

Vi presentiamo in queste pagine un programma di trasmissione RTTY che usa il Commodore 64. Il suo autore, già noto al pubblico di MC, lo ha approntato, documentato e completato con una serie di applicazioni: noi pubblichiamo per ora la base del lavoro, ripromettendoci magari di tornare sull'argomento con l'aiuto suo e degli altri appassionati.

TRASMISSIONE RTTY

di Giorgio Leo Rutigliano
Via degli Olandri 7, 85100 Potenza

L'RTTY, acronimo di RadioTeleType (radiotelecrivente), è stata negli ultimi quaranta anni una delle metodologie di comunicazione più utilizzate. A partire dalle vecchie telescriventi meccaniche, il sistema è andato via via progredendo. L'avvento dei Personal e Home computer, comunque, ha reso definitivamente popolare l'RTTY nel campo degli amatori, rendendo possibile, con modesti investimenti, l'apertura di una finestra nel mondo delle comunicazioni professionali.

Prima, infatti, la telescrivente era sempre un ammasso di ferraglia MOLTO rumorosa ed ingombrante: queste sole caratteristiche, unite anche ad un costo generalmente elevato, erano sufficienti a far desistere molti appassionati dal tentare la via dell'RTTY. D'altro canto gli stimoli erano notevoli: la possibilità di colloquiare attraverso messaggi scritti e l'ascolto delle varie Agenzie di Stampa sono esempi significativi delle attrattive che il sistema RTTY è in grado di offrire.

Alcuni anni or sono cominciarono ad apparire sul mercato sistemi digitali di ricezione RTTY. Questi apparati, all'inizio realizzati in logica cablata e successivamente basati su microprocessore, erano e sono tutt'ora costosi e poco flessibili.

Il vero impulso alla diffusione della RTTY è stato dato dal proliferare del Personal e Home computer. L'appassionato può avere, infatti, ad un prezzo pari (e a volte minore) di una 'tastiera dedicata' una macchina incomparabilmente più flessibile e che, comunque, può essere utilizzata anche per altri scopi sia nell'ambito dell'hobby della radio, sia in altri settori.

Cenni teorici

Per trasferire dati tra apparati digitali vengono impiegati principalmente due sistemi: parallelo e seriale. Nel primo tutti i bit costituenti l'informazione vengono trasmessi contemporaneamente su fili diversi, mentre un altro circuito indica al dispositivo ricevente quando poter leggere il dato. La trasmissione seriale, invece, avviene attraverso un solo canale di comunicazione, ed i vari bit sono trasmessi sequenzialmente, partendo dal bit meno significativo.

Inoltre ogni dato viene preceduto e seguito da due impulsi (start e stop bit) che indicano al dispositivo ricevente l'inizio e la fine della parola. Opzionalmente può essere aggiunto tra il dato e lo stop-bit un ulteriore impulso, detto bit di parità, che serve a verificare che la parola sia stata ricevuta correttamente. Per definizione lo stato logico 1 viene denominato 'mark' e lo stato logico 0 'space'.

Ovviamente è necessario stabilire un 'alfabeto' che consenta ai due sistemi — trasmettente e ricevente — di poter decodificare correttamente le informazioni. Tra computer è molto diffuso l'ASCII (American Standard Code for Information Interchange), codice a 7 bit, che consente di trasmettere 128 caratteri diversi. Di questi alcuni — detti caratteri di controllo — non vengono stampati ma sono usati per impartire comandi al dispositivo ricevente. In

RTTY viene invece utilizzato il codice BAUDOT, a 5 bit, che consente di trasmettere 32 caratteri. Questi, in effetti, sono pochi rispetto alle esigenze: è necessario poter trasferire i 26 caratteri dell'alfabeto inglese, le dieci cifre ed alcuni segni di interpunzione, oltre ai caratteri di controllo, e 32 posizioni non bastano. La soluzione a questo problema è abbastanza semplice: far corrispondere ad ogni codice due caratteri — uno alfabetico l'altro simbolo/numerico — e definire due codici che permettano di selezionare quale set è correntemente in uso. Il sistema Baudot, quindi, è un 'alfabeto' a due livelli. Il set di caratteri alfabetici viene definito 'letters', quello simbolo/numerico 'figs'. Due caratteri, chiamati appunto 'figs' e 'ltrs' consentono di modificare il set in uso. Inoltre, per brevità, alcuni caratteri molto usati (spazio, ritorno carrello, interlinea) hanno lo stesso significato in entrambi i set di caratteri.

