

Rapporti tecnici

INGV

Helicorder digitale

145



Istituto Nazionale di
Geofisica e Vulcanologia

Direttore

Enzo Boschi

Editorial Board

Raffaele Azzaro (CT)

Sara Barsotti (PI)

Mario Castellano (NA)

Viviana Castelli (BO)

Rosa Anna Corsaro (CT)

Luigi Cucci (RM1)

Mauro Di Vito (NA)

Marcello Liotta (PA)

Simona Masina (BO)

Mario Mattia (CT)

Nicola Pagliuca (RM1)

Umberto Sciacca (RM1)

Salvatore Stramondo (CNT)

Andrea Tertulliani - Editor in Chief (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

HELICORDER DIGITALE

Catello Acerra, Sandro Rao, Carlo Salvaterra

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

145

Indice

Introduzione.....	5
1. Descrizione del progetto.....	6
2. Sviluppi futuri.....	11
Appendice A - Programma assembler HELIDAC.ASM.....	13
Appendice B - Struttura dati del pacchetto trasmesso dalla stazione digitale.....	24
Appendice C - Programma in linguaggio ABEL implementato nella GAL22V10.....	26
Bibliografia.....	27

Introduzione

L'helicorder digitale è uno strumento d'interfaccia che offre la possibilità di registrare su un classico supporto cartaceo termo-sensibile le forme d'onda di segnali sismici provenienti da un sistema di acquisizione di nuova generazione usato nelle attuali stazioni sismiche.

Le nuove tecnologie elettroniche hanno infatti portato a un radicale cambiamento della tipologia della nuova Rete Sismica Nazionale e delle stazioni sismiche remote che ne fanno parte; nel giro di pochi anni le vecchie stazioni analogiche sono state affiancate, e poi completamente sostituite, da apparecchiature di tipo digitale, cambiando radicalmente il trattamento del segnale sismico.

Tale evoluzione ha chiaramente influito sul trattamento e sulla trasmissione del segnale sismico verso la sala di monitoraggio dell'INGV di Roma. Il segnale sismico non è più trasmesso dalla stazione remota attraverso un processo di modulazione in frequenza e demodulazione in sede di un segnale elettrico, ma viaggia sotto forma di serie numerica (ottenute da un processo di digitalizzazione locale), organizzata in opportuni pacchetti di dati, spedita dalle stazioni remote in continuo ed acquisiti in sede direttamente dai calcolatori.

Le vecchie linee di comunicazione a trasmissione analogica CDA (Collegamento Diretto Analogico) sono state sostituite da linee a trasmissione numerica CDN (Collegamento Diretto Numerico). Oggi la maggior parte dei dati sismici viaggiano dalle stazioni remote verso la sala centrale di monitoraggio su rete internet.

Presso le sale di monitoraggio dell'INGV sono ancora diffusi tradizionali sistemi di registrazione su carta termo-sensibile avvolta in un rullo rotante [5], dove la traccia del segnale sismico è ottenuta dal movimento di un ago (pennino) che riscaldato elettricamente lascia una traccia del segnale (helicorder); per ragioni costruttive questi dispositivi accettano solo un segnale d'ingresso di tipo analogico.

In questa nota viene presentato il progetto e la realizzazione di una interfaccia digitale-analogica in grado di ricostruire la forma d'onda del segnale sismico, completo di impulso marcato tempo, da applicare all'helicorder a partire dal flusso di dati numerici. Un display LCD visualizza alcune importanti informazioni, come l'orario associato alla traccia sismica e lo stato di funzionamento del flusso dati. L'informazione temporale è di fondamentale importanza dal momento che i pacchetti di dati possono subire ritardi dovuti ai tipici processi di instradamento sulla rete (non sempre uguali per tutti i pacchetti); con le vecchie linee analogiche l'assegnazione temporale sui segnali sismici avveniva al loro arrivo in sala di monitoraggio (tramite un GPS e/o orologio atomico), e i segnali provenienti da stazioni remote poste a diverse distanze dalla sede, si propagavano ad una velocità comunque tale da rendere trascurabile l'errore di associazione temporale ai fini della localizzazione del terremoto. Le nuove stazioni sono dotate di ricevitore GPS e il riferimento temporale, inserito nel pacchetto dati direttamente in loco e al momento della formazione del dato dal spedire, diventa un elemento di estrema precisione ai fini della localizzazione di un terremoto.

1. Descrizione del progetto

Il funzionamento della scheda di interfaccia si basa sull'impiego del microprocessore Dallas 80C320 [4], simile all'8051 della Intel, del DAC (*Digital-to-Analog Converter*) AD569 [2], della Analog Device e di altri componenti periferici collegati come da schema elettrico di figura 1 e figura 2: una RAM esterna, una logica programmata GAL [3] per la codifica degli indirizzi, un buffer 74573 per lo split del *bus* dati e indirizzi e, infine, una *flash-eprom* per gli aggiornamenti del *firmware* e la diagnostica dei vari registri del processore attraverso un programma residente (DIAG320).

L'80C320 ha due porte seriali: una utilizzata per acquisire i dati sismici (*seriale1*) a 19200 bit/s e l'altra (*seriale0*) per colloquiare con il sistema in caso di malfunzionamento o per trasferire eventuali aggiornamenti *firmware*. Il ciclo di operazioni eseguite dal processore sono implementate mediante codice *assembler* (programma HELIDAC.ASM in appendice A) e schematizzate nel diagramma di flusso di figura 3.

