il primo sistema utilizza la rete ethernet e consente un reset completo dei sistemi ad esso collegati.

Anche il secondo permette di resettare i sistemi ad esso collegati, ma differisce dal primo sia per l'utilizzo della rete GSM che per la funzione che consente il monitoraggio h24 dell'alimentazione.

Non sempre è presente la rete ethernet o una sufficiente copertura del segnale GSM in corrispondenza dei siti presso i quali sono effettuate installazioni; per tale motivo sono stati sviluppati sistemi che sfruttano mezzi di trasmissione diversi.

Di seguito saranno descritti:

1. Sistema di telecontrollo su rete ethernet.
2. Sistema di telecontrollo su rete GSM.

1. Sistema di telecontrollo su rete ethernet (RECON_ETH)

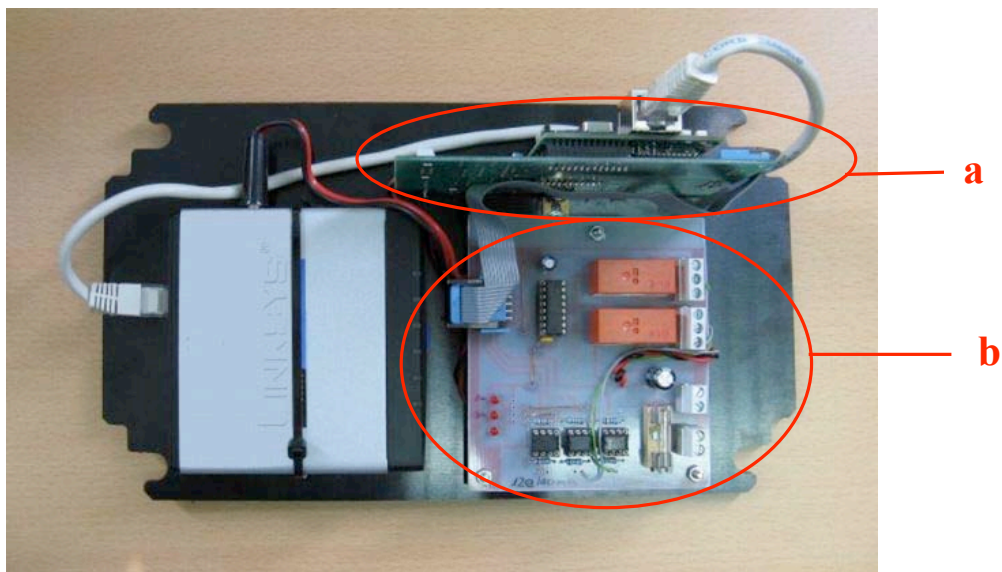


Figura 1. Il dispositivo RECON_ETH.

Il RECON_ETH è in grado di inviare comandi di reset ai sistemi ad esso collegati tramite la rete ethernet. È attualmente installato al Vesuvio presso un Array sismico multicanale (figura 2).

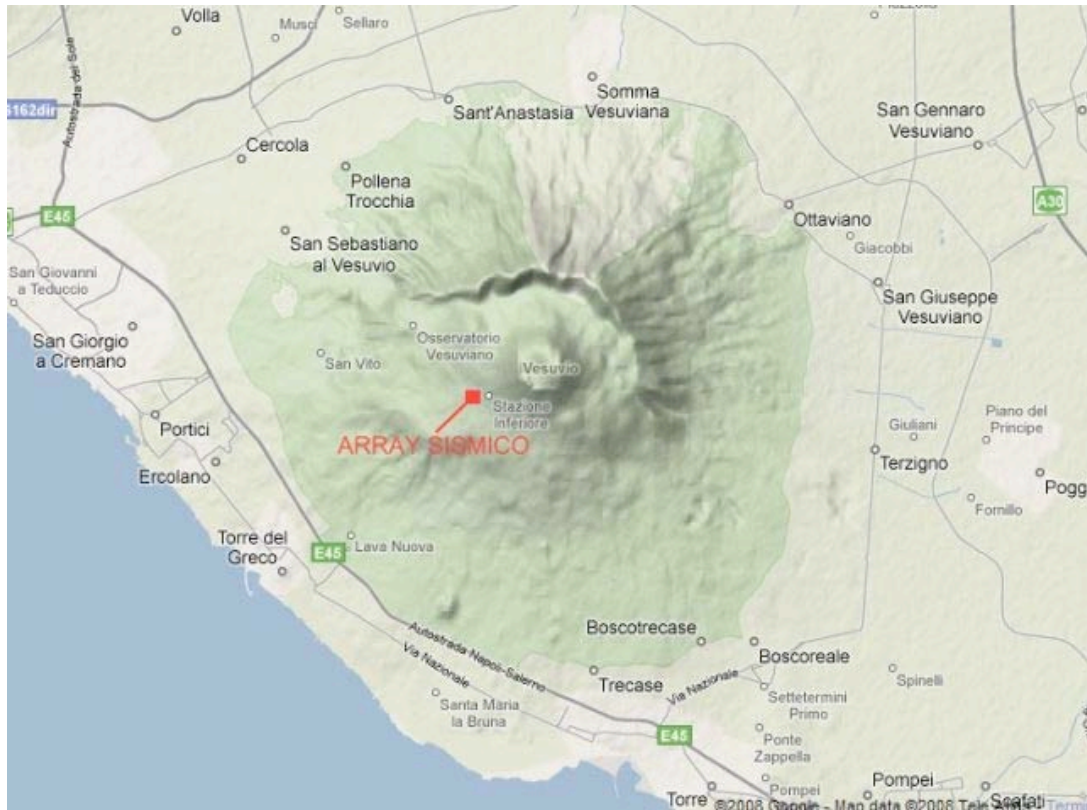


Figura 2. Mappa con ubicazione dell'installazione.