Una volta stabilito l'alfabeto e la metodologia di trasmissione, è necessario codificare anche la velocità con cui i dati debbono essere trasferiti. Mentre nella trasmissione parallela questo dato ha importanza relativa poiché attraverso opportuni segnali il dispositivo ricevente può condizionare quello trasmettente, nella trasmissione seriale questo fattore è di importanza vitale. Due dispositivi operanti a velocità diverse, infatti, non potranno mai scambiare dati su un canale seriale. In ASCII la velocità, generalmente, varia da 110 a 9600 baud (il baud è l'unità di misura della velocità ed è pari a 1 bit al secondo). Poiché una parola è composta da 7 bit + 1 bit di start, 1 di stop ed 1 di parità, questo significa che in ASCII è possibile trasmettere da 12 a 1066 caratteri al secondo. In Baudot, invece, la velocità varia da 45 a 100 baud. Poiché la parola è composta da 5 bit + 1 bit di stop e 1 di start il numero di caratteri/secondo varia da 7 a 15. Come è possibile constatare il sistema baudot viene utilizzato su canali di comunicazioni a bassa velocità. Questo è un fattore molto importante nella trasmissione radio, poiché una velocità più bassa significa maggiore immunità ai disturbi (a parità di rumore sul canale di trasmissione). Inoltre, a parità di velocità, il sistema baudot risulta più efficiente dell'ASCII.

Dovendo trasferire i dati via radio è necessario convertire le variazioni elettriche tra gli stati logici 0 e 1 in una forma che ne consenta la trasmissione via canali audio. A questa operazione sono preposti i MODEM (modulator-demodulator), che convertono le variazioni di tensione in variazioni di frequenza. Allo stato logico '1' viene assegnata una certa frequenza audio, ad esempio 1200 Hz, mentre allo '0' ne viene assegnata un'altra, nello stesso esempio 2200 Hz. Lo scostamento tra le due frequenze (in questo caso 2200-1200 = 1000 Hz) viene definito Shift. Generalmente allo '0' viene assegnata la frequenza superiore e la trasmissione detta 'normale', in caso opposto, invece, la trasmissione viene detta 'reverse'. In RTTY

Comandi in ricezione

^ (or m)	REGISTRA MEMORIA DINAMICA 0... 9
^ B	BELL ABILITATO/DISABILITATO
^ C	IDENTIFICAZIONE MORSE ABILITATA/DISABILITATA
^ H	VISUALIZZA ELENCO COMANDI
^ I	IDLE INSERITO/DISINSERITO
^ Q	OSP INSERITO/DISINSERITO
^ S	SELEZIONE UOS/SOS
^ W	SALVA HOLD-BUFFER SU DISCO
F1	SET VELOCITÀ
F2	SET ORA/DATA
F3	PULISCE LO SCHERMO
F4	STATO HOLD BUFFER
F5	INTRODUZIONE NOMINATIVO CORRISPONDENTE
F6	VISUALIZZAZIONE/STAMPA HOLD BUFFER
F7	PASSAGGIO IN TRASMISSIONE
F8	VISUALIZZAZIONE STATO

Comandi in trasmissione

^ C	TRASMETTE CALLSIGN A CARATTERI GIGANTI
^ (or m)	TRASMETTE MEMORIA DINAMICA 0... 9
^ B	TRASMETTE FRASE INIZIO MESSAGGIO
^ C	GENERAZIONE CO AUTOMATICO
^ D	TRASMETTE DESCRIZIONE STAZIONE RADIO
^ E	TRASMETTE FRASE FINE MESSAGGIO E PASSAGGIO IN RX
^ F	TRASMETTE "THE QUICK BROWN FOX..."
^ O	TRASMETTE ORA E DATA (SE ABILITATI)
^ Q	TRASMETTE FILE ASCII RESIDENTE SU FLOPPY DISK
^ R	TRASMETTE "RYRYRY..."
^ W	SALVA HOLD BUFFER SU FLOPPY DISK
^ X	TRASMETTE "THIS IS A TEST TX FROM..."
F1	TRASMETTE MEMORIA PREDEFINITA 1
F2	TRASMETTE MEMORIA PREDEFINITA 2
F3	TRASMETTE MEMORIA PREDEFINITA 3
F4	TRASMETTE MEMORIA PREDEFINITA 4
F5	TRASMETTE MEMORIA PREDEFINITA 5
F6	TRASMETTE HOLD-BUFFER
F7	PASSAGGIO NORMALE IN RICEZIONE
F8	PASSAGGIO RAPIDO IN RICEZIONE