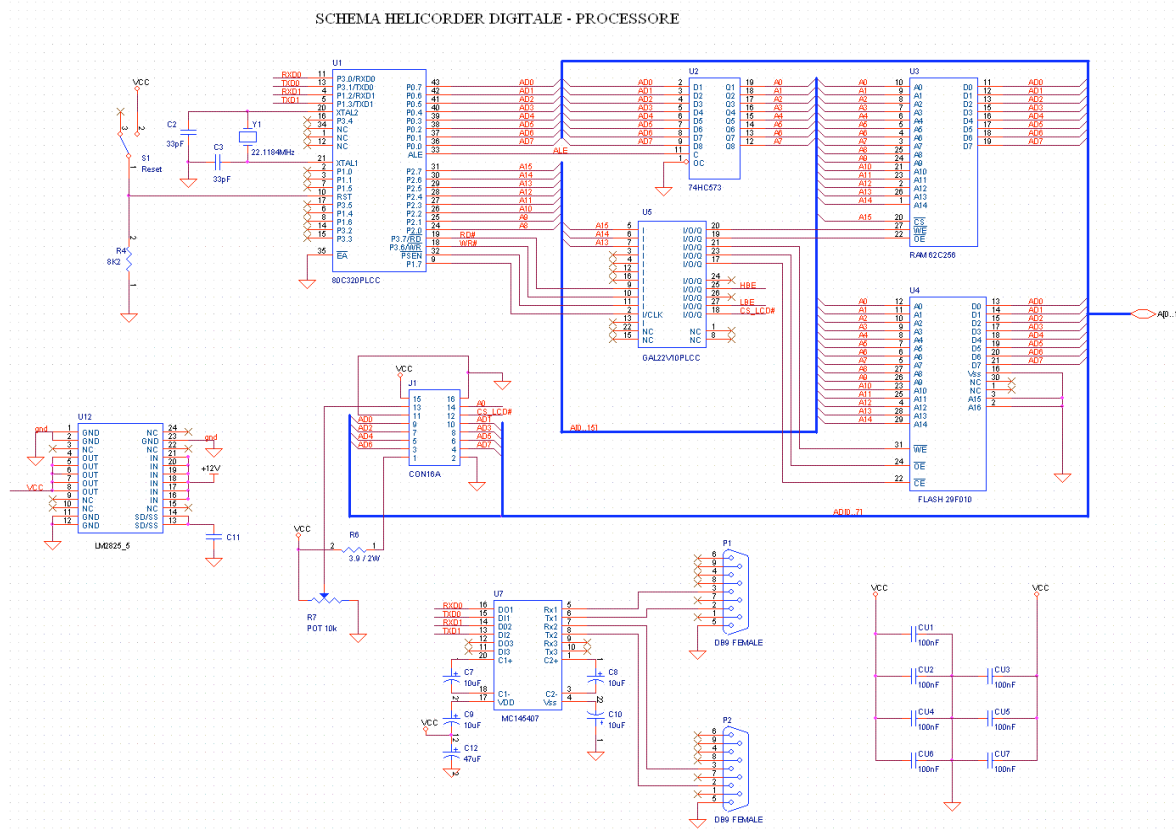


Figura 1. Schema elettrico dell'Helicorder digitale (stadio processore).

I dati numerici dei segnali sismici (ottenuti con campionamento continuo a 50Hz) arrivano sulla porta *seriale1* sottoforma di pacchetti (organizzati come mostrato in appendice B) e sono etichettati da un *header* proprietario [1] che contiene una sequenza fissa di 8 caratteri esadecimali (primi 8 byte di inizio di ogni pacchetto), l'orario in cui è avvenuto il campionamento e il nome della stazione sismica di provenienza. Il processore al suo avvio inizializza i vari registri ed attiva il *watch-dog*; questo permette il riavvio del sistema dopo circa 2 secondi di inattività. Successivamente il processore controlla se sulla porta *seriale0* è presente un carattere inviato dall'utente qualora si voglia effettuare un controllo sul funzionamento dell'interfaccia; se la *seriale0* riceve un carattere "F" (codice ascii 46h) il programma si interrompe, passando in modalità diagnostica trasferendo il controllo al *software* DIAG320 presente nella *flash-eprom*; se il carattere è una "B" (codice ascii 42h) allora il processore invia al DAC un segnale di prova (rampa di 2 Hz); infine, se viene

inviato un carattere “A” (codice ascii 41h) o nessun carattere è presente sulla porta *seriale0* il processore prosegue in modalità normale leggendo i dati ricevuti sulla *seriale1*.

Alla ricezione dei dati il processore controlla i primi dei 4 *byte* d'apertura dell'*header* ed aggancia il primo pacchetto dati disponibile, quindi estrae l'ora, il nome della stazione remota e la serie numerica dei dati sismici.

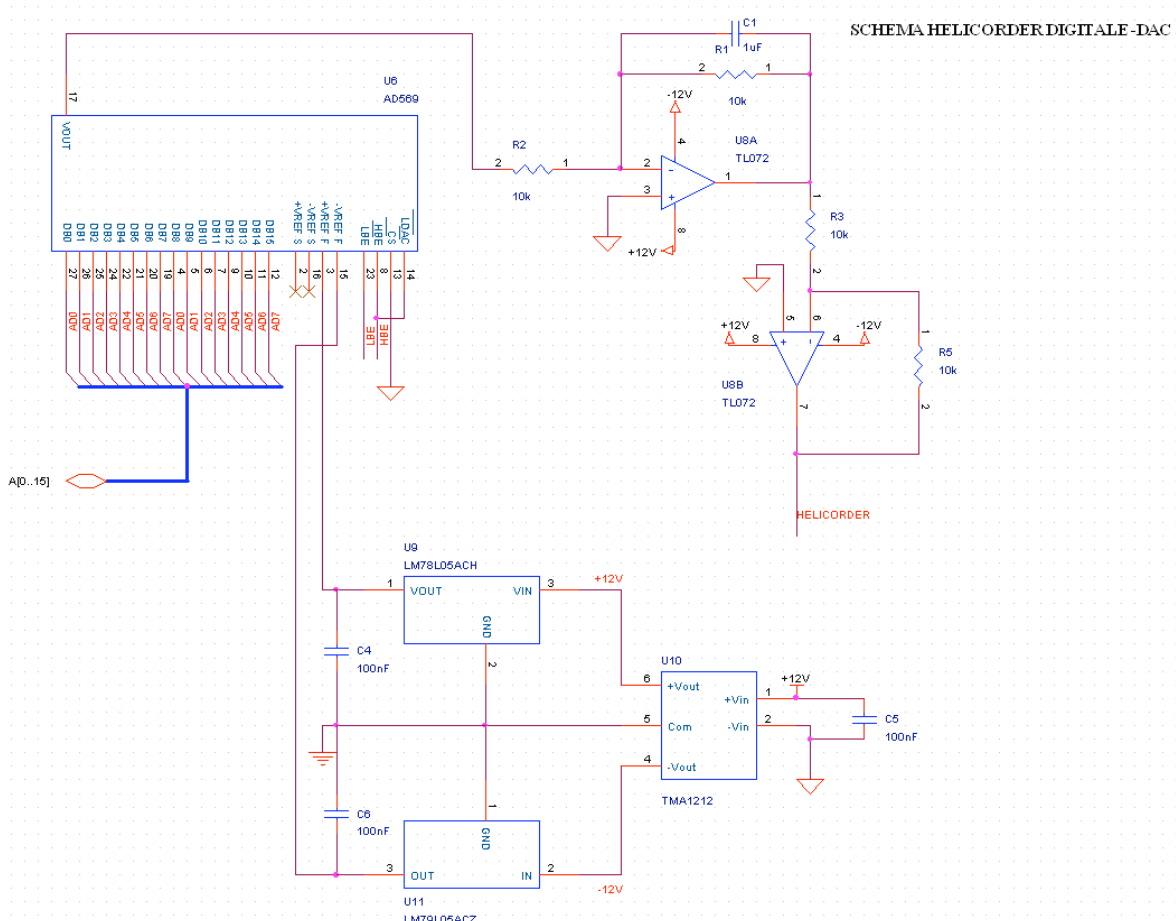


Figura 2. Schema elettrico dell'Helicorder digitale (stadio DAC).

