Torniamo ad occuparci questo mese del software per il PC-1211, con tre programmi di carattere molto eterogeneo fra loro, e per questo rivolti ad un pubblico "assortito".

Si tratta del calcolo dei parametri per la realizzazione di un multivibratore astabile a transistor, inviatoci dal lettore Fabrizio Fabiani di Napoli; un altro programma per trasformare il PC-1211 in un orologio con sveglia, di Andrea Casali, ed infine Fabrizio D'Amore ci propone il suo lavoro effettuato allo scopo di calcolare le n radici complesse di un'equazione algebrica di grado n.

Multivibratore astabile

di Fabrizio Fabiani - Portici (NA)

Questo programma consente il dimensionamento dei componenti di un multivibratore astabile a transistor per una data frequenza di lavoro. Prima di analizzare il programma, diamo un breve cenno sul funzionamento di tale circuito (vedi fig. 2). Il multivibratore astabile è un dispositivo in grado di generare una forma d'onda rettangolare, disponibile su uno dei collettori dei transistor. A prima vista il circuito si presenta come due amplificatori R-C connessi con l'uscita del primo all'ingresso del secondo e viceversa; ciò produce un fenomeno di instabilità che provoca la commutazione alternativa dei due transistor dall'interdizione (OFF) alla saturazione (ON). La frequenza di oscillazione è determinata dalla costante di tempo associata ai condensatori ed alle resistenze di base secondo la relazione:

$$f = 1/(1.38\tau)$$

con:

$$\tau = R_b \cdot C$$

La massima frequenza di funzionamento è limitata dai tempi di salita dell'onda rettangolare, e per ottenere un buon andamento della stessa, è conveniente non superare i 30 kHz.

Il programma è piuttosto semplice e pratico ed è stato strutturato in modo da evitare errori nella fase di immissione dei dati: infatti esso non accetta valori di frequenza superiori a 30 kHz o tensioni di alimentazione Vcc superiori alla massima Vce del transistor.

Inoltre, nella fase di visualizzazione dei dati, il programma sceglie automaticamente l'unità di misura del condensatore (picofarad o microfarad).

Osservando le linee 35 e 45, le quali contengono i due test sulla frequenza e su Vcc, possiamo trovare un metodo molto interessante di indirizzamento controllato. Ad esempio nella linea 35 se la frequenza immessa è maggiore di 30.000 o minore di 0 (i due casi non potranno sussistere contemporaneamente!), il programma salta alla linea 30 richiedendo un nuovo valore, altrimenti prosegue alla 40. Lo stesso metodo di indirizzamento è impiegato nella linea 60 per stabilire l'unità di misura dei condensatori.

Un'altra possibilità che offre il programma è di poter inserire le caratteristiche del transistor usato sia da tastiera, sia assegnare automaticamente ad esso le caratteristiche di un BC108 standard (Vcesat = 0.25 V, Ibsat = 0.13 mA, Vcemax = 20 V, Icsat = 45 mA).

Il programma gira in modo DEF con SHIFT M e può funzionare sia con la stampante, che senza di essa.

```
 5:"M"CLEAR
10:INPUT "VUOI
   USARE IL BC1
   08?";A$
15:IF A$="SI"
   GOTO 25
16:IF A$="NO"
   GOTO 20
17:GOTO 10
20:INPUT "VCE S
   AT(V)=";A:
   INPUT "ICSAT
   (MA)=";B:
   INPUT "IBSAT
   (MA)=";C
21:B=B*1E-3:C=C
   *1E-3
22:INPUT "VCE M
   AX(V)=";D:
   GOTO 30
25:A=.25:C=.13E
   -3:D=20:B=45
   E-3
30:INPUT "FREQU
   ENZA(HZ)=";F
35:GOTO (40-10*
   ((F>30000)+(
   F<0)))
40:INPUT "VCC(V
   )=";E
45:GOTO (50-10*
   ((E>D)+(E<=0
   )))
50:G=(E-A)/B:C=
   C+300E-6
51:H=((E-.7)/C)
   :L=H*1E-3
55:LET I=1/(1.3
   8*F*H)
56:PRINT "RC(OH
   M)=",USING "
   #####";G:
   PRINT "RB(KO
   HM)=",USING
   "###.#";L
60:GOTO ((I>E-7
   )+61
61:M=I*E12:
   PRINT "C(PF)
   =",USING "##
   ####";M:END
62:M=I*E6:PRINT
   "C(MICROF)="
   ,USING "####
   #.#";M:END
```

Figura 1 - Listing del programma "Multivibratore astabile".