Il sistema di gestione utilizza un controller programmabile dotato di porta ethernet, per le connessioni con protocollo TCP/IP, ed una interfaccia di controllo con ingressi optoisolati per comandare le linee I/O . In dettaglio RECON_ETH è composto da tre sezioni logiche distinte (figura 1):

- scheda microcontrollore appositamente programmata (a)
- interfaccia di I/O che connette il controller ai dispositivi da gestire (b)
- interfaccia di gestione software per l'interconnessione.

Il controllo hardware è basato su una scheda con processore Rabbit serie 2000.

La scheda di interfaccia I/O (figura 3) è dotata di 3 ingressi optoisolati e di 2 uscite a relè in grado di pilotare carichi di 5A.

2 dei 3 ingressi sono utilizzati nella presente implementazione per ottenere feedback dell'attività delle uscite. Poiché il controller è alimentato a soli 5V, l'utilizzo di optoisolatori protegge la scheda da interazioni pericolose con le grandezze controllate. Le linee di input sono inviate al microcontrollore, utilizzando degli opto isolatori 4N35. Il consumo totale del sistema è di circa 120 mA a 12V.

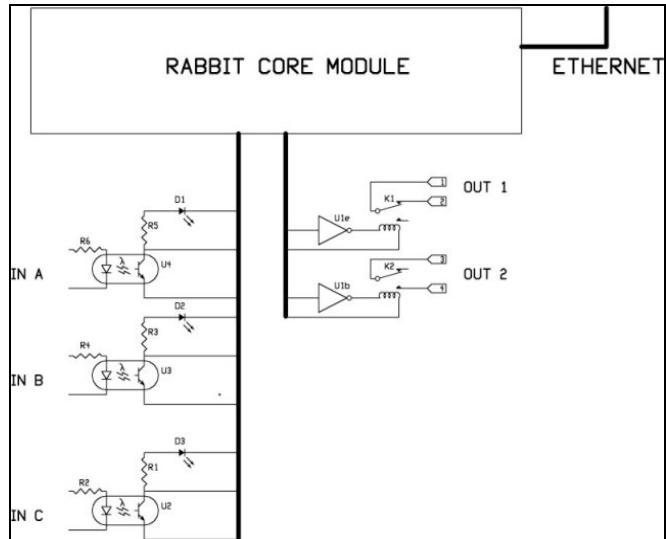


Figura 3. Schema dell'interfaccia I/O. IN A è una linea di riserva per future espansioni. IN B è connesso all'uscita 1 IN C è connesso all'uscita 2.

In tabella 1 è riportato l'elenco dei componenti utilizzati per la realizzazione dell'interfaccia di I/O.

R1, R2, R3	1K Ω
D1-D3	Led rossi 3 mm
K1-K2	Relè' 12V 1 scambio
R4, R5, R6	470 Ω
U1	ULN2003A
U2-U5	4N35

Tabella 1. Elenco dei componenti interfaccia I/O.

In figura 4 è riportato il circuito stampato dell'interfaccia di I/O.

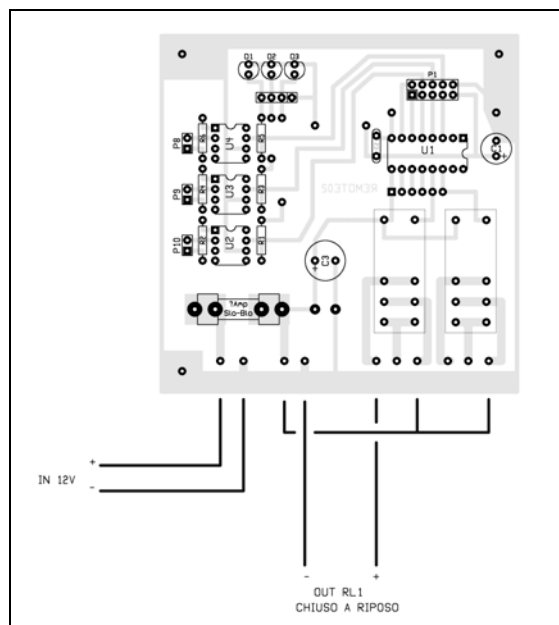


Figura 4. Circuito stampato dell'interfaccia I/O.