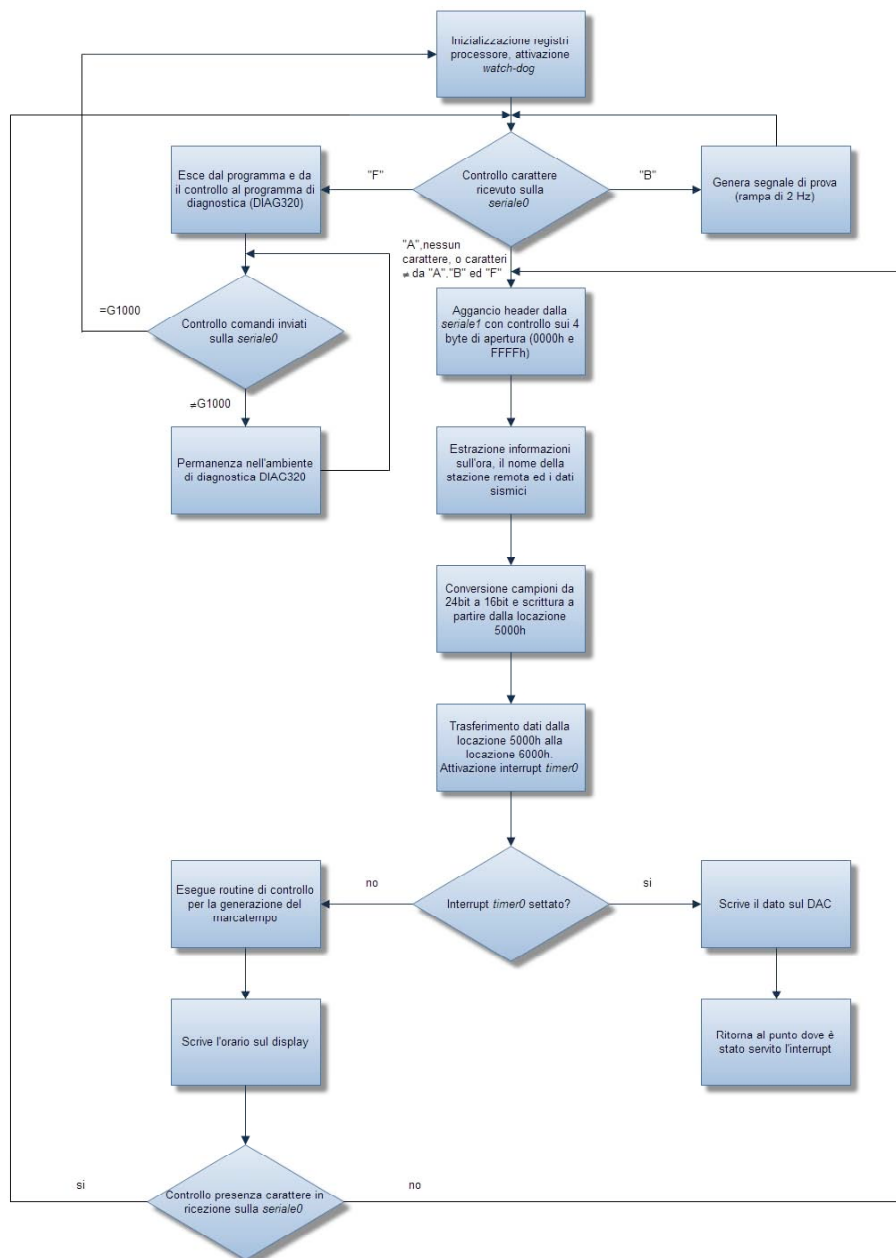


Figura 3. Diagramma di flusso del programma in codice assembler HELIDAC.ASM.

Per la conversione digitale analogica è stato utilizzato un DAC AD569 a 16 bit ed è stato necessario introdurre una procedura di programma che converte i dati codificati a 24 bit e complemento a 2, come in figura 4, in dati codificati a 16 bit secondo il formato di figura 5:

<u>formato dati ricevuti in complemento a due</u>				
	MSB	CSB	LSB	
+Vmax	01111111	11111111	11111111	(7FFFFFFH)

	00000000	00000000	00000001	(000001h)
0V	00000000	00000000	00000000	(000000h)
	11111111	11111111	11111111	(FFFFFFH)

-Vmax	10000000	00000000	00000000	(800000H)

Figura 4. Formato dei dati ricevuti con codifica a 24bit in complemento a due.

<u>formato dati normalizzato da scrivere nel DAC</u>			
	MSB	LSB	
Vmax	11111111	11111111	(FFFFH)

	10000000	00000001	(8001H)
0V	10000000	00000000	(8000H)
	01111111	11111111	(7FFFH)

-Vmax	00000000	00000000	(0000H)

Figura 5. Range e codifica dati a 16 bit che sono trasferiti al DAC.

Il diagramma di flusso di figura 6 riassume la procedura di conversione dei dati: il processore legge il dato numerico codificato a 24 bit (rappresentato nei tre byte LSB_D, CSB_D e MSB_D) ed effettua un controllo sul byte più significativo (MSB_D) e su quello intermedio (CSB_D): sono considerati validi e trasferiti su una variabile i bit meno significativi dei 24 bit iniziali. I dati con rappresentazione binaria fuori tale *range* di conversione del DAC (figura 5) sono considerati come fondo scala e quindi vengono posti ai valori 00000000h o FFFFFFFFh. Tutti gli altri valori vengono invece trasformati e normalizzati secondo lo schema di figura 5 tramite la negazione del bit più significativo del CSB_D e memorizzati in memoria a partire dalla locazione 5000h fino alla locazione 6000h. Il processore presenta un bus dati a 8 bit quindi i nuovi dati codificati a 16 bit sono trasferiti nel registro del DAC attraverso due operazioni di scrittura (una per byte); difatti il registro d'ingresso del DAC viene visto dal processore come due byte (o locazioni) ai quali sono assegnati rispettivamente gli indirizzi E000h e C000h nella mappa degli indirizzi.

Al termine della scrittura un interrupt generato dal *timer0* avvia la conversione del dato in formato analogico e questa operazione si ripete ogni 20 ms mantenendo il sincronismo con la rotazione del rullo (1 mm/s). Parallelamente all'azione dell'interrupt il programma genera un segnale marcatempo e scrive sul display l'orario da associare ai campioni convertiti. Terminata la conversione della serie numerica trasportata dal pacchetto ricevuto, il processore riparte per un nuovo ciclo controllando la presenza di un carattere ricevuto nella *seriale0*.

La *seriale0* consente all'utente di effettuare dei controlli sul funzionamento dell'*hardware*, interagendo attraverso un PC sul quale sia installato un qualsiasi software per la gestione della RS232 come ad esempio il Terminale di Windows, Procomm, Teraterm ecc.; inoltre sul display LCD viene segnalata l'eventuale interruzione del flusso dati.

I segnali di controllo del DAC e del display sono ottenuti attraverso una decodifica del bus indirizzi operata da un dispositivo a logica programmata della Lattice (GAL22V10, del quale si riporta il codice ABEL in appendice B), affinché l'operazione di scrittura verso i due dispositivi sia completamente trasparente al processore.

In figura 7 è mostrato il prototipo montato nella sala operativa INGV, mentre nelle figure 8 e 9 sono riportate due tracce sismiche relative al medesimo terremoto e registrate su stazioni differenti: la prima con l'helicorder digitale e la seconda con quello classico.