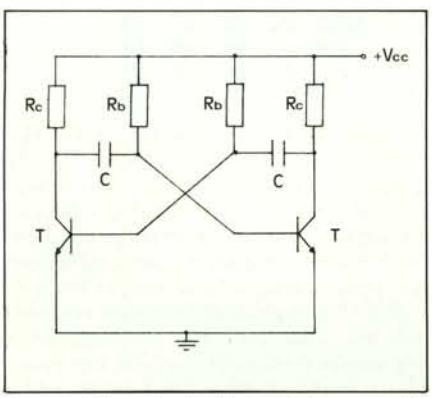

Figura 2 - Schema elettrico di un multivibratore astabile a transistor.

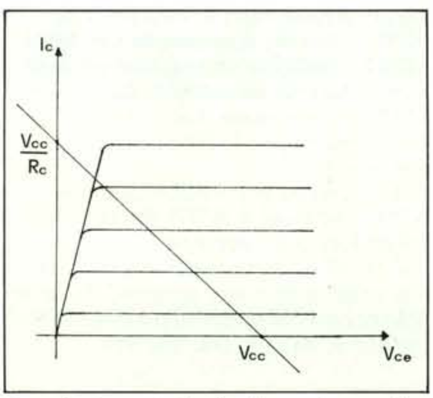

Figura 3 - Caratteristiche di collettore di un transistor con relativa retta di carico.

Orologio con sveglia

di Andrea Casali - Pavia

Come indicato dal titolo, questo programma svolge le funzioni di un orologio, con possibilità di segnalazione d'allarme programmata. Per il calcolo dell'ora, la routine si avvale del tempo impiegato dalla macchina per eseguire le istruzioni Basic, per cui si consiglia vivamente di non modificare nulla del listing, per evitare un errato conteggio del tempo.

Il programma inizia a girare premendo SHIFT A, e verrà subito richiesto se si desidera o no la segnalazione d'allarme. Ad una risposta positiva seguirà la richiesta d'impostazione dell'ora d'allarme. A questo punto va inserita l'ora attuale (OROLOGIO ORA? e MINUTI?), quindi il programma potrà iniziare l'elaborazione.

Il PC-1211 inizierà a lampeggiare l'ora ed i minuti, mentre allo scoccare di ogni ora suonerà un Beep. La segnalazione d'allarme è costituita da 19 Beep consecutivi e si consiglia di non modificare tale numero, pena la perdita della precisione dell'orologio.

Ovviamente il PC-1211 andrà posizionato nel modo DEF e per uscire dal programma occorrerà premere il tasto BREAK.

Radici complesse di equazioni algebriche

di Fabrizio D'Amore - Roma

Tra gli algoritmi per il calcolo delle n radici complesse di un'equazione algebrica di grado n (a coefficienti reali), uno dei migliori è senza dubbio quello di Lin-Bainstrow che unisce la rapidità di convergenza (quadratica) alla semplicità. Com'è noto, questo metodo approssima i coefficienti di un divisore quadratico di p(z), il polinomio dato; calcola, cioè, in modo iterativo, i coefficienti p e q del trinomio $d(z) = z^2 +$