* Il simbolo '^' denota il tasto CTRL.

sono utilizzati due sistemi distinti di modulazione: a toni bassi (space a 1275 Hz) e a toni alti (space a 2125 Hz). Comunque l'esatta frequenza di Space ha una importanza relativa. Infatti la trasmissione viene effettuata sopprimendo l'onda portante per cui, agendo sulla sintonia fine del ricevitore, è possibile adattare al proprio modem i segnali in arrivo. È invece necessario che lo shift adottato dai due modem sia identico. Per questo parametro sono generalmente utilizzati tre diversi valori: 170Hz (radioamatori), 425Hz (agenzie di stampa), 850Hz (stazioni meteo) sebbene sia possibile, in qualche caso, trovare anche shift non standard. È pertanto consigliabile utilizzare modem che consentano la regolazione fine dello shift, per adattarsi alle necessità che dovessero presentarsi.

Per saperne di più è utile consultare le raccomandazioni CCITT V1, V3, V4, V21.

Interfacciamento

Il modem viene interfacciato, in maniera estremamente semplice, alla User Port. Poiché il Commodore 64 è dotato di interfaccia seriale a livello TTL (mark 5V, space 0V) è necessario che il modem sia dotato di un ingresso/uscita compatibile con questi livelli. Per consentire la commutazione ricezione/trasmissione sul pin 4 della porta, è presente un segnale a +5V durante la trasmissione che può essere utilizzato (tramite relé) per commutare ricevitore, trasmettitore e modem.

Per il collegamento delle apparecchiature si rimanda alle istruzioni fornite con il modem.

Attenzione

In qualche caso si sono verificate interferenze tra varie periferiche e l'interfaccia RS-232 incorporata nel C64. In particolare è possibile il verificarsi di errori di parità (in ricezione) proporzionalmente più frequenti quanto più alta è la velocità di ricezione. Queste interferenze sono purtroppo estremamente saltuarie e risulta molto difficile appurarne l'origine. Pare comunque accertato che il problema risieda nella capacità dei cavi di collegamento con il registratore a cassette CN2N. Nel caso dovessero presentarsi problemi di ricezione è quindi buona norma *scollegare il registratore dal C64*. Nella totalità dei casi investigati questa soluzione ha posto fine al problema.

Generalità del programma

TERM4 consente di emulare un terminale RTTY di tipo KSR (trasmittente/ricevente). In più incorpora molte caratteristiche accessorie che semplificano enormemente l'uso nella stazione del radioamatore. Le caratteristiche salienti sono:

- gestione di 5 memorie permanenti da 255 caratteri;
- gestione di 10 memorie (dinamiche) da 72 caratteri l'una predefinita in fase di installazione e modificabili durante l'uso;
- gestione di una memoria di accumulo di 24000 caratteri che consente, durante la

ricezione, di memorizzare i messaggi e, successivamente, di stamparli, ritrasmetterli o conservarli permanentemente su disco;

- possibilità di trasmettere file ASCII residenti su supporto magnetico;
- gestione dinamica di ora e data, con possibilità di trasmissione automatica a fine messaggio;
- identificazione morse;
- possibilità di selezionare la velocità di operazione in maniera continua da 25 a 110 bit/secondo;
- CQ, messaggi di inizio e fine passaggio automatici, con memorizzazione del corrispondente;
- RYRYRY... e THE QUICK BROWN FOX... pre-programmati;
- descrizione stazione radio e "big-id".

La gestione del programma è realizzata, in maniera abbastanza flessibile, attraverso l'uso di comandi di uno o due caratteri. Inoltre in qualsiasi momento premendo

CTRL-(HELP) è possibile visualizzare l'elenco completo dei comandi.