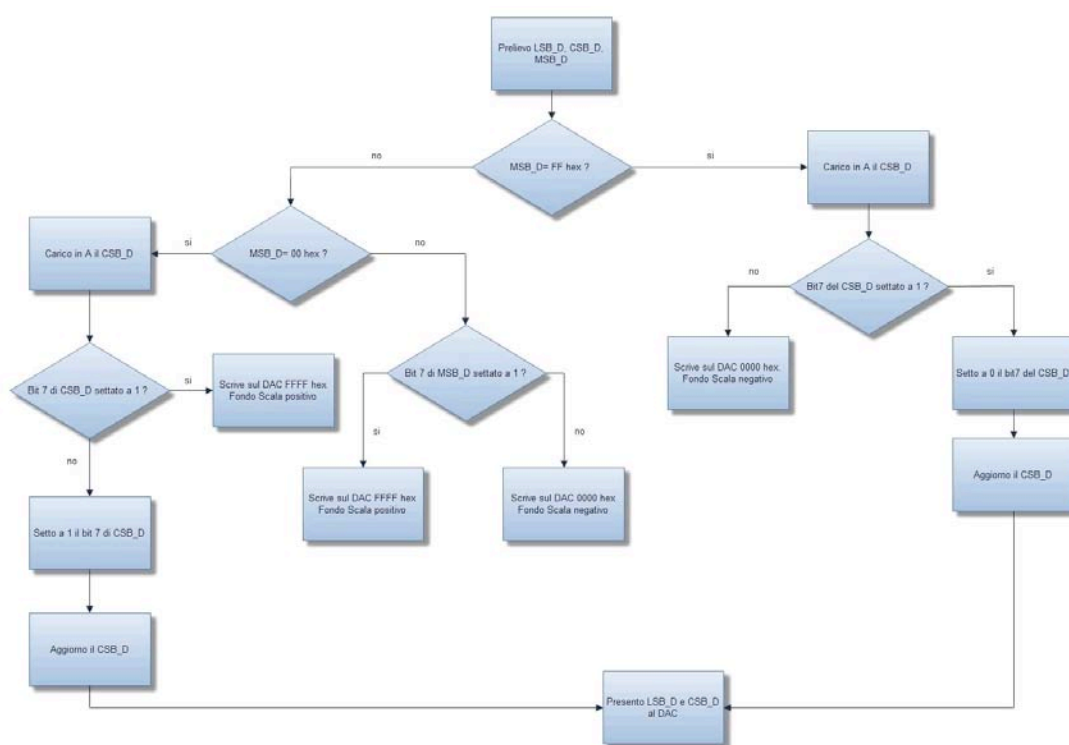


Figura 6. Diagramma di flusso della *routine* di conversione.



Figura 7. Prototipo dell'interfaccia per il rullo digitale.

2. Futuri sviluppi

Il prototipo descritto in questa nota è attualmente utilizzato con successo per la visualizzazione di un segnale sismico ricevuto in forma numerica su una linea CDN secondo lo standard RS-232.

Esso rappresenta una prima fase di sviluppo di un sistema in grado di ricevere un flusso di dati sismici direttamente attraverso un collegamento di rete; infatti, la nuova e sempre più diffusa modalità di trasferimento dati basata sull'accesso a reti informatiche, sta soppiantando le trasmissioni di tipologia CDN. Quindi per un prossimo futuro si può pensare di equipaggiare il sistema descritto con un opportuno modulo elettronico dotato di porta ethernet con accesso a internet e in grado di ricevere i pacchetti di dati e trasferirli su uno standard RS-232. Con l'aggiunta di questo modulo e l'uso dell'elettronica descritta in questa nota è possibile visualizzare sui classici rulli helicorder i segnali sismici provenienti dalla nuova rete sismica nazionale.

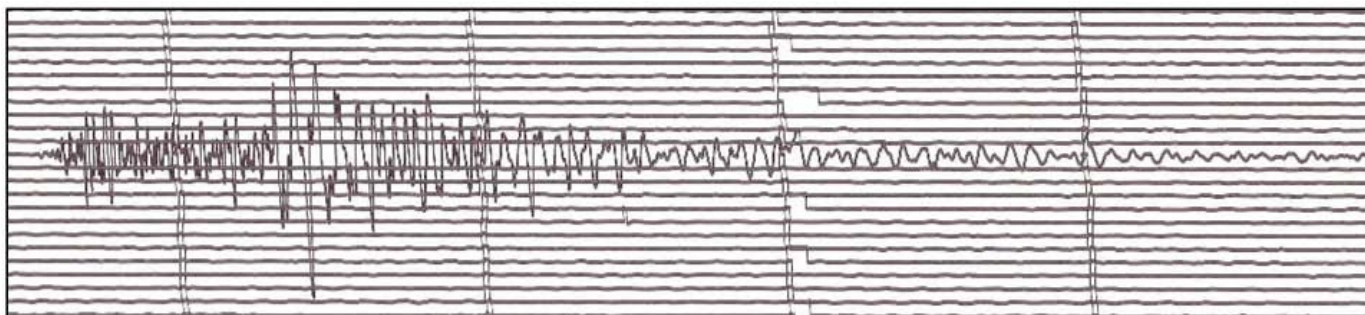


Figura 8. Traccia sismica ottenuta con l'helicorder digitale. Il segnale proviene dalla stazione digitale di Villavallelonga (AQ). Il sensore impiegato è un Lennartz LE-3D 5 sec. L'evento si è verificato il 25/11/01 alle ore 00:57 GMT, Md=4.3 localizzato nel Casentino lat. 43.6N lon. 12.1E.

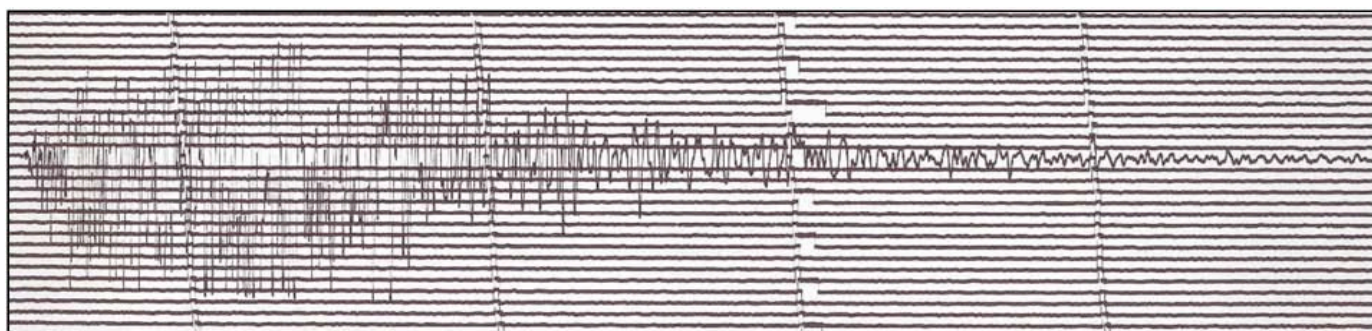


Figura 9. Traccia sismica ottenuta con l'helicorder classico. Il segnale proviene dalla stazione analogica di Pietraquaria (AQ). Il sensore impiegato è un Geotech S-13 1sec. L'evento è il medesimo della figura precedente.