pz + q in modo tale che la divisione p(z)/d(z) fornisca resto identicamente nullo. La convenienza della determinazione di d(z) sta nel fatto che i suoi zeri sono anche zeri di p(z) e cioè le radici dell'equazione p(z) = 0.

```
 10:"A"INPUT "VU
    OI L^ALLARME
    ?(S) (N)",L$
    :H=99:I=99
 12:IF L$="N"
    GOTO 18
 15:INPUT "ALLAR
    ME ORA?",H:
    INPUT "MINUT
    I?",I
 18:INPUT "OROLO
    GIO ORA?",A:
    INPUT "MINUT
    I?",B:C=0:D=
    0
 20:PAUSE A,B:C=
    C+1:IF C=29
    GOTO 100
 30:IF C=1GOTO 4
    00
 35:GOTO 20
100:C=0:B=B+1:IF
    B=60GOTO 150
110:PAUSE A,B:
    GOTO 200
150:BEEP 1:B=0:A
    =A+1:PAUSE A
    ,B
200:PAUSE A,B:C=
    C+1:IF C=31
    GOTO 300
205:IF C=1GOTO 4
    00
207:GOTO 200
300:C=0:B=B+1:IF
    B=60GOTO 350
310:PAUSE A,B:
    GOTO 20
350:BEEP 1:B=0:A
    =A+1:PAUSE A
    ,B:GOTO 20
400:IF A<>HGOTO
    200
410:IF B=IGOTO 4
    20
415:GOTO 200
420:BEEP 19:GOTO
    500
500:PAUSE A,B:C=
    C+1:IF C=28
    GOTO 100
510:GOTO 500
```

Figura 4 - Listing del programma "Orologio con sveglia".

```
 10:"B"CLEAR :
    USING "###"
 20:INPUT "GRADO
    =";N
 30:IF N<2THEN 2
    0
 40:O=27:FOR W=0
    TO N
 50:PAUSE W;"":
    INPUT A(O):O
    =O+1:NEXT W:
    A=A(27):GOTO
    70
 60:BEEP 1:GOTO
    50
 70:"J"I=0:USING
    "^":INPUT "P
    0=";P
 80:INPUT "Q0=";
    Q
 90:I=I+1,B=A,C=
    A,E=0,G=0,O=
    28:FOR W=1TO
    N-1
100:D=E,F=G,E=B,
    G=C,B=A(O)-E
    P-DQ:C=B-GP-
    FQ:O=O+1:
    NEXT W
110:D=E,E=B,B=A(
    O)-EP-DQ:Y=(
    BG-CE)/(GG-C
    F):X=(E-FY)/
    G:R=P,S=Q
120:P=P+X,Q=Q+Y,
    R=P-R,S=Q-S:
    IF (ABS R>L)
    +(ABS S>L)
    PAUSE R,S:
    GOTO 90
130:BEEP 1:PRINT
    USING ;"P=";
    P:PRINT "Q="
    ;Q:IF N>2
    PRINT USING
    "###";"ITERA
    TE=";I:USING
140:U=-P/2:V=UU-
    Q:IF ABS V<E
    -8PRINT "DOP
    PIA: ";U:
    GOTO 170
150:IF VLET H=U+
    √V,K=U-√V:
    PRINT "Z1=";
    H:PRINT "Z2=
    ";K:GOTO 170
160:V=√-V:PRINT
    "RE=";U:
    PRINT "IM=";
    V
170:IF N=2END
180:B=A,C=A,E=0,
    G=0,N=N-2,O=
    28:FOR W=0TO
    N-1
190:D=E,E=B,B=A(
    O)-EP-DQ,A(O
    )=B,O=O+1:
    NEXT W
200:IF N>2THEN 7
    0
210:P=A(28)/A:IF
    N=2LET Q=A(2
    9)/A:GOTO 13
    0
220:T=-P:PRINT "
    Z=";T:END
```

Figura 5 - Listing del programma "Radici complesse".

```
P=-1.819760245E-        P=1.567789765E-0
01                      1
Q=6.622091724E-0        Q=-9.611963492E-
1                       01
ITERATE=  6             ITERATE=  8
RE=9.098801225E-        Z1=9.051455956E-
02                      01
IM=8.086596033E-        Z2=-1.061924572
01                      P=4.025197049
                        Q=-1.571059725
                        Z1=0.358395475
                        Z2=-4.383592525
```

Figura 6 - Risultati dell'esempio citato nel testo.