Il programma è logicamente suddiviso in tre sezioni principali: la prima cura la sezione ricevente, la seconda quella trasmittente mentre la terza viene utilizzata all'inizio per l'inizializzazione delle variabili. Le due sezioni principali (ricevente e trasmittente) sono organizzate entrambe con una routine principale di polling (che scandisce continuamente tastiera e porta RS-232), un "command-interpretor" molto semplificato ed una serie di routine che eseguono i singoli comandi. Molta parte ha avuto nella scelta di questa organizzazione l'aspetto velocità. Il Basic, infatti, è un linguaggio abbastanza lento per questo tipo di applicazione, direttamente dipendente dalla rapidità di esecuzione, e solo la bassa velocità intrinseca al sistema Baudot ha permesso l'implementazione di questo tipo di applicazione. La struttura, comun-

```

10 REM TERM-4.4 (BRUDOT)
20 REM REV.4 3/1/1984
30 REM (C) GIORGIO LEO RUTILTRAV
40 PRINT "DETTY TERMINAL EMULATOR"
50 PRINT "V4.4 (R2SE 11.1983) --M"
60 OPEN 1:2:0:CHR$(64+32)+CHR$(0)+CHR$(25)+CHR$(30)
70 POKE 56,60:POKE 50,60
80 POKE 53280,0:POKE 53281,0
90 POKE 50577,251:GOSUB 3700
100 PRINT "LA VELOCITA' E' "GO BRUDOT:
110 POKE 54276,33 FOR I=1050 NEXT POKE 54276,0
120 REM ***
130 REM *** SEZIONE RICEZIONE
140 REM ***
150 REM ***
160 REM *** POLLING RY
170 REM ***
180 POKE54276,0 GET#1,RR IF RR="" THEN 250
190 A=ASC(RR):IF A#27 THEN C=1:GOTO 250
200 IF A#11 THEN IF C AND BE THEN POKE 54276,33
210 IF A#31 THEN C#0:GOTO 250
220 PRINT CHR$(30)(C,R):IF SO AND A#4 THEN C#0
230 IF HS THEN A#A#1:IF H#2#1 THEN H#A#1:H#B#0
240 IF HS THEN POKE H#16000,30(C,R)
250 SET SE IF SE="" THEN 180
260 REM SERVICE RX
270 IF SE="" THEN C#0:GOTO 180
280 IF SE="F" THEN C#1:GOTO 180
290 IF SE="M" THEN 500
300 IF SE="B" THEN 550
310 IF SE="P" THEN PRINT "D":GOTO 180
320 IF SE="E" THEN 500
330 IF SE="W" THEN 180
340 IF SE="Z" THEN PRINT "OGGUS 3550"
350 IF SE="R" THEN 1430
360 IF SE="H" THEN 1200
370 IF SE="F" THEN 310
380 IF SE THEN BE#A#ASC(SE):IF A#47 AND A#50 THEN H#A#40:GOTO 1340
390 IF SE="Z" THEN H#F#1
400 IF SE="E" THEN 1360
410 IF SE="W" THEN 730
420 IF SE="M" THEN 500
430 IF SE="R" THEN 1180
440 IF SE="H" THEN 1000
450 IF SE="P" THEN GOSUB 3290:PRINT "D":
460 GOTO 180
470 REM ***
480 REM *** CAMBIO VELOCITA'
490 REM ***
500 PRINT
510 INPUT "MODORA VELOCITA' (BRUDOT):"SP
520 IF SP=C OR SP=118 THEN PRINT "VELOCITA' NON IMMESSA":GOTO 180
530 A=ASC(SP):POKE 665,R/256
540 POKE 665,R-PEE/665/R/256
550 PRINT "LA VELOCITA' E' "SP:BRUDOT:GOTO 180
560 REM ***
570 REM *** UOS/SOS
580 REM ***
590 SOUND(50):PRINT
600 IF SO THEN PRINT "BOS ON":
610 IF NOT SO THEN PRINT "BOS OFF":
620 GOTO 180
630 REM ***
640 REM *** SET ORA/DATA
650 REM ***
660 PRINT
670 TH#0:BE#1:PRINT "ORA ATTUALE "T#1
680 INPUT "NUOVO ORARIO (MMSS):"SE
690 IF LEN(SE)<6 THEN PRINT "ORA/DATA OFF":GOTO 180
700 IF SE="" THEN T#BE#T#1
710 SE="" INPUT "DATA (GGMMYY):"DM
720 PRINT "ORA/DATA ON":GOTO 180
730 REM ***
740 REM *** STATO BELL
750 REM ***
760 BE#NOT(BE):PRINT
770 IF BE THEN PRINT "BELL ON":
780 IF NOT BE THEN PRINT "BELL OFF":
790 GOTO 180
800 REM ***
810 REM *** SELEZIONE STATO HOLD
820 REM ***
830 PRINT
840 IF H#B# AND H# THEN INPUT "SE-CANCELLO L'HOLD? SE IF A#(C) THEN H#B#
850 IF H#B# AND A#(C) THEN H#B# PRINT "HOLD CANCELLATO"
860 IF H#B# THEN PRINT "HOLD ON":H#1:GOTO 180
870 H#B# PRINT "HOLD OFF":GOTO 180
880 REM ***
890 REM *** DISPLAY STATO
900 REM ***
910 PRINT PRINT "STATO "PRINT "BELL "SE#BE#0:PRINT "CHID "SE#H#B#0:PRINT "IDLE "
920 PRINT SE#(C#1)SE#(C#2)SE#(C#3)SE#(C#4)SE#(C#5)SE#(C#6)SE#(C#7)SE#(C#8)SE#(C#9)SE#(C#10)
930 PRINT "UOS "SE#(NOT(SO#0)):PRINT "HOLD "SE#(H#B#0):PRINT "Z"SE#(Z#1)SE#(Z#2)SE#(Z#3)SE#(Z#4)SE#(Z#5)SE#(Z#6)SE#(Z#7)SE#(Z#8)SE#(Z#9)SE#(Z#10)
940 H#1:PRINT "HOLD PIENO":
950 PRINT TAB(10)"ORA/LETTA":T#1,2:PRINT TAB(10)"DATA":DM,2:DM
960 PRINT "D":GOTO 180
970 REM ***
980 REM *** IDLE ON/OFF
990 REM ***
1000 I#(NOT(I#)):PRINT
1010 IF I# THEN PRINT "IDLE ON":
1020 IF NOT I# THEN PRINT "IDLE OFF":
1030 GOTO 180
1040 REM ***
1050 REM *** DUMP HOLD
1060 REM ***
1070 PRINT
1080 IF H#B# THEN PRINT "HOLD-BUFFER NUOTO":GOTO 180
1090 OPEN "A:"
1100 C#"" INPUT "ELETTORE":CA
1110 IF NOT(C#="S" OR C#="B") THEN 1140
1120 PRINT#4,"":IF ST#0 THEN PRINT "PRINTER CHECK":GOTO 1140
1130 C#0:4
1140 A#0:B#0:IF H#1 THEN A#A#1:B#B#1
1150 FOR I=1 TO 3:A#A#1:IF A#H# THEN A#B#
1160 PRINT CHR$(PEE/665/R/256):NEXT I
1170 IF C#="V" OR C#="YES" THEN PRINT#4,"":
1180 PRINT "D":CLOSE#4:GOTO 180
1190 REM ***