APPENDICE A - Programma assembler HELIDAC.ASM

```
$debug
$xref
$nomod51          ;disattiva i parametri 8051
$include(reg320.pdf) ;attiva i parametri 80320
                  ;helidac.asm aggiornato al 12/07/2001 11.30

;***** Applicativo per modulo D/A converter helicorder *****
;***** header 249 *****

display0          equ 0a000h;          ;indirizzo display scrittura caratteri
display1          equ 0a001h;          ;indirizzo display scrittura istruzioni
dac_l             equ 0e000h          ;locazione dac LSB
dac_m             equ 0c000h          ;locazione dac MSB + strobe scrittura
sign_zero_l       equ 00h            ;valore azzeramento segnale dac LSB
sign_zero_m       equ 80h            ;valore azzeramento segnale dac MSB
status            equ 5eh            ;registro codice status di esecuzione
car_ric           equ 5fh            ;registro carattere ricevuto dalla seriale
head_ff_qta       equ 04h            ;q.ta' di FFH nel codice apertura header
head_00_qta       equ 04h            ;q.ta' di 00H nel codice apertura header
head_S_pos        equ 32d            ;posizione dati sismici
codiceA           equ 41h            ;codice comando "A"
codiceB           equ 42h            ;codice comando "B"
codiceF           equ 46h            ;codice comando "F"
appo_dac_l        equ 60h            ;registro per lsb di scrittura del dac
appo_dac_m        equ 61h            ;registro per msb di scrittura del dac
rit0              equ 62h            ;registro riservato per cicli di ritardo
rit1              equ 63h            ;registro riservato per cicli di ritardo
rit2              equ 64h            ;registro riservato per cicli di ritardo
rit3              equ 65h            ;registro riservato per cicli di ritardo
rit4              equ 66h            ;registro riservato per cicli di ritardo
rit5              equ 67h            ;registro riservato per cicli di ritardo
buffer_1          equ 5000h          ;buffer 1 dati sismici validi
q_buff_1          equ 50d            ;quantita' campioni a 24bit da ricevere
r_buff_1          equ 68h            ;registro riservato contatore campioni
lsb_d             equ 69h            ;registro riservato dati sismici
csb_d             equ 6ah            ;registro riservato dati sismici
msb_d             equ 6bh            ;registro riservato dati sismici
buff_1h           equ 6ch            ;registro riservato msb buffer 1
buff_1l           equ 6dh            ;registro riservato lsb buffer 1
buffer_dac        equ 6000h          ;buffer scrittura dac
buff_dac_h        equ 6eh            ;registro riservato msb buffer dac
buff_dac_l        equ 6fh            ;registro riservato lsb buffer dac
q_buff_d          equ 100d           ;quantita' byte per buffer dac
r_buff_di         equ 70h            ;registro byte per buffer dac interrupt
q_buff_di         equ 50d            ;quantita' byte per buffer dac interrupt
time_h            equ 71h            ;registro appoggio ora
time_m            equ 72h            ;registro appoggio minuti
time_s            equ 73h            ;registro appoggio secondi
flag_m            equ 74h            ;registro flag lunghezza marca minuto
flag_h            equ 75h            ;registro flag lunghezza marca ora
flag_g            equ 76h            ;registro flag lunghezza marca giorno
old_sec           equ 77h            ;registro appoggio secondo precedente
bta1              equ 7bh            ;reg. app.conv. bin_to_ascii "100"
bta2              equ 7ch            ;reg. app.conv. bin_to_ascii "10"
bta3              equ 7dh            ;reg. app.conv. bin_to_ascii "1"
staz1             equ 50h            ;appoggio 1º carattere nome stazione
staz2             equ 51h            ;appoggio 2º carattere nome stazione
staz3             equ 52h            ;appoggio 3º carattere nome stazione
staz4             equ 53h            ;appoggio 4º carattere nome stazione
staz5             equ 54h            ;appoggio 5º carattere nome stazione
```

```

    regis1      equ 55h                ;registro quantita' caratteri display
    xxx1        equ 56h                ;appoggio scrittura messaggio
    xxx2        equ 57h                ;appoggio scrittura messaggio
;*****
;***** MAPPA INDIRIZZAMENTO MEMORIA *****
;*****
;
; 0000H-3FFFFH Lettura e scrittura (su ram, come ram)
; 0000H-3FFFFH Lettura e scrittura (su eprom flash, come codice macchina)
; 4000H-7FFFFH Lettura e scrittura (su ram, come ram)
; 4000H-7FFFFH Lettura (su ram come rom simulata codice macchina)
;ATTENZIONE:le operazioni di scrittura su eprom flash sono condizionate dalla
;porta processore P1.7 (vedi documento descrittivo firmware AGDFBASE.DOC)
; 0C000H indirizzo porta DAC byte lsb
; 0E000H indirizzo porta DAC byte msb
;*****
;***** TABELLA CODICI FUNZIONE *****
;*****
;          carattere "A" (41H) normale conversione analogico/digitale
;          carattere "B" (42H) genera un segnale di prova 0000H-FFFFH
;          carattere "F" (46H) interruzione applicativo
;*****
;***** inizio programma applicativo *****
;*****
        org 000bh
        ljmp dacca_int ;instradamento vettore interrupt timer 0

        org 1000h
        jmp avvio
versione: db 'PPRODACZ '                ;dati identificativi firmware
avvio:    lcall ini_displ
        lcall messaggio2                ;scrive sul display messaggio errore
        clr p3.4                        ;azzeramento segnale marca tempo
        orl ckcon,#0c0h                 ;attivazione watch-dog
        mov ta,#0aah
        mov ta,#55h
        setb rwt
        mov ta,#0aah
        mov ta,#55h
        clr wdif
        clr wtrf
        mov ta,#0aah
        mov ta,#55h
        setb ewt
        mov old_sec,#0ffh
        mov flag_m,#00h
        mov flag_h,#00h
        mov flag_g,#00h
        jnb ri,proseg
        clr ri
        mov car_ric,sbuf                ;codice che esegue carat. comando
        jmp prosegg
proseg:   mov car_ric,#codiceA           ;codice che esegue all'avvio
prosegg:  lcall azz_dac
        mov th1,#0f7h                    ;imposta la seriale 1 a 19200b/s
cont_car: mov a,car_ric
        cjne a,#codiceA,cont_B
        lcall run_A
        jmp cont_car
cont_B:   cjne a,#codiceB,cont_F
        lcall run_B
        jmp cont_car

```

```

cont_F:      cjne a,#codiceF,cont_esec
             lcall azz_dac
             clr ie.7           ;disabilita tutte le interruzioni
             clr ie.1           ;disabilita interrupt timer 0
             mov ta,#0aah       ;disattiva w.d.timer reset
             mov ta,#55h
             clr ewt
             ljmp 2000h
cont_esec:   mov car_ric,status
             jmp cont_car

;***** routine di esecuzione normale conversione analogico/digitale *****
run_A:      setb ie.7           ;abilita tutte le interruzioni
             mov status,#codiceA ;riconosce codice A e invia messaggio
             mov dptr,#mess1     ;dello stato di funzionamento sulla
             lcall send           ;seriale 0
             mov dptr,#messA
             lcall send
             mov a,tmod
             orl a,#00000001b     ;imposta timer 0 modo 1 (counter 16bit)
             mov tmod,a
ddd:        mov rit0,#head_ff_qta ;procedura per il riconoscimento
ddd1:       jnb 0c0h,$           ;del codice di apertura dell'header
             clr 0c0h           ;ricevuto sulla seriale 1
             mov a,0c1h
             cjne a,#0ffh,ddd     ;caratteri ffh
             djnz rit0,ddd1
ddddff:     jnb 0c0h,$           ;routine per l'aggancio sulla
             clr 0c0h           ;variazione ffffffff000000
             mov a,0c1h         ;i dati sismici nel header 208
             cjne a,#00h,ddddff  ;sono in coda e se contengono
             jmp ddd00          ;valori continui di fffffff possono
dddff:      cjne a,#0ffh,ddd     ;essere confusi con l'header stesso
             jmp dddddff
ddd00:      mov rit0,#head_00_qta
             dec rit0
ddd2:       jnb 0c0h,$
             clr 0c0h
             mov a,0c1h
             cjne a,#00h,ddd     ;caratteri 00h
             djnz rit0,ddd2
             mov rit0,#6d        ;salta 6 caratteri e raggiunge
abc2:       jnb 0c0h,$           ;l'ora , minuti , secondi
             clr 0c0h
             djnz rit0,abc2
             jnb 0c0h,$         ;appoggia nel registro ora
             clr 0c0h
             mov time_h,0c1h
             jnb 0c0h,$         ;appoggia nel registro minuti
             clr 0c0h
             mov time_m,0c1h
             jnb 0c0h,$         ;appoggia nel registro secondi
             clr 0c0h
             mov time_s,0c1h
             mov rit0,#7d        ;salta 7 caratteri e raggiunge
abc3:       jnb 0c0h,$           ;nome stazione
             clr 0c0h
             djnz rit0,abc3
             jnb 0c0h,$
             clr 0c0h
             mov staz1,0c1h      ;primo carattere nome stazione
             jnb 0c0h,$