1.1 Firmware di gestione

Il programma eseguito dal microcontrollore si divide in due parti. La prima si occupa di inizializzare tutte le linee di I/O, i parametri necessari alla corretta configurazione dell'interfaccia ethernet e avviare il server, che resterà in attesa su un determinato *socket tcp* per accettare eventuali connessioni.

In figura 5 è illustrata una parte di codice, scritto in linguaggio C, necessario all'inizializzazione del socket su cui RECON_ETH accetterà le connessioni.

```
//inizializza il socket
sock_init();

while(1) {

    tcp_listen(&s, PORT, 0L, 0, NULL, 10);
    sock_wait_established(&s, 0, NULL, &status);
    //printf("Connessione in corso \n");
    sock_mode(&s, TCP_MODE_ASCII); //imposta la modalità ASCII
    if (sock_established(&s)) {
        while(sock_established(&s)) {
            sock_wait_input( &s , 0, NULL, &status);
            sock_gets(&s, buffer, 512);
            printf(">%s\n", buffer);
        }
    }
}
```

Figura 5. Esempio di codice.

La seconda parte si occupa della gestione delle richieste client. In particolare, accettata una connessione, vengono interpretate le richieste utente. Nel caso di richiesta di reset viene commutata un'apposita linea di I/O dallo stato in cui si trova a quello opposto per 60 secondi. In tal modo si ottiene la commutazione del relé da *normalmente chiuso* a *normalmente aperto* per 60 secondi, garantendo che il sistema si ponga in uno stato ben definito prima della riaccensione.

In figura 6 viene illustrato il diagramma di flusso, che descrive le azioni da compiere per effettuare un reset da remoto.

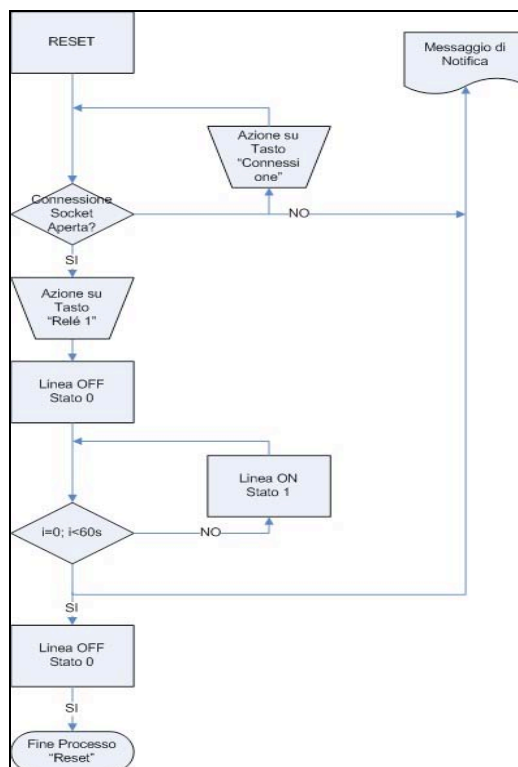


Figura 6. Diagramma di flusso dell'operazione di reset.

1.2 Interfaccia Utente

Per semplificare la gestione è stata realizzata un'interfaccia utente semplificata, che rende disponibili tutte le funzionalità del RECON_ETH (figura 7).

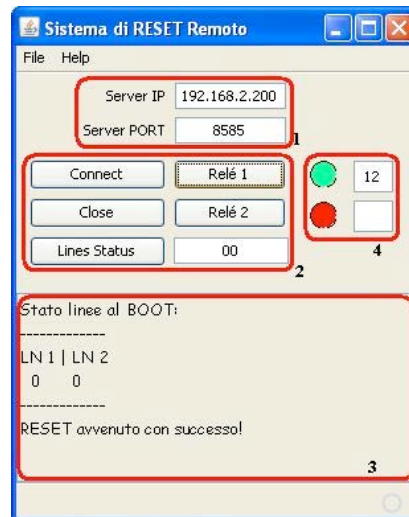


Figura 7. Interfaccia utente del RECON_ETH.

Essa è suddivisa in quattro sezioni logiche:

- sezione 1: consente l'inserimento delle informazioni utili alla connessione remota, quali indirizzo IP e porta socket su cui RECON_ETH accetta le connessioni. Tali connessioni sono protette tramite firewall di sistema.
- sezione 2: consente di interagire da remoto con RECON_ETH, eseguendo operazioni quali attivazione o chiusura della connessione, reset (agendo sui pulsanti “Relé 1” o “Relé 2”). Nel campo di testo della sezione 2 è visualizzato, in tempo reale, lo stato delle linee di I/O del rabbit durante la fase di reset.
- sezione 3: mostra le informazioni di servizio, quali lo stato della connessione, l'esito della procedura di reset o errori di connessione.
- sezione 4: mostra lo stato delle linee di controllo durante le operazioni di reset.

Il diagramma delle classi, relativo al funzionamento dell'interfaccia utente, è illustrato in figura 8.

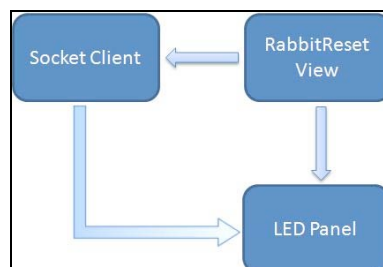


Figura 8. Diagramma delle classi.