```

que, è stata anche condizionata dalla necessità di creare un prodotto lineare, semplice da leggere e quindi da modificare ed espandere.

L'occupazione generale di memoria è di circa 11KByte di programma (44 settori su disco) e circa 2KByte di dati.

Il resto della memoria è utilizzato dall'Hold Buffer, il cui inizio è posto alla locazione 16000.

Questa memoria di accumulo ha una capacità di 24000 caratteri. Poiché vengono memorizzati i caratteri effettivamente visualizzati sul display (in ASCII) l'effettiva capacità di memorizzazione corrisponde a minimo 52 minuti di ricezione a 50 baud. Nel caso l'Hold-buffer dovesse riempirsi, il contenuto della memoria viene salvaguardato. I nuovi dati vengono aggiunti al buffer mentre dalla cima della memoria vengono scartati i byte in eccesso. Con questo sistema in caso di overflow del buf-

fer vengono comunque conservati gli ultimi dati di memoria.

Installazione

L'installazione del programma è estremamente semplice e richiede pochissimo tempo. È sufficiente caricare in memoria il programma di "distribuzione" TERM44D e modificare alcuni parametri:

1) Assegnare alla stringa ZF\$ il proprio nominativo, preceduto da uno spazio.

2) Assegnare alla stringa CW\$ l'equivalente del proprio nominativo in codice morse. Nella stringa i punti sono rappresentati da "1", le linee da "3" e gli intervalli da spazio. Il mio nominativo (I8ZSE) diventa quindi "11 33311 3311 111 1".

3) Assegnare alle cinque stringhe dell'array MS\$ il contenuto desiderato, considerando che il ritorno carrello è memorizzato nella stringa RS.

4) Assegnare ai DATA valori di inizia-

lizzazione delle dieci stringhe dell'array MES, in cui sono memorizzati i contenuti delle dieci memorie dinamiche M0... 9

5) Assegnare ai DATA il messaggio contenente la descrizione della stazione radio, lasciando il primo inalterato (stringa = null). Nel caso la lunghezza differisse dalle nove linee, variare il contenuto della variabile "DE", indicando l'effettivo numero di linee costituenti il messaggio.

6) Variare le linee contenenti il nominativo gigante.

7) Memorizzare la versione personalizzata.

È inoltre possibile variare lo stato iniziale dei parametri modificabili da tastiera (bell, id morse, idle, sos, qsp). Per fare ciò è sufficiente variare nella sezione inizializzazione i dati assegnati alle variabili be (be), id (idle), cw (id morse), so (sos) e qs (qsp). Il valore "0" equivale a comando disabilitato, "-1" a comando abilitato. Non assegnare altri valori altrimenti diverrebbe impossibile modificare il parametro da tastiera.

Comandi eseguibili durante la ricezione

* nota: in tutti i comandi il simbolo \wedge denota il tasto CTRL

\wedge @ < numero della memoria >

Questo comando permette di modificare il contenuto di una delle memorie dinamiche 0...9. Può essere utilizzato in qualsiasi momento durante la ricezione. Il contenuto assegnato non può eccedere i 77 caratteri ed è terminato da un RETURN. Inoltre la stringa non deve contenere virgole e "due punti", che vengono interpretate come separatori e generano il messaggio di errore "EXTRA IGNORED". Rispondendo solo "RETURN" il contenuto della memoria non viene alterato. È possibile posizionare il cursore su una parte del display, editare la linea con i comandi appropriati e assegnarla ad una memoria.

\wedge B

Questo comando consente l'abilitazione e la disabilitazione dell'avvisatore acustico (Bell) che viene comandato dalla ricezione del corrispondente carattere Baudot. Sullo schermo questo carattere viene comunque visualizzato da un simbolo simile ad una campanella. L'avvisatore è inizialmente disabilitato. Successivi comandi \wedge B variano lo stato dell'avvisatore acustico. È possibile variare lo stato iniziale modificando l'inizializzazione della variabile BE (vedi installazione).

\wedge C

Questo comando consente l'abilitazione e la disabilitazione dell'identificazione morse a fine messaggio, che genera automaticamente la trasmissione di "CW ID:" seguito dal nominativo in codice morse. La trasmissione del nominativo in "chiaro" è prevista dalla regolamentazione legislativa dell'attività dei radioamatori, per cui lo stato iniziale di questo parametro è di abilitato.

\wedge H

Questo comando genera la visualizzazione sullo schermo di un elenco sommario

```

1200 REM *** DSP ON/OFF
1210 REM
1220 GOSUB 1000:PRINT
1230 IF OS THEN PRINT "DSP ON:"
1240 IF NOT OS THEN PRINT "DSP OFF:"
1250 GOTO 180
1260 REM
1270 REM *** NOMINATIVO CORRISP.
1280 REM
1290 PRINT INPUT "CORRISP.":CS#
1300 PRINT "OK" GOTO 180
1310 REM
1320 REM *** MESSAGGI 11-110
1330 REM
1340 PRINT PRINT "MESSAGGIO #":(MC: 70: CRR#)
1350 INPUT "RE:":R# PRINT "OK" GOTO 180
1360 REM
1370 REM *** STATO IDENTIFIC.
1380 REM
1390 PRINT "ID":ID
1400 IF NOT ID THEN PRINT "ID MORSE ON:"
1410 IF NOT ID THEN PRINT "ID MORSE OFF:"
1420 GOTO 180
1430 REM
1440 REM *** SEZIONE TRASMISSIONE
1450 REM ***
1460 PRINT "ID":ID
1470 FOR I=1 TO 5:PRINT "M":M(I)
1480 REM
1490 REM *** POLLING TX
1500 REM
1510 FOR I=1 TO 10:PRINT "S"
1520 PRINT "S"
1530 GET #1:IF #1 THEN PRINT CHR$(20):GOTO 1560
1540 IF PEEK(673) AND 1:GOTO 1570
1550 PRINT#1:CHR$(0):GOTO 1530
1560 #PRC:R#-23:IF #R# THEN PRINT#1:CHR$(0)+CHR$(2):PRINT GOTO 1510
1570 IF # THEN #R#-1:IF #R# AND #Z THEN #R#-(R#-15):GOTO 3240
1580 IF #R#-1 THEN PRINT#1:CHR$(4):PRINT " ":GOTO 1530
1590 IF #R# OF #R# THEN #R#-1
1600 IF #R# AND #R# THEN #R#-1
1610 IF TC:CHR$(R#) THEN TC:CHR$(R#):PRINT#1:CHR$(TC):)
1620 PRINT#1:CHR$(R#):R#):PRINT #R#:GOTO 1530
1630 REM TX SERVICE
1640 IF #R# THEN 1670
1650 IF #R# THEN 2290
1660 IF #R# THEN 2270
1670 IF #R# THEN 2060
1680 IF #R# THEN 2650
1690 IF #R# THEN 2740
1700 IF #R# THEN 2910
1710 IF #R# THEN GOSUB 2890
1720 IF #R# THEN 1510
1730 IF #R# THEN 2590
1740 IF #R# THEN 2670
1750 IF #R# THEN #R#-(R#):GOTO 3240
1760 IF #R# THEN #R#-(1):GOTO 3240
1770 IF #R# THEN #R#-(2):GOTO 3240
1780 IF #R# THEN #R#-(3):GOTO 3240
1790 IF #R# THEN #R#-(4):GOTO 3240
1800 IF #R# THEN #R#-(5):GOTO 3240
1810 IF #R# THEN PRINT "M":M# GOTO 1530
1820 IF #R# THEN 2470
1830 IF #R# THEN GOSUB 2390:PRINT "M":
1840 IF #R# THEN GOSUB 3420:PRINT "M":
1850 IF #R# THEN GOSUB 3500:PRINT "M":
1860 GOTO 1530
1870 REM
1880 REM *** AVVVISORE IN RICEZIONE
1890 REM
1900 GOSUB 1950
1910 IF PEEK(673) AND 1 THEN 1910
1920 FOR I=1 TO 15
1930 POKE 5679,I
1940 PRINT "RICEZIONE":GOTO 180
1950 REM
1960 REM *** IDENTIFICAZIONE MORSE
1970 REM
1980 IF TX THEN GOSUB 2320
1990 IF # THEN RETURN
2000 #R#-(R#-10):ID="GOSUB 2110
2010 IF PEEK(673) AND 3 THEN 2010
2020 #R#-(5275):POKE 5676,175:R#-80
2030 FOR I=1 TO LEN(R#):#R#-(CHR$(I)):
2040 FOR J=1 TO I:NEXT
2050 IF #R# THEN #R#-(1):GOTO 2070
2060 POKE 5676,151
2070 FOR J=1 TO I:PRINT#1:CHR$(J):
2080 POKE 5676,155
2090 NEXT
2100 POKE 5676,R# PRINT RETURN
2110 REM
2120 REM *** TRASMISSIONE STRINGA
2130 REM
2140 #R#-(R#-1) TO LEN(R#):#R#-(R#):I=1
2150 IF #R# THEN GET #1:CHR$(R#):CHR$(0):IF #R# THEN I-999
2160 #R#-(R#):IF #R# THEN PRINT#1:CHR$(0)+CHR$(2):PRINT GOTO 2220
2170 IF #R# THEN PRINT#1:CHR$(4):PRINT " ":GOTO 2220
2180 IF #R# OF #R# THEN 2220
2190 #R#-(23):IF #R# AND #Z THEN 2220
2200 IF TC:CHR$(R#) THEN TC:CHR$(R#):PRINT#1:CHR$(TC):)
2210 PRINT#1:CHR$(R#):R#):PRINT #R#
2220 IF #R#-(R#-1):PEEK(673) AND 1 THEN 2220
2230 NEXT I:RETURN
2240 REM ***
2250 REM *** ROUTINE VARIE
2260 REM *** SEZIONE TRASMISSIONE
2270 REM ***
2280 REM
2290 REM *** QUICK BROWN (CTRL-F)
2300 REM
2310 #R#-(R#-1) TO LEN(R#):#R#-(R#):I=1
2320 #R#-(R#):
2330 GOSUB 2110:R#
2340 IF #R# THEN 2330
2350 GOTO 1560
2360 REM
2370 REM *** BV (CTRL-T)
2380 REM

```

(continua a pag. 84)