```

```

        clr 0c0h
        mov staz2,0c1h          ;secondo carattere nome stazione
        jnb 0c0h,$
        clr 0c0h
        mov staz3,0c1h          ;terzo carattere nome stazione
        jnb 0c0h,$
        clr 0c0h
        mov staz4,0c1h          ;quarto carattere nome stazione
        mov rit0,#3d            ;salta 3 caratteri e raggiunge
abc4:    jnb 0c0h,$              ;componente
        clr 0c0h
        djnz rit0,abc4
        jnb 0c0h,$
        clr 0c0h
        mov staz5,0c1h          ;carattere componente
        mov a,0c1h
        cjne a,'#Z',add         ;controllo componente "Z"
        jnb ti,$                ;invio carattere sulla seriale 0
        clr ti
        mov sbuf,a
        jmp bddd
add:     ljmp ddd
bddd:    mov rit0,#head_S_pos    ;riceve 32 byte e raggiunge dati sismici
dddd:    jnb 0c0h,$
        clr 0c0h
        djnz rit0,dddd
        mov dptr,#buffer_1      ;inizializza buffer 1 dati sism. validi
        mov buff_1h,dph
        mov buff_1l,dpl
        mov r_buff_1,#q_buff_1 ;inizializza registro contatore
rianalisi: lcall analisi         ;analisi e memor.datati validi
        djnz r_buff_1,rianalisi
        mov r_buff_1,#q_buff_d ;inizializza registro contatore
        mov dptr,#buffer_1      ;trasferimento in zona scrittura dac
        inc dps
        mov dptr,#buffer_dac
        inc dps
buffa_dac: movx a,@dptr
        inc dptr
        inc dps
        movx @dptr,a
        inc dptr
        inc dps
        djnz r_buff_1,buffa_dac
        mov dptr,#buffer_dac
        mov buff_dac_h,dph
        mov buff_dac_l,dpl
        mov r_buff_di,#q_buff_di;inizializza contatore routine interrupt
        lcall cont_seq_sec
        setb ie.1                ;attiva interrupt timer 0
        setb tf0                ;attiva il primo interrupt per sincronizz.
        lcall marcatempo
        lcall scrivi_orario
        jnb ri,jjj
        clr ri
        mov car_ric,sbuf
        clr ie.7
        clr ie.1
        clr tr0
        lcall azz_dac
        ret
jjj:     ljmp ddd

```

```

marcatempo:  mov a,time_s
              cjne a,#00h,marca_esci
              mov a,time_m
              cjne a,#00h,marca_1sec
              mov a,time_h
              cjne a,#00h,marca_4sec
              setb p3.4                ;scrive marca tempo da 8 secondi
              mov flag_g,#0ffh
              jmp marca_esci

marca_4sec:   setb p3.4                ;scrive marca tempo da 4 secondi
              mov flag_h,#0ffh
              jmp marca_esci

marca_1sec:   setb p3.4                ;scrive marca tempo da 1 secondi
              mov flag_m,#0ffh
              jmp marca_esci

marca_esci:   mov a,flag_m
              cjne a,#0ffh,esci_m
              mov a,time_s
              cjne a,#01h,esci_g
              clr p3.4
              mov flag_m,#00h

esci_m:        mov a,flag_h
              cjne a,#0ffh,esci_h
              mov a,time_s
              cjne a,#04h,esci_g
              clr p3.4
              mov flag_h,#00h

esci_h:        mov a,flag_g
              cjne a,#0ffh,esci_g
              mov a,time_s
              cjne a,#08h,esci_g
              clr p3.4
              mov flag_g,#00h

esci_g:        ret

cont_seq_sec:  mov a,old_sec
              cjne a,#0ffh,ccont_seq_sec
              mov old_sec,time_s
              ret

ccont_seq_sec: cjne a,#59d,controlla
              mov a,time_s
              cjne a,#00d,errore_s
              mov old_sec,time_s
              ret

controlla:    add a,#01d
              cjne a,time_s,errore_s
              mov old_sec,time_s
              ret

errore_s:     jnb ti,$
              clr ti
              mov sbuf,'#E'
              jmp errore_s
              ret

```

```

;***** routine di generazione segnale di prova (rampa a circa 2 Hz) *****
run_B:
        clr ie.7                ;disabilita tutte le interruzioni
        clr ie.1                ;disabilita interrupt timer 0
        mov status,#codiceB
        mov dptr,#mess1
        lcall send
        mov dptr,#messB
        lcall send
        lcall messaggio3
rannaB:
        mov appo_dac_l,dpl
        mov appo_dac_m,dph
        inc dptr
        lcall dacca
        mov ta,#0aah            ;azzerare conteggio w.d.timer reset
        mov ta,#55h
        setb rwt
        jnb ri,rannaB
        clr ri
        mov car_ri,sbuf
        lcall azz_dac
        ret

;*****routine di conversione dati sismici da 24bit in complemento a 2 in
;*****16bit normalizzati.

analisi:
        jnb 0c0h,$
        clr 0c0h
        mov lsb_d,0c1h         ;appoggio temporaneo lsb
        jnb 0c0h,$
        clr 0c0h
        mov csb_d,0c1h         ;appoggio temporaneo csb
        jnb 0c0h,$
        clr 0c0h
        mov msb_d,0c1h         ;appoggio temporaneo msb
        mov a,msb_d
        cjne a,#0ffh,vedi00
        mov a,csb_d
        jnb 0e7h,non_val_neg
        clr 0e7h
        mov csb_d,a
        jmp memo_buff1
vedi00:
        cjne a,#00h,cont_non
        mov a,csb_d
        jb 0e7h,non_val_pos
        setb 0e7h
        mov csb_d,a
        jmp memo_buff1
cont_non:
        jb 0e7h,non_val_neg
        jmp non_val_pos

memo_buff1:
        mov dph,buff_1h
        mov dpl,buff_1l
        mov a,lsb_d
        movx @dptr,a
        inc dptr
        mov a,csb_d
        movx @dptr,a
        inc dptr
        mov buff_1h,dph
        mov buff_1l,dpl

```

```

ret

non_val_neg:  mov dph,buff_1h
               mov dpl,buff_1l
               mov a,#00h
               movx @dptr,a
               inc dptr
               movx @dptr,a
               inc dptr
               mov buff_1h,dph
               mov buff_1l,dpl
               ret

non_val_pos:  mov dph,buff_1h
               mov dpl,buff_1l
               mov a,#0ffh
               movx @dptr,a
               inc dptr
               movx @dptr,a
               inc dptr
               mov buff_1h,dph
               mov buff_1l,dpl
               ret

dacca:        push dpl
               push dph
               push acc
               mov dptr,#dac_l
               mov a,appo_dac_l
               movx @dptr,a
               mov dptr,#dac_m
               mov a,appo_dac_m
               movx @dptr,a
               pop acc
               pop dph
               pop dpl
               ret

azz_dac:      mov dptr,#dac_l
               mov a,#sign_zero_l
               movx @dptr,a
               mov dptr,#dac_m
               mov a,#sign_zero_m
               movx @dptr,a
               ret

mess1:        db 0ah
               db 0ah
               db 0ah
               db 0ah
               db 0ah
               db 0ah
               db 0ah
               db 0ah
               db 0ah
               db 0dh
               db 'Applicativo per modulo D/A converter helicorder'
               db 0ah
               db 0dh
               db '          I.N.G.V. Roma laboratorio R.M.S.'
               db 0ah
               db 0dh

```

```

        db '          by C. Acerra e S. Rao - Luglio 2000'
        db 0ah
        db 0ah
        db 0ah
        db 0ah
        db 0dh
        db '      **** MENU CODICI COMANDO ****'
        db 0ah
        db 0ah
        db 0dh
        db 'carattere "A" conversione digitale/analogico dati RS232'
        db 0ah
        db 0dh
        db 'carattere "B" genera segnale di prova 0000H-FFFFH'
        db 0ah
        db 0dh
        db 'carattere "F" interruzione applicativo'
        db 'œ'

messA:   db 0ah
        db 0ah
        db 0ah
        db 0dh
        db 'Codice  "A" in esecuzione.....'
        db 'œ'

messB:   db 0ah
        db 0ah
        db 0ah
        db 0dh
        db 'Codice  "B" in esecuzione.....'
        db 'œ'

send:    mov a,#00h
        movc a,@a+dptr
        cjne a,#'œ',senda
        ret

senda:   jnb ti,$
        clr ti
        mov sbuf,a
        inc dptr
        jmp send

sends:   jnb ti,$
        clr ti
        mov sbuf,a
        ret

dacca_int:  push dpl          ;routine interrupt timer 0
        push dph
        push acc
        clr tr0
        mov th0,#70h
        mov tl0,#00h
        setb tr0
        mov dph,buff_dac_h
        mov dpl,buff_dac_l
        movx a,@dptr
        inc dptr
        mov buff_dac_h,dph
        mov buff_dac_l,dpl
        mov dptr,#dac_l
        movx @dptr,a

```

```

    mov dph,buff_dac_h
    mov dpl,buff_dac_l
    movx a,@dptr
    inc dptr
    mov buff_dac_h,dph
    mov buff_dac_l,dpl
    mov dptr,#dac_m
    movx @dptr,a
    djnz r_buff_di,ff
    clr ie.1

    mov ta,#0aah          ;azzerà conteggio w.d.timer reset
    mov ta,#55h          ;ogni secondo
    setb rwt

ff:    pop acc
    pop dph
    pop dpl
    reti

;ROUTINE DI INIZIALIZZAZIONE DISPLAY CARATTERI
ini_displ:    mov dptr,#display0;indirizzo instruction register
    mov a,#38h          ;set 8 bit,caratteri 5x7
    lcall writ
    mov a,#0ch          ;display on e cursor off
    lcall writ
    mov a,#06h          ;cursor shift to the right
    lcall writ
    mov a,#01          ;display clear
    lcall writ
    lcall rit
    ret

rit:    mov rit1,#80h
rit9:    mov rit2,#0ffh
    djnz rit2,$
    djnz rit1,rit9
    ret

writ:    movx @dptr,a
    mov rit1,#0ffh
    djnz rit1,$
    ret

;routine che trasforma un byte binario in tre byte ascii
;il byte in ingresso deve essere contenuto nell'accumulatore
;i byte di uscita sono bta1=centinaia,bta2=decine,bta3=unita'

bin_to_ascii: mov b,#100d          ;divisione del numero per 100 decimale
    div ab          ;l'intero e' in (A) ed il resto in (B)
    add a,#30h      ;trasformazione in numero ASCII
    mov bta1,a      ;trasferimento nel registro di uscita
    mov a,b          ;resto da dividere in (A)
    mov b,#10d      ;divisione del resto per 10 decimale
    div ab          ;l'intero e' in (A) ed il resto in (B)
    add a,#30h      ;trasformazione in numero ASCII
    mov bta2,a      ;trasferimento nel registro di uscita
    mov a,b          ;resto in (A)
    add a,#30h      ;trasformazione in numero ASCII
    mov bta3,a      ;trasferimento nel registro di uscita
    ret            ;fine routine.

```

```

scrivi_orario:
    mov a,staz1          ;scrive nome stazione
    lcall scrivi_displ
    mov a,staz2
    lcall scrivi_displ
    mov a,staz3
    lcall scrivi_displ
    mov a,staz4
    lcall scrivi_displ
    mov a,staz5
    lcall scrivi_displ

    mov a,'#' '          ;spazi di separazione
    lcall scrivi_displ
    lcall scrivi_displ

    mov a,time_h          ;preleva ore
    lcall bin_to_ascii ;conversione in numero ascii
    mov a,bta2
    lcall scrivi_displ
    mov a,bta3
    lcall scrivi_displ
    mov a,#':' '
    lcall scrivi_displ
    mov a,time_m          ;preleva minuti
    lcall bin_to_ascii ;conversione in numero ascii
    mov a,bta2
    lcall scrivi_displ
    mov a,bta3
    lcall scrivi_displ
    mov a,#':' '
    lcall scrivi_displ
    mov a,time_s          ;preleva secondi
    lcall bin_to_ascii ;conversione in numero ascii
    mov a,bta2
    lcall scrivi_displ
    mov a,bta3
    lcall scrivi_displ

    lcall messaggiol      ;messaggio di corretto funzionamento

    mov dptr,#display0;indirizzo instruction register
    mov a,#02h           ;cursor home
    movx @dptr,a

    ret

scrivi_displ: mov dptr,#display1
    movx @dptr,a
    mov rit1,#0ffh
    djnz rit1,$
    ret

messaggiol:  mov regis1,#65d
    mov dptr,#mess_ok
    mov xxx1,dph
    mov xxx2,dpl
messag:      mov dph,xxx1
    mov dpl,xxx2
    mov a,#00h
    movc a,@a+dptr

```

```

        inc dptr
        mov xxx1,dph
        mov xxx2,dpl
        lcall scrivi_displ
        djnz regis1,messag
        ret

messaggio2:  mov regis1,#80d
             mov dptr,#mess_errore
             mov xxx1,dph
             mov xxx2,dpl
messagg:    mov dph,xxx1
             mov dpl,xxx2
             mov a,#00h
             movc a,@a+dptr
             inc dptr
             mov xxx1,dph
             mov xxx2,dpl
             lcall scrivi_displ
             djnz regis1,messagg
             ret

messaggio3:  mov regis1,#80d
             mov dptr,#mess_test
             mov xxx1,dph
             mov xxx2,dpl
messaggg:    mov dph,xxx1
             mov dpl,xxx2
             mov a,#00h
             movc a,@a+dptr
             inc dptr
             mov xxx1,dph
             mov xxx2,dpl
             lcall scrivi_displ
             djnz regis1,messaggg
             ret

mess_ok:
db 'SENS.: Lennartz 3D-5sec.  ***** FUNZIONAMENTO REGOLARE ***** '

mess_errore:
db '*****  A T T E N Z I O N E !  *****'
db 'Flusso dati non presente o non corretto!'

mess_test:
db '***** Service Hardware Test *****'
db '***** SEGNALE DI PROVA ATTIVO! *****'

        end

