LIBRERIA PER LA GESTIONE DELLE PORTE SERIALI (RS232/TTL) SOTTO CP/M E MANIPOLAZIONE DATI

S. Cerri

B4-40 (1987)

Libreria per la gestione delle porte seriali (RS232/TTL) sotto CP/M e manipolazione dati

Sirio Cerri

Istituto di Elaborazione della Informazione del CNR - Pisa

La presente libreria, studiata per il pilotaggio dello scanner ad ultrasuoni ETIS-1 tramite un MINUS KYBER con CP/M, nell'ambito della collaborazione tecnico-scientifica fra l'Istituto di Elaborazione della Informazione e la Societa` Aeritalia G.V.T., permette la gestione delle porte seriali RS232 e TTL richiamabili da Basic sotto CP/M con cui il MINUS controlla il sistema ETIS-1 e il collegamento sotto MPX-32 con il GOULD per la gestione delle periferiche di cui quest'ultimo e` dotato.

Nella fig. 1 e` schematizzato l'hardware utilizzato.

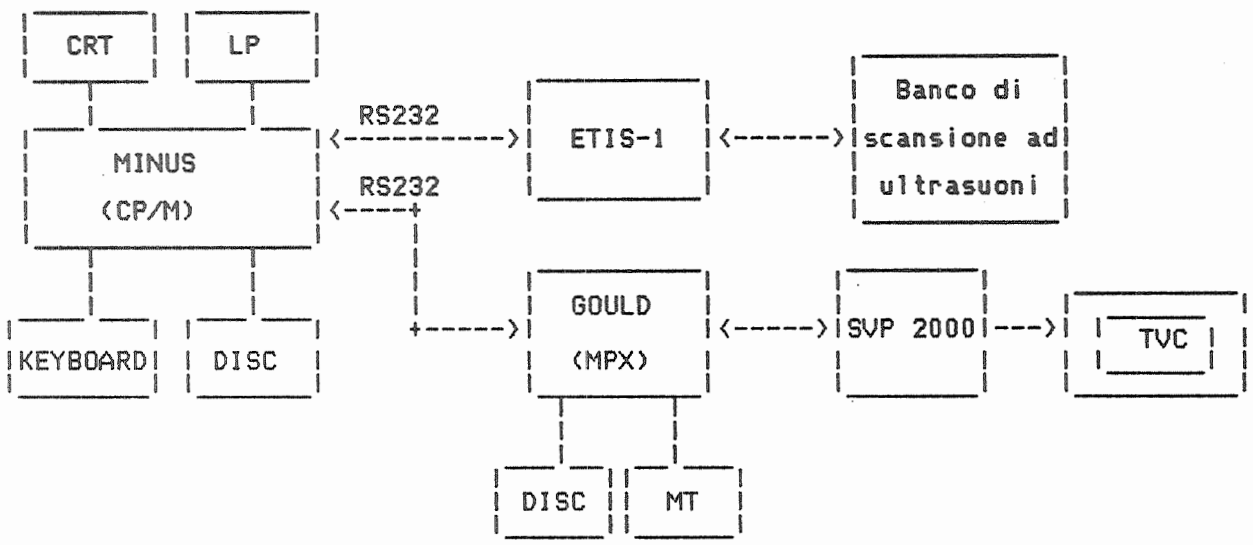


Fig. 1

Le routine, scritte il linguaggio macchina, hanno i seguenti compiti:

INIZ - inizializzazione e configurazione di un canale seriale
BRWV - lettura o scrittura di vettori sul canale seriale
BRWS - lettura o scrittura di stringhe sul canale seriale.

INIZ - Questa routine ha il compito di inizializzare e configurare uno dei due canali seriali A o B ai parametri attualmente previsti come segue:

19200 bauds, 8 bits, 1 stop bit, no parita`.

Questi parametri sono fissati da una tabella in coda alla libreria, ma per soddisfare altre esigenze di configurazione, essi possono essere variati facilmente con delle semplici Poke da programma basic.

La chiamata da programma basic e` cosi` strutturata:

```
CALL INIZ(P1,P2)
```

dove:

P1 (integer*2) e` l'indirizzo del canale comandi che si vuole inizializzare, nel nostro caso tale indirizzo e` 42 esadecimale (66 decimale) per il canale A, e 43 esadecimale (67 decimale) per il canale B.

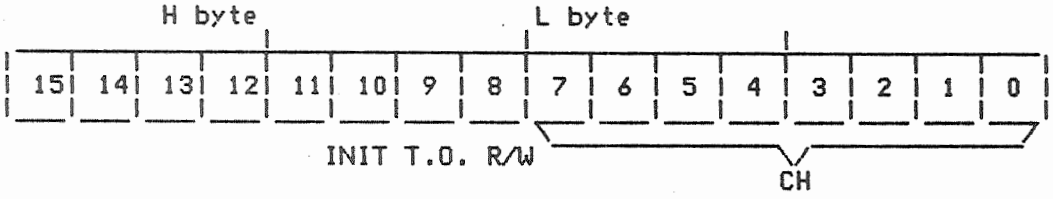
P2 (integer*2) e` il parametro di ritorno in risposta alla richiesta di inizializzare ed assumera` il valore 0 se l'operazione e` stata effettuata correttamente, ