

## In questo numero

Torna nostro ospite il **CW**. Il rapporto fra il computer e la telegrafia è come quello che si può avere con una donna bellissima che, però, non si lascia facilmente circuire... Tutti sappiamo delle difficoltà oggettive che incontra chi voglia programmare un elaboratore per decodificare il CW, dovute principalmente al diverso numero di impulsi di codice necessari per formare i vari caratteri e, nel caso di trasmissione manuale, alla inevitabile diversa durata degli stessi. Il problema non si pone certo per la trasmissione, dove l'uso di un computer ben programmato è quanto di meglio ci possa essere. L'orecchio di un buon "grafista" è, oggi, insostituibile, ma già adesso il computer è in grado di supplire, in parte, a carenze di ricezione e di demodulazione del segnale di bassa frequenza. Apparatì migliori, elaboratori sempre più veloci e potenti, in grado di svolgere in tempo utile algoritmi molto più sofisticati, potranno fare, in futuro, sempre meglio e anche più dell'orecchio umano. Di ciò non me ne vogliono i "puristi" del CW: la mia stima per loro, che grafista non sono, è grande.

La differenza fra il **Morse** ricevuto a "macchina" e quello ad "orecchio" è come quella che intercorre fra la guida di un'auto diesel con il cambio automatico in autostrada e di una sportiva a benzina in salita...: quest'ultima è tutta un'altra cosa!

La penna passa, pertanto, a **Marco Ibridi - I4IBR** - di Finale Emilia (BO) che presenta "Ricevere la telegrafia con il computer". Marco è nel ramo informatico e questo lo si evince subi-

to guardando il suo lavoro e le note che lo accompagnano. Ma il suo programma è apprezzabile soprattutto perché è utilizzabile, in pratica, su molti personal, anche se è implementato in ambiente MS-DOS.

Ricevo lagnanze perché la rubrica non ospita lavori per i computer meno diffusi. Il grosso del materiale che arriva è per il Commodore 64 e, ora, anche per gli IBM e compatibili. Ho voluto inserire questo mese il programma suddetto, anche se è fra gli ultimi pervenuti, proprio perché facilmente adattabile a molte macchine.

Ho sempre sentito dire, e sono d'accordo, che un buon radioamatore debba "fare la radio" dopo aver provveduto al QRA e al QRL. Come per quasi noi tutti, il mio tempo da dedicare alla radio e alla rubrica non è molto. Capita anche di non riuscire a fare il QSO con S. Peter I..., ma spero che vada meglio con le Andaman. Sto rivedendo il materiale pervenuto, non molto a dire il vero, e sto cercando di rendere pubblicabili cose giacenti anche da molti mesi proprio perché non facilmente riproducibili nella rubrica. Non possiamo essere solleciti come un quotidiano: confido nella comprensione di tutti.

Nel ringraziare Marco Ibridi a nome dei lettori per il lavoro inviato ed i lettori per la paziente attenzione prestatami, non mi rimane che augurare buona attività in radio con il computer a tutti.

Francesco - IOFLY

Marco Ibridi - I4IBR

## Ricevere il CW con il computer

1) La telegrafia è un sistema per la trasmissione dati inventato ed adottato molto prima dell'avvento di macchine elettroniche per la trasmissione dati.

Utilizzato prevalentemente nelle comunicazioni radio, specie per quelle d'amatore, si rivela molto efficace e tuttora molto in uso, se non altro per i risultati possibili con mezzi relativamente limitati.

La decodifica automatica di questo sistema diventa quindi estremamente appe-

tibile se consideriamo la difficoltà oggettiva di interpretare "ad orecchio" il codice adottato.

Il più grosso problema per il trattamento automatico dell'informazione con il codice Morse (codice universalmente adottato per la telegrafia classica) è la lunghezza non costante dei singoli caratteri del set.

2) La lettura automatica del codice Morse è da sempre possibile con macchine a

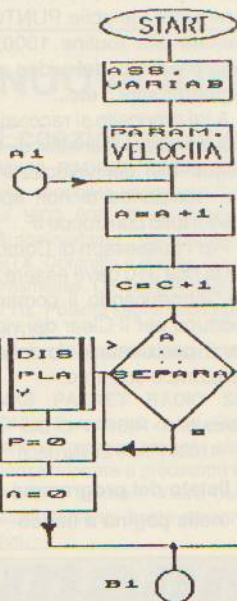


Fig. 1

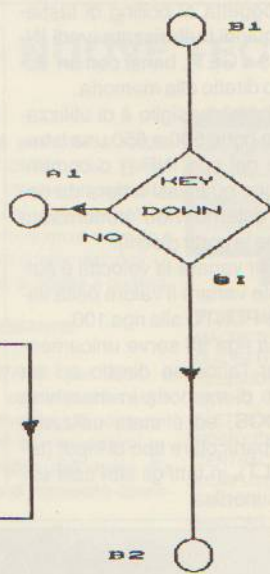


Fig. 2

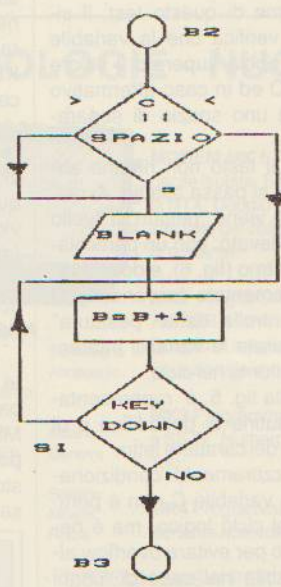


Fig. 3

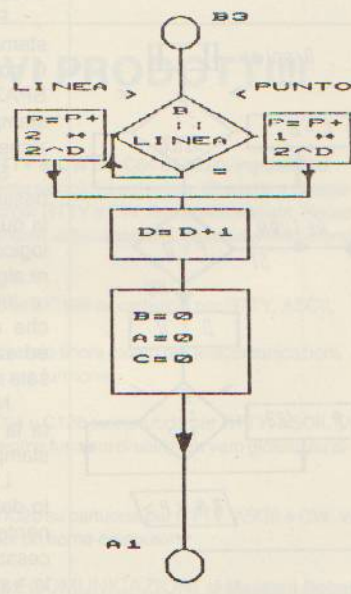


Fig. 4

banda cartacea, che pur non provvedendo in alcun modo alla decodifica del carattere ricevuto restituiscono nella forma grafica di punti e linee, il carattere trasmesso.

3) Con l'avvento di sistemi per il trattamento automatico delle informazioni - dedicati o meno - è stato possibile trattare questo tipo di codice.

Vediamo ora un possibile metodo di lettura automatica del codice Morse, con l'ausilio di un home o di un personal computer.

Il diagramma a blocchi nelle figure 1, 2, 3, 4 e 5 rappresenta il ciclo logico della lettura del carattere.

L'evento determinante per la rilevazione dell'informazione è l'intercettazione del tasto abbassato (key down). In questa maniera l'utente di un sistema con codice Morse avvia la procedura di trasmissione del carattere, avvalendosi (per la diversificazione dello stato logico) della durata temporale dell'evento stesso. In poche parole un tasto abbassato per breve tempo rappresenta l'unità logica PUNTO, un tasto abbassato per maggior tempo rappresenta l'unità logica LINEA.

Diventa quindi evidente che determinante è la corretta

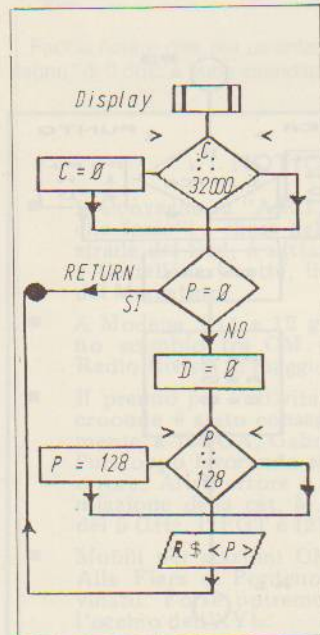


Fig. 5

sincronizzazione della rilevazione degli eventi e delle loro lunghezze relative.

In figura 1 notiamo, dopo l'allocaggio e l'assegnazione delle variabili utilizzate, l'assegnazione dei parametri VELOCITA'. Questi parametri sono i rapporti fondamentali per leggere correttamente questo tipo di codice.

Verranno definiti i valori dell'unità fondamentale che è il PUNTO e delle altre unità derivate (LINEA SEPARAZIONE SPAZIO).

Subito dopo inizia il ciclo di conteggio del tempo in cui il tasto rimane alzato. La variabile utilizzata è la A. Si incrementa pure il contatore C per la separazione di due parole.

Se il valore della variabile A supera quello dell'unità fondamentale per la separazione dei caratteri (SEPARA), il controllo passa alla routine di decodifica e display, per ritornare con l'azzeramento delle variabili significative.

Se la condizione precedente non è verificata si passa alla fig. 2 dove viene testato lo stato del tasto per rilevarne un eventuale abbassamento. Se ciò avviene, in fig. 3 è riportato il ciclo del conteggio del tasto abbassato, per verificare tramite la variabile B se ci troviamo di fronte ad un PUNTO o ad una LINEA.

Prima di questo test, il sistema verifica che la variabile C non abbia superato il valore SPAZIO ed in caso affermativo stampa uno spazio di separazione.

Se il tasto non rimane abbassato si passa alla fig. 4, nella quale viene "pesato" il livello logico rilevato, con un particolare algoritmo (fig. 6), e dopo avere incrementato la variabile D che controlla detta "pesatura" ed azzerate le variabili interessate si ritorna nel ciclo.

Nella fig. 5, è rappresentata la routine di decodifica e di stampa del carattere letto.

L'azzeramento condizionale della variabile C non è pertinente al ciclo logico, ma è necessario per evitare overflow alla variabile nel caso di lunghi tempi di attesa di segnale.

Fig. 6

- Algoritmo di "pesatura" carattere -

|       |   |   |   |   |    |    |    |      |
|-------|---|---|---|---|----|----|----|------|
|       | 1 | 2 | 4 | 8 | 16 | 32 | 64 |      |
| MORSE | 1 | 2 | 2 | 1 |    |    |    | P    |
| IBR   | 1 | 4 | 8 | 8 |    |    |    | = 21 |

4) Implementazione di un semplice programma in linguaggio BASIC universale.

Dal diagramma a blocchi alla stesura del programma, il passo è breve.

Il listato riportato è stato sviluppato su di un sistema MS-DOS, ma è praticamente riproducibile su qualsiasi macchina con interprete BASIC.

Nell'ambiente di sviluppo citato, l'input dell'informazione è costituito dal tasto ALT presente sulla tastiera standard in basso a sinistra.

Non bisogna però farsi fuorviare da questa soluzione. Lo scrivente ha adottato come input un tasto della tastiera per evitare di montare un particolare dispositivo hardware, e comunque lo stato del tasto non è stato rilevato con una istruzione soggetta al polling di tastiera e quindi bufferizzata (vedi INKEY \$ o GET), bensì con un accesso diretto alla memoria.

Il mio consiglio è di utilizzare alle righe 500 e 650 una istruzione del tipo INP () o comunque una qualsiasi lettura da device esterno non bufferizzata (anche le porte giochi).

Per variare la velocità è sufficiente variare il valore della variabile PUNTO alla riga 100.

La riga 95 serve unicamente per l'accesso diretto ad un banco di memoria in macchine MS-DOS, ed è stata utilizzata per il particolare tipo di input (tasto ALT), in tutti gli altri casi essa è superflua.

Per codificare il carattere si è adottato questo metodo:

- in caso di lettura di un PUNTO il sistema moltiplica il valore 1 per la potenza di base 2 elevata alla variabile D.

- in caso di lettura di una LINEA il sistema moltiplica il valore 2 per la potenza di base 2 elevata alla variabile D.

Ad ogni lettura di PUNTO o LINEA detto valore viene sommato nella variabile P ed il valore della variabile D incrementato di una unità - Figura 6.

Il programma così proposto vuole essere solo un esempio ed un canovaccio per ulteriori esperienze ed implementazioni, quali la variazione interattiva della velocità (incremento o decremento variabile PUNTO e chiamata alla routine 1000), il migliore editing del video con adeguati prompt... etc...

A tal proposito si raccomanda di utilizzare variabili di tipo "integer" (%) per velocizzare il ciclo, ricordando di non appesantire il tutto con troppe IF.

Per i possessori di Commodore la riga 310 deve essere variata, introducendo il comando opportuno per il Clear del video e per il posizionamento del cursore ad inizio schermo.

Marco Ibridi - I4IBR - P.O. Box 15 - 41034 Finale Emilia (MO)