```

APPENDICE B- Struttura dati del pacchetto trasmesso dalla stazione digitale

BYTE TRASMESSO		HEX	DESCRIZIONE
1	00	FFH	codice apertura header
2	01	FFH	codice apertura header
3	02	FFH	codice apertura header
4	03	FFH	codice apertura header
5	04	00H	codice apertura header
6	05	00H	codice apertura header
7	06	00H	codice apertura header
8	07	00H	codice apertura header
9	08	00H	contatore pacchetto LSB (da 0000H a FFFFH)
10	09	00H	contatore pacchetto MSB (da 0000H a FFFFH)
11	0A		indicazione ANNO GPS LSB
12	0B		indicazione ANNO GPS MSB
13	0C		indicazione MESE GPS
14	0D		indicazione GIORNO GPS
15	0E		indicazione ORE GPS
16	0F		indicazione MINUTI GPS
17	10		indicazione SECONDI GPS
18	11	00H	riservato
19	12	00H	UTC TICS LSB (decimillesimi di secondo)
20	13	00H	UTC TICS MSB (decimillesimi di secondo)
21	14	F9H	grandezza pacchetto LSB (valore 249 decimale)
22	15	00H	grandezza pacchetto //
23	16	00H	grandezza pacchetto //
24	17	00H	grandezza pacchetto MSB
25	18	41H	1° carattere nome stazione (carattere A ascii)
26	19	47H	2° carattere nome stazione (carattere G ascii)
27	1A	44H	3° carattere nome stazione (carattere D ascii)
28	1B	46H	4° carattere nome stazione (carattere F ascii)
29	1C	4EH	5° carattere nome stazione (carattere N ascii)
30	1D	48H	codice ch (carattere H ascii)
31	1E	48H	codice ch (carattere H ascii)
32	1F	4EH	codice ch (carattere N ascii)
33	20	01H	codice strumento
34	21	01H	flag attività
35	22	28H	valore di fondo scala LSB (9000 decimale 9000mV)
36	23	23H	valore di fondo scala MSB (9000 decimale 9000mV)
37	24	32H	indicazione quantità campioni nel pacchetto LSB (50 decimale)
38	25	00H	indicazione quantità campioni nel pacchetto //
39	26	00H	indicazione quantità campioni nel pacchetto //
40	27	00H	indicazione quantità campioni nel pacchetto MSB
41	28	32H	indicazione quantità campioni al secondo LSB (50 decimale)
42	29	00H	indicazione quantità campioni al secondo MSB
43	2A	18H	word lenght in bit (24 decimale)
44	2B	00H	aux data opcode
45	2C		indicazione numero satelliti visibili
46	2D		indicazione numero satelliti agganciati
47	2E		indicazione stato di funzionamento (vedi tabella)
48	2F		errore temporale PPS/DRDY con segno compl.2 LSB
49	30		errore temporale PPS/DRDY con segno compl.2 MSB
50	31	00H	aux data opcode
51	32	00H	aux data opcode
52	33	00H	aux data opcode
53	34	00H	aux data opcode
54	35	00H	aux data opcode

55	36	00H	aux data opcode
56	37	00H	aux data opcode
57	38	00H	aux data opcode
58	39	00H	aux data opcode
59	3A	00H	aux data opcode
60	3B	00H	aux data opcode
61	3C	55H	check code intermedio
62	3D	55H	check code intermedio
63	3E	55H	check code intermedio
64	3F	55H	check code intermedio
65	40		LSB 1° CAMPIONE
66	41		CSB 1° CAMPIONE
67	42		MSB 1° CAMPIONE
68	43		LSB 2° CAMPIONE
69	44		CSB 2° CAMPIONE
70	45		MSB 2° CAMPIONE
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
209	D0		LSB 49° CAMPIONE
210	D1		CSB 49° CAMPIONE
211	D2		MSB 49° CAMPIONE
212	D3		LSB 50° CAMPIONE
213	D4		CSB 50° CAMPIONE
214	D5		MSB 50° CAMPIONE
215	D6		“i” dati GPS relativi ai satelliti canale 1
216	D7		“m” dati GPS relativi ai satelliti canale1
217	D8		“s” dati GPS relativi ai satelliti canale 1
218	D9		“d” dati GPS relativi ai satelliti canale 1
219	DA		“i” dati GPS relativi ai satelliti canale 2
220	DB		“m” dati GPS relativi ai satelliti canale2
221	DC		“s” dati GPS relativi ai satelliti canale 2
222	DD		“d” dati GPS relativi ai satelliti canale 2
223	DE		“i” dati GPS relativi ai satelliti canale 3
224	DF		“m” dati GPS relativi ai satelliti canale3
225	E0		“s” dati GPS relativi ai satelliti canale 3
226	E1		“d” dati GPS relativi ai satelliti canale 3
227	E2		“i” dati GPS relativi ai satelliti canale 4
228	E3		“m” dati GPS relativi ai satelliti canale4
229	E4		“s” dati GPS relativi ai satelliti canale 4
230	E5		“d” dati GPS relativi ai satelliti canale 4
231	E6		“i” dati GPS relativi ai satelliti canale 5
232	E7		“m” dati GPS relativi ai satelliti canale5
233	E8		“s” dati GPS relativi ai satelliti canale 5
234	E9		“d” dati GPS relativi ai satelliti canale 5
235	EA		“i” dati GPS relativi ai satelliti canale 6
236	EB		“m” dati GPS relativi ai satelliti canale6
237	EC		“s” dati GPS relativi ai satelliti canale 6
238	ED		“d” dati GPS relativi ai satelliti canale 6
239	EE		“i” dati GPS relativi ai satelliti canale 7
240	EF		“m” dati GPS relativi ai satelliti canale7
241	F0		“s” dati GPS relativi ai satelliti canale 7
242	F1		“d” dati GPS relativi ai satelliti canale 7
243	F2		“i” dati GPS relativi ai satelliti canale 8
244	F3		“m” dati GPS relativi ai satelliti canale8
245	F4		“s” dati GPS relativi ai satelliti canale 8
246	F5		“d” dati GPS relativi ai satelliti canale 8
247	F6	45H	end (carattere E ascii)
248	F7	4FH	of (carattere O ascii)
249	F8	42H	block (carattere B ascii)

APPENDICE C - Programma in linguaggio ABEL implementato nella GAL22V10

```
MODULE helidac// programma per gal22v10 plcc
//file helidac.abl ver. del 12/07/01 ore 7:45
// programma di controllo per helicorder digitale con dispaly

a15 pin 5;
a14 pin 6;
a13 pin 7;
H,L,X = 1,0,.X.;
Address = [a15,a14,a13,X,X,X,X,X,X,X,X,X,X,X,X,X,X];

!pgm pin 2;
!rd pin 9;
!wr pin 10;
!psen pin 11;
cs_f pin 17;
cs_d pin 18;
rd_r pin 19;
wr_r pin 20;
wr_f pin 21;
rd_f pin 23;
LBE pin 27;
HBE pin 25;

equations
!cs_f = (Address <= ^h3fff) # ((Address > ^H3fff) & pgm);
!rd_f = !pgm & psen;
!wr_f = pgm & wr;
!rd_r = (psen & (Address >= ^H4000)) # rd;
!wr_r = !pgm & wr;
!cs_d = a15 & !a14 & a13 & wr ; //a000h (display)
!LBE = a15 & a14 & a13 ; //e000h (LSB DAC)
!HBE = a15 & a14 & !a13 ; //c000h (MSB DAC)

END
```

Bibliografia

- [1]. S. Pintore, L. Salvaterra (2007) – Il progetto TN-1. Rapporti tecnici INGV, n 40
- [2]. 16-Bit Monotonic Voltage Output D/A Converter AD569, datasheet della Analog Devices. Web: www.analog.com
- [3]. GAL22V10, high performance E²CMOS PLD, Generic Array Logic (July 1997) web : www.latticesemi.com
- [4]. DS80c320 High-Speed User's Guide (January 1994)
- [5]. "Operation and maintenance manual helicorder, model RV-301B", Teledyne Geotech

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