L'applicazione è stata scritta in linguaggio Java, per renderla *platform-independent*.

Maggiori dettagli sul funzionamento, corredati da esempi di codice, si possono trovare nell'Appendice 1.

2. Sistema di telecontrollo su rete GSM (RECON_GSM)

Il RECON_GSM si propone lo scopo di controllare a distanza lo stato dell'alimentazione proveniente da pannelli solari, inviando, all'occorrenza, allarmi tramite SMS; può, inoltre, inviare comandi di reset ai sistemi ad esso collegati tramite semplici squilli provenienti da cellulari abilitati.

L'apparato è stato installato per la prima volta al vulcano Solfatara di Pozzuoli (Figure 9, 10 e 11) presso una stazione multiparametrica alimentata con pannelli fotovoltaici.

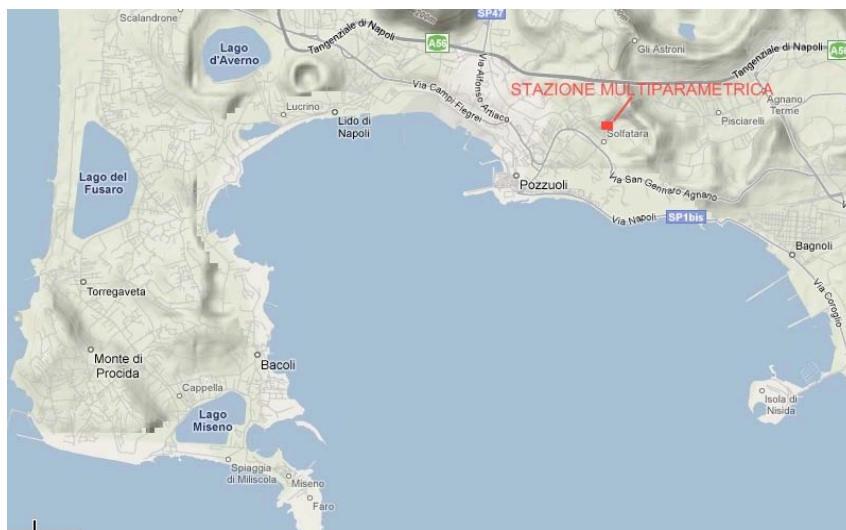


Figura 9. Mappa con ubicazione dell'installazione.

Il sistema è configurato per inviare un SMS ai numeri abilitati in caso di interruzione dell'alimentazione fornita dal sistema di pannelli fotovoltaici ed un ulteriore SMS al ripristino delle condizioni di alimentazione ottimale. È possibile impostare soglie di controllo di tensione tali da poter intervenire prima della totale scarica delle batterie, garantendo, così, la continuità del monitoraggio.

Presso questa installazione, allo scopo di migliorare la ricezione del segnale GSM, è stata usata un'antenna esterna a pannello da 9 dBi.



Figura 10. Installazione presso il vulcano Solfatara



Figura 11. Installazione presso il vulcano Solfatara (particolare).

2.1 Descrizione del sistema hardware

Il sistema è composto da due parti (figura 12):

- a. Modulo di controllo alimentazione e relativi allarmi.
- b. Unità di telecontrollo su rete GSM.