Una volta determinata la prima coppia di zeri (reali distinti, reali coincidenti o complessi coniugati) si effettua la divisione p(z)/d(z) e si applica di nuovo il metodo al polinomio risultante finché non si giunge ad un'equazione di grado $\leqslant 2$. L'algoritmo si basa sulla ripetizione (per un numero di volte non noto a priori) di un particolare ciclo di calcolo (iterata) che fornisce ogni volta delle approssimazioni di p e q. Se all'i-esima iterata (i = 1, 2, 3...) tali approssimazioni sono p_i e q_i e $p(z) = a_0z^n + a_1z^{n-1} + ... + a_{n-1}z + a_n = 0$ allora i valori $\triangle p$ e $\triangle q$ forniti dal seguente sistema lineare:

$$C_{n-2}\triangle p + C_{n-3}\triangle q = b_{n-1}$$
$$C_{n-1}\triangle p + C_{n-2}\triangle q = b_n$$

con

$$b_k = a_k - p_i b_{k-1} - q_i b_{k-2},$$
$$c_k = b_k - p_i c_{k-1} - q_i C_{k-2},$$

e

$$b_{-2} = b_{-1} = 0,\ K = 0, 1, 2, \ldots n$$
$$C_{-2} = C_{-1} = 0,\ K = 0, 1, 2, \ldots n-1$$

vanno a correggere le precedenti approssimazioni: $p_{i+1} = p_i + \triangle p$; $q_{i+1} = q_i + \triangle q$; i = 1, 2, 3... È chiaro che sarà necessario fornire, magari cercandoli a lume di naso, i valori p_0 e q_0, approssimazioni iniziali di p e q. La convergenza è assicurata se tali approssimazioni iniziali sono "abbastanza vicine" ai valori reali. Il procedimento termina al j-esimo passo quando $p_j = p_{j-1}$ e $q_j = q_{j-1}$.

Per inserire i dati si preme SHIFT B; ciò provoca la visualizzazione dell'indice del coefficiente da inserire. In pratica bisogna digitare prima il grado e quindi, in sequenza, i valori a_0, a_1... a_n; vengono poi richieste le approssimazioni iniziali p_0 e q_0.

Al termine di ogni iterata sono mostrati gli ordini di grandezza di $\triangle p$ e $\triangle q$ che dovranno tendere a zero. Se da un esame dei valori mostrati dopo 4 o 5 iterate non si nota una soddisfacente convergenza a zero, è consigliabile interrompere l'elaborazione, premere SHIFT J ed inserire delle nuove approssimazioni iniziali. Se dopo un certo numero di iterate i valori di convergenza mostrati sono molto piccoli ma persistono in un non miglioramento, è consigliabile cambiare le approssimazioni iniziali oppure ci si può accontentare di una minore precisione inserendo in L un valore molto basso (ad esempio 10^{-7}).

Vediamo una sommaria descrizione del programma:

Linee 10/60: lettura dati; la riga 60 occorre nel caso che durante l'input si prema inavvertitamente il tasto Enter senza aver inserito il dato;

Linee 70/80: lettura approssimazioni iniziali; azzeramento del contatore di iterate;

Linee 90/100: inizializzazioni varie e calcolo dei b_i e c_i;

Linee 110/120: calcolo di b_n, $\triangle p$, $\triangle q$; correzione di p_i e q_i; test di convergenza con eventuale messaggio; il test è fatto non direttamente sui $\triangle p$ e $\triangle q$ calcolati, ma sulle differenze $|p_i - p_{i+1}|$ e $|q_i - q_{i+1}|$ che, a causa degli arrotondamenti nei calcoli, non coincidono necessariamente con $|\triangle p|$ e $|\triangle q_x|$;

Linea 130: visualizzazione di p e q;

Linee 140/160: calcolo e visualizzazione delle radici di d(z);

Linea 170: test di fine;

Linea 180/190: calcolo degli a_i, coefficienti del polinomio quoziente;

Linea 200: test per controllo necessità di altri cicli per il calcolo di p e q;

Linee 210/220: calcolo di p e q nel caso n = 2; calcolo e visualizzazione della soluzione se n = 1.

In figura 6 è rappresentato il printout della stampante per il calcolo delle radici del polinomio:

$$p(z) = z^6 + 4z^5 - 2z^4 - z^3 + z^2 - 3z + 1 = 0$$

con approssimazioni iniziali $p_0 = 0$ $q_o = 1$ e successivamente $p_0 = 0$ $q_0 = 1$. **MC**