Il listato del programma è nella pagina a fianco

Collaborate a Radio Rivista

## Microcomputer

### Ricevere il CW con il computer

```
10 REM *****
20 REM **** RXCW 12/1986 rel.2.1 **
30 REM **** CENTROdata 0535/92750 **
40 REM **** M. IRRIDI - i41BR **
50 REM *****
60 REM ** ASSEGNAZIONI
64 A%=0;B%=0;C%=0;D%=0;F%=0;U%=0
65 PUNTO%=0;LINEA%=0;SPAZIO%=0;SEPARA%=0
66 REM A%=conteggio tasto alzato
67 REM B%=conteggio tasto abbassato
68 REM C%=conteggio tasto alzato dall'ultimo key-down
69 REM D%=peso esponente base 2 per algoritmo decodifica
70 REM F%=peso complessivo carattere per decodifica
71 REM U%=input fisico dal device selezionato
80 REM ** DIMENSIONI
90 DIM R$(128)
95 DEF SEG=0
100 PUNTO%=10;GOSUB 1000
110 REM ** SET-UP
120 REM ** Alfabeto **
130 FOR T=1 TO 128:R$(T)="":NEXT T
140 R$(5)="A";R$(16)="B";R$(20)="C";R$(8)="D"
150 R$(1)="E";R$(19)="F";R$(10)="G";R$(15)="H"
160 R$(3)="I";R$(17)="L";R$(6)="M";R$(4)="N"
170 R$(14)="O";R$(21)="P";R$(26)="Q";R$(9)="R"
180 R$(7)="S";R$(2)="T";R$(11)="U";R$(23)="V"
190 R$(18)="Z";R$(29)="J";R$(12)="K";R$(13)="W"
200 R$(24)="X";R$(28)="Y"
210 REM ** Numeri **
220 R$(61)="1";R$(59)="2";R$(55)="3";R$(47)="4"
230 R$(31)="5";R$(32)="6";R$(34)="7";R$(38)="8"
240 R$(46)="9";R$(62)="0"
250 REM ** Segni **
260 R$(105)=".";R$(114)=",";R$(75)="?";R$(70)=":"
270 R$(84)="!";R$(96)="-";R$(93)="'";R$(40)="/"
280 R$(44)="(";R$(108)=")";R$(81)=CHR$(34)
290 REM **
300 REM ** CLEAR VIDEO
310 PRINT CHR$(12)
320 PRINT "Rx cw on !"
400 REM **
410 REM ** CICLO
420 REM ** Conteggio dello spazio e della separazione
430 REM ** a tasto alzato
450 A%=A%+1
460 C%=C%+1
470 IF A%=>SEPARA% THEN GOSUB 2000:F%=0:A%=0
490 REM ** INPUT STATO TASTO
495 REM ** In macchine MS-DOS il tasto usato e' "ALT"
500 U%=PEEK(&H417):U%=B AND U%
550 IF U%=0 GOTO 450
610 REM **
620 REM ** Tasto abbassato
630 REM **
635 IF C%=>SPAZIO% THEN PRINT CHR$(32);
640 B%=B%+1
645 REM ** INPUT STATO TASTO
650 U%=PEEK(&H417):U%=B AND U%
655 IF U%=B GOTO 640
660 IF B%>LINEA% THEN F%=F%+(2*(2^D%)) ELSE F%=F%+(1*(2^D%))
665 D%=D%+1:IF D%>6 THEN D%=0
666 REM ** Azzeramento variabile D% necessario, per evitare
667 REM ** che brevi segnali incoerenti o ORM vengano ad
668 REM ** incrementare il contatore causando un overflow.
670 B%=0:A%=0;C%=0
680 GOTO 450
1000 REM ** PARAMETRI
1010 LINEA%=PUNTO%*2
1020 SEPARA%=PUNTO%*2
1030 SPAZIO%=PUNTO%*8
1040 RETURN
2000 REM ** DECODIFICA E DISPLAY
2010 IF C%>32000 THEN C%=0
2020 IF F%>128 THEN RETURN
2030 D%=0
2035 IF F%>128 THEN F%=128
2040 PRINT R$(F%);
2050 RETURN
5000 REM *****
5010 REM ** Modifiche per utilizzo key **
5020 REM ** esterno con M24 Olivetti **
5030 REM ** servendosi dell'uscita parallela **
5040 REM ** per stampante :insering il tasto **
5050 REM ** o l'uscita di uno switch **
5060 REM ** elettronico pilotato da decoder, **
5070 REM ** tra i pin 11 e 25 (GND) dell'OUT **
5080 REM ** PARALLELE **
5090 REM *****
5100 REM ** La linea 95 non e' piu' necessaria **
5110 REM ** sostituire le seguenti righe con **
5120 REM ** i nuovi contenuti : **
5130 REM ** 500 U%=INP(&H379) **
5140 REM ** 550 IF U%<>255 GOTO 450 **
5150 REM ** ----- **
5160 REM ** 650 U%=INP(&H379) **
5170 REM ** 655 IF U%=255 GOTO 640 **
5180 REM *****
```