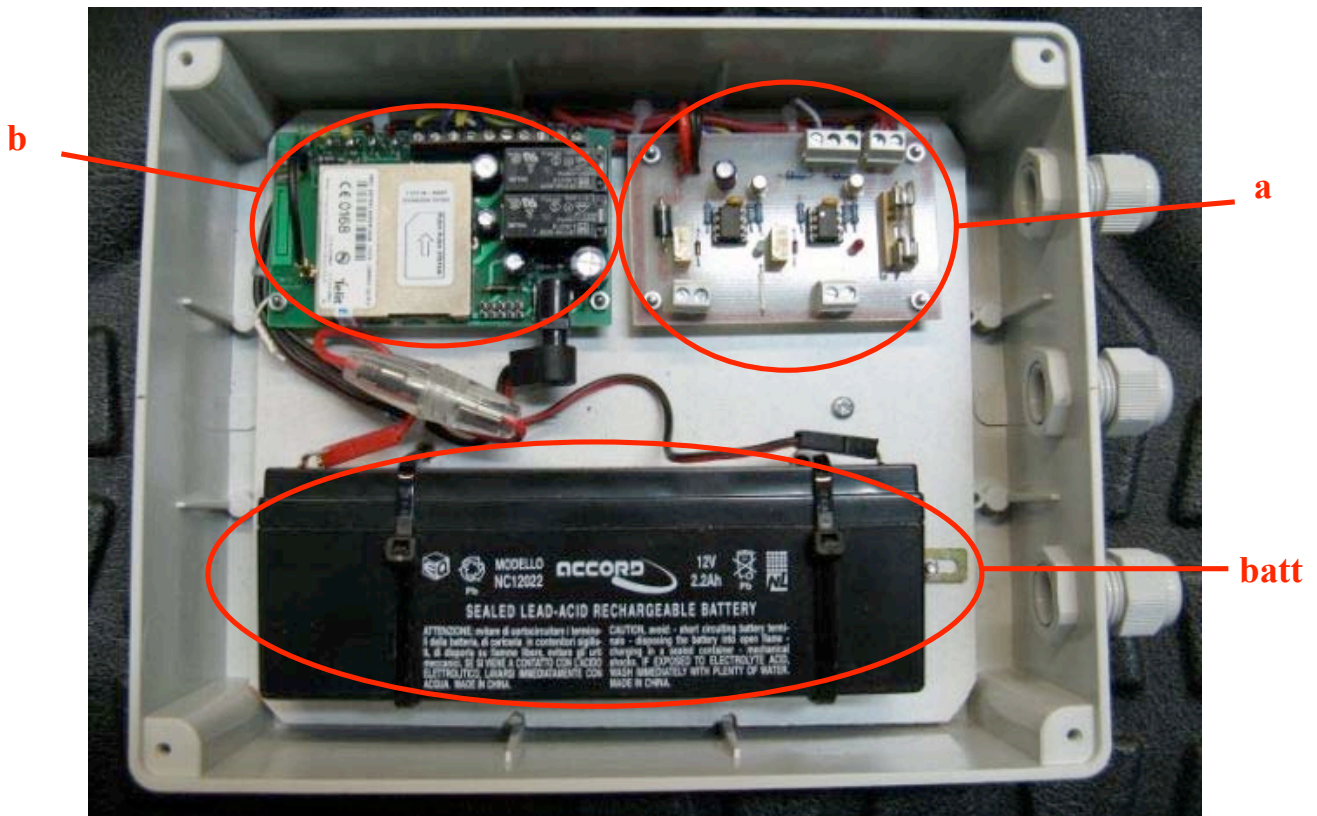


Figura 12. Le diverse parti del sistema.

- a. Il modulo di controllo è composto da due comparatori di tensione che consentono di monitorare lo stato di efficienza del sistema di alimentazione fotovoltaica. È gestita una doppia soglia, di cui una di attenzione (inizio scarica batteria) e una di allarme (interruzione alimentazione). Il modulo è fornito di un sistema di batteria tampone (batt) che consente di ricevere od inviare comandi anche in assenza dell' alimentazione principale per almeno 48 ore. Il consumo totale medio del sistema è di circa 50 mA a 12V.

In figura 13 è rappresentato lo schema elettrico del modulo “a” e le connessioni con il modulo GSM. In tabella 2 sono elencati i rispettivi componenti.

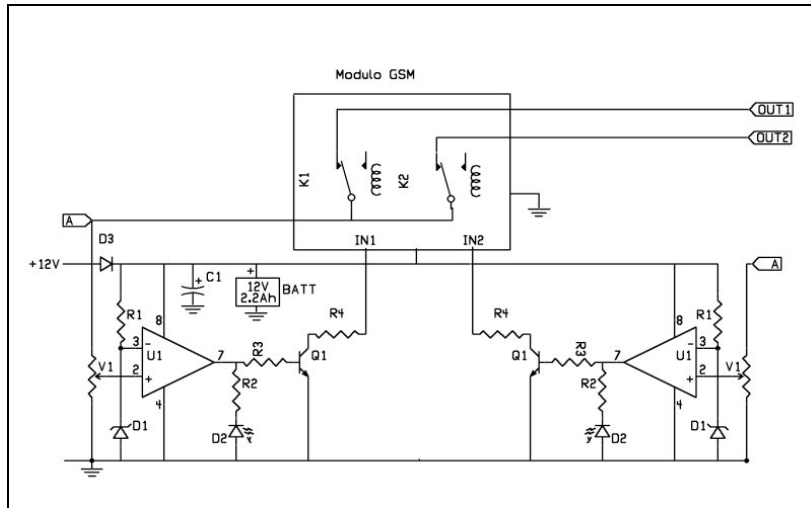


Figura 13. Schema elettrico del modulo “a”.

R1	10 KOhm
R2	1 KOhm
R3	4.7 KOhm
R4	1KOhm
V1	47 KOhm trimmer multigiri
C1	100 μ F 25V
D1	5V Zener
D2	LED
D3	1N5401
Q1	BC109
BATT	12V 2.2 Ah
U1	LM311

Tabella 2. Elenco componenti modulo “a”.

b. Unità telecontrollo GSM (figura 14):



Figura 14. Il modulo TDG33.

È stato utilizzato un modulo commerciale GSM/GPRS della Futura Elettronica denominato TDG33 (figura 14). In tabella 3 sono riassunte le principali caratteristiche tecniche. Si tratta di un telecomando bidirezionale, con il quale è possibile attivare da remoto due relé (in modalità bistabile o monostabile) mediante squilli o appositi SMS di comando (completi di password) inviati dai telefoni cellulari appositamente abilitati. È possibile abilitare fino ad un massimo di 8 numeri telefonici/amministratore in grado di configurare il dispositivo ed ai quali il dispositivo invia SMS e/o telefonate in base alla configurazione effettuata.

Modulo GSM/GPRS	Telit GM862-Quad, E-GSM 850-900 MHz, DCS 1800-1900 MHz
Potenza di uscita:	Class 4 (2 W @ 850-900 MHz) Class 1 (1 W @ 1800-1900 MHz)
Antenna	Antenna GSM integrata
Alimentazione	5 ÷ 32 Vdc
Corrente assorbita	20 mA a riposo, 500 mA nei picchi
Uscite a relé	2, per controllo di carichi a bassa tensione tipo SELV (< 60 Vdc)
Corrente massima contatti relé	10 A
Ingressi digitali	2 (livello 1 = 5 ÷ 32 Vdc; livello 0 = 0 Vdc)

Tabella 3. Principali caratteristiche tecniche del modulo TDG33.

È possibile, inoltre, inserire fino a 200 numeri/utente in grado di inviare i soli comandi di reset. Tali numeri non sono in grado di effettuare modifiche alla configurazione.

Il dispositivo necessita di una SIM Card voce attiva fornita da un qualsiasi gestore che utilizzi le reti GSM 900/1800 MHz.

È dotato di 2 ingressi digitali e di 2 uscite a relé da 10A .

Le due uscite a relé possono essere configurate come segue:

Bistabile : un SMS comanda l'attivazione e un successivo comando SMS comanda la disattivazione.

Monostabile : un singolo comando attiva l'uscita per un tempo fissato (comando attivato da uno squillo).

Per configurare l'unità GSM è necessario inviare SMS opportunamente formattati, contenenti i comandi di configurazione.

Nella tabella 4 è riportata la sintassi dei principali comandi che è possibile inviare al modulo GSM; per ulteriori informazioni si rinvia al manuale di servizio (http://www.futuraelettronica.net/pdf_ita/8100-TDG33.pdf).

FUNZIONE	COMANDO SMS	PASSWORD NECESSARIA
Cambio password	PWDxxxxx	Sì
Memorizzare un numero (massimo 8 numeri) (max 19 caratteri per numero)	x È LA POSIZIONE NUMx+393359999999	SOLO SE LA POSIZIONE È OCCUPATA
Cancellare un numero	NUMx	Sì
Verificare i numeri memorizzati	NUM?	Sì
Impostare i numeri cui verranno inviati gli SMS	SMSxxxxxxxx:ON	Sì
Impostare i numeri cui verrà fatto uno squillo	VOCxxxxxxxx:ON	Sì

Tabella 4. Principali comandi di configurazione del modulo TDG33.

2.2 Utilizzo operativo

Come già accennato il sistema è utilizzato essenzialmente per due scopi:

- Monitoraggio della tensione di alimentazione.
- Reset dei sistemi.

2.2.1 Monitoraggio della tensione di alimentazione

Se la tensione di alimentazione del sistema scende al di sotto di 13V (soglia di attenzione) le batterie passano nella condizione di scarica indicando un insufficiente apporto energetico da parte del sistema fotovoltaico di ricarica. In seguito a questa condizione il sistema invia un SMS di allarme. Al ripristino dell'alimentazione il sistema invierà un secondo SMS.

Se permane il problema di alimentazione, quando il valore della tensione scende sotto i 12V (soglia inferiore), ulteriori SMS comunicheranno lo stato di batteria scarica.

L'eventuale ripristino dell'alimentazione sarà segnalata, anche in questo caso da un SMS. Anche in caso di alimentazione insufficiente il telecontrollo è in condizioni operative grazie ad una batteria tampone che consente almeno 48 ore di autonomia.

2.2.2 Reset dei sistemi

Un router bloccato o un computer in errore, sono solo due dei possibili casi che richiedono un reset.

In questi casi basterà effettuare una telefonata attraverso un cellulare abilitato e il modulo GSM, dopo due o tre squilli, chiuderà la comunicazione e attiverà il relè 1 configurato in modalità monostabile, che sarà eccitato per un tempo di 60 secondi. Si è scelto tale intervallo di tempo per assicurare che ogni tensione interna dei circuiti controllati raggiunga il valore zero. In tale modo è possibile ottenere il riavvio del sistema in totale sicurezza. Per ottimizzare affidabilità e consumi il relè è in configurazione di scambio normalmente chiuso a riposo. Ciò significa che si attiverà solo per il tempo minimo necessario ad effettuare la procedura di reset, riposizionandosi in condizioni di riposo alla fine del comando.

Si è scelto di utilizzare il modo di comando monostabile per una serie di considerazioni:

- è possibile comandare la funzione monostabile sulla linea 1 inviando un semplice squillo da cellulari abilitati a costo zero;
- eventuali problemi sulla rete GSM non pongono il sistema in uno stato di blocco in quanto, al termine dell'esecuzione del comando, il relè torna nello stato precedente.

Su entrambi i sistemi si possono facilmente implementare funzioni aggiuntive. Sul RECON_GSM , ad esempio, il relè 2 non è attualmente utilizzato. Esso può essere impostato in modalità bistabile per effettuare calibrations di una stazione sismica implementando un circuito per la calibrazione automatica realizzato da M. Capello [2001] o settare parametri di configurazione (es. il gain).

Il RECON_ETH può sfruttare l'ingresso denominato IN A (figura 3) per ulteriori controlli di tipo on/off.

3. Bibliografia

Capello M., (2001). *Calibrazione automatica di stazioni sismiche dotate di sensori a corto periodo*. Open-File Report 3 – 2001, INGV – sezione di Napoli, “Osservatorio Vesuviano”

Futura Elettronica s.r.l. <http://www.futuraelettronica.net>

Fairchild semiconductor. <http://www.fairchildsemi.com>

Telit. <http://www.telit.com>

Calero TLC. <http://www.calearotlc.com>

ST Microelectronics. <http://www.st.com>

Rabbit Semiconductor. <http://www.rabbit.com>

Ringraziamenti

Si ringraziano Guglielmo e Maurizio Clery della ditta V.I.P. Elettronica di Napoli, per la competenza e la professionalità dimostrate.

Appendice 1 – Interfaccia grafica. Esempi di codice

L'interfaccia grafica è stata sviluppata in linguaggio Java, basandosi su un'architettura a strati (o livelli).

Il livello più alto, in particolare, quello prossimo all'utente, si occupa di gestirne le azioni.

Esso controlla lo stato delle linee di I/O del rabbit o ne cambia lo stato, consentendo il riavvio della periferica ad esso connessa.

La classe *ResetRabbitView*, quella di alto livello (figura 15), è la principale dell'architettura. Essa genera l'interfaccia grafica accessibile all'utente, interpretandone le azioni e comunicandole ai moduli dei livelli inferiori.

```
ResetRabbitView
+led1 : LedPanel
+led2 : LedPanel
-jButton1 : JButton
-jButton2 : JButton
-jButton3 : JButton
-jButton4 : JButton
-jButton5 : JButton
-jLabel1 : JLabel
-jLabel2 : JLabel
-jPanel1 : JPanel
-jPanel2 : JPanel
-jPanel3 : JPanel
-jScrollPane1 : JScrollPane
-jTextArea1 : JTextArea
-jTextField1 : JTextField
-jTextField2 : JTextField
-jTextField3 : JTextField
-jTextField4 : JTextField
-jTextField5 : JTextField
-mainPanel : JPanel
-menuBar : JMenuBar
-progressBar : JProgressBar
-statusAnimationLabel : JLabel
-statusMessageLabel : JLabel
-statusPanel : JPanel
-messageTimer : Timer
-busyIconTimer : Timer
-idleIcon : Icon
-busyIcons : Icon[] = new Icon[15]
-busyIconIndex : int = 0
+socket : SocketClient = null
-aboutBox : JDialog
~messageTimer2 : Timer

+ResetRabbitView(app : SingleFrameApplication)
+showAboutBox() : void
-initComponents() : void
-jButton1ActionPerformed(evt : ActionEvent) : void
-jButton2ActionPerformed(evt : ActionEvent) : void
-jButton4ActionPerformed(evt : ActionEvent) : void
-jButton3ActionPerformed(evt : ActionEvent) : void
-jButton5ActionPerformed(evt : ActionEvent) : void
-exitMenuItemActionPerformed(evt : ActionEvent) : void
~inizializzazione() : void
~chiusura() : void
```

Figura 15. La classe “ResetRabbitView”.

La classe *SocketClient* (figura 16), del package *resetrabbit*, offre servizi di connettività al modulo di livello superiore. In particolare, svolge le seguenti azioni:

- instaura una connessione socket con il modulo rabbit;
- mantiene attiva la connessione per il tempo necessario alle operazioni di reset;
- risponde alle possibili azioni dell'utente, come ad esempio effettuare un riavvio di un oggetto alimentato da RECON_ETH;
- chiude la connessione.

Tali operazioni sono implementate rispettivamente tramite i metodi *SocketClient()*, *openSocket()*, *setOFF()*, *setON()*, *closeSocket()*. I metodi *check(String)*, e *checkStatus(String)* sono usati per controllare lo stato delle linee del modulo rabbit, in modo da avere un quadro descrittivo dello stato di RECON_ETH.

SocketClient
<pre> ~in : InputStream = null ~out : PrintStream = null +client : Socket = null ~IPSERVER : String = "192.168.10.102" ~CMD : String ~PORT : int = 1007 +connected : boolean +result_global : BufferedReader = null ~err : String = null ~runner : Thread = null +test : Timer = null ~NOTIFY : JTextArea = null ~LED1 : LedPanel ~LED2 : LedPanel ~FIELD1 : JTextField ~FIELD2 : JTextField ~FIELD3 : JTextField ~COUNTER : int = 0 +SocketClient(ip : String, port : int) +Watch(cmd : String) : String +setON(cmd : String) : void +setOFF(cmd : String) : String +setON(cmd : String, debug : boolean) : void +setOFF(cmd : String, debug : boolean) : void +check(cmd : String) : String +checkStatus(cmd : String) : String +openSocket(JTEXTAREA : JTextArea) : void +openSocket(JTEXTAREA : JTextArea, pallina1 : LedPanel, pallina2 : LedPanel) : void +openSocket(JTEXTAREA : JTextArea, pallina1 : LedPanel, pallina2 : LedPanel, field1 : JTextField, field2 : JTextField) : void +openSocket(JTEXTAREA : JTextArea, pallina1 : LedPanel, pallina2 : LedPanel, field1 : JTextField, field2 : JTextField, field3 : JTextField) : void +closeSocket() : void +actionPerformed(e : ActionEvent) : void ~stopTimer() : void </pre>

Figura 16. La classe “Reset Rabbit view”.

Infine la classe *LedPanel*, illustrata in figura 17, genera un pannello contenente led grafici, grazie ai quali si ottiene un riscontro grafico, in tempo reale, delle operazioni che RECON_ETH sta svolgendo in seguito a richieste utente.

LedPanel
<pre> ~X : int ~Y : int ~W : int ~H : int ~NLED : int ~BACKGROUND : Color ~BACKGROUNDnew : Color ~STATUS : String ~FONT : Font ~LAELED : boolean +LedPanel() +LedPanel(x : int, y : int, width : int, height : int, background : Color, labeled : boolean) +LedPanel(x : int, y : int, width : int, height : int, background : Color, nled : int, labeled : boolean) +LedPanel(x : int, y : int, width : int, height : int, background : Color, nled : int, LABEL : String, labeled : boolean) +setStatus(status : String, font : Font) : void +setBackground(sfondo : Color) : void +setBackground(background : Color, backgroundnew : Color) : void +setLed() : void +setLed(text : String) : void +paintComponent(comp : Graphics) : void </pre>

Figura 17. La classe “Led panel”.

Appendice 2 – Caratteristiche tecniche del modulo Rabbit (www.rabbit.com)

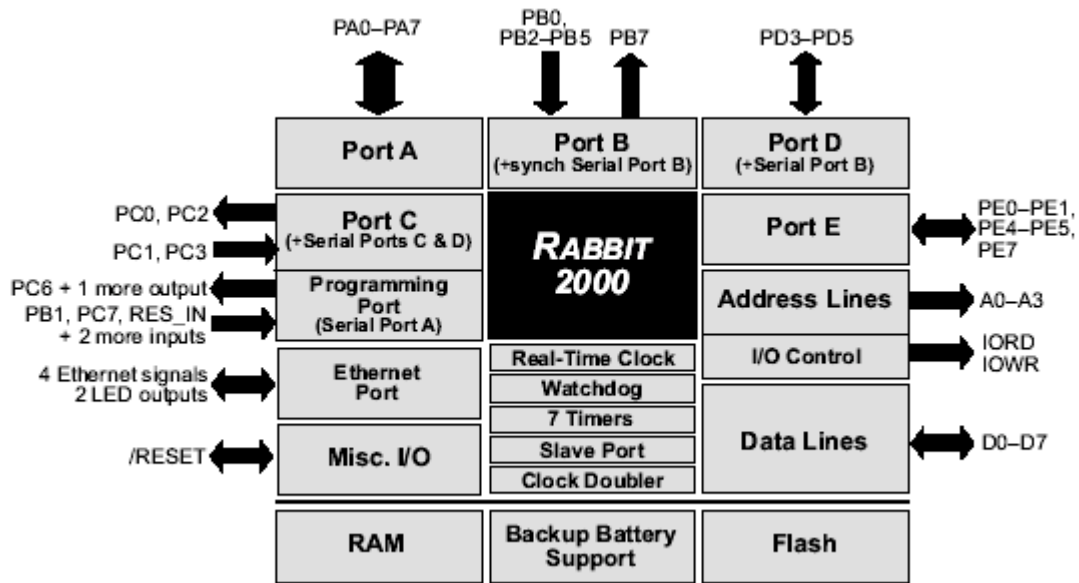


Figura 18. Schema del modulo Rabbit 2000.

- Dimensioni: 1.60" × 2.30" × 0.86"
- Microprocessore: Rabbit 2000 running at 22.1 MHz
- 26 parallel I/O lines: 16 configurable for input or output, 7 fixed inputs, 3 fixed outputs
- 8 data lines (D0–D7)
- 4 address lines (A0–A3)
- Memory I/O read, write
- External reset input
- Five 8-bit timers (cascadable in pairs) and two 10-bit timers
- 256K–512K flash memory, 128K–512K SRAM
- Real-time clock
- Watchdog supervisor
- Provision for customer-supplied backup battery via connections on header J5
- 10Base-T RJ-45 Ethernet port
- Raw Ethernet and two associated LED control signals available on 26-pin header
- Three CMOS-compatible serial ports: maximum asynchronous baud rate of 691,200 bps, maximum synchronous baud rate of 5,529,600 bps. One port is configurable as a clocked port.
- Six additional I/O lines are located on the programming port, can be used as I/O lines when the programming port is not being used for programming or in-circuit debugging
- one synchronous serial port can also be used as two general CMOS inputs and

Coordinamento editoriale e impaginazione

Centro Editoriale Nazionale | INGV

Progetto grafico e redazionale

Laboratorio Grafica e Immagini | INGV Roma

© 2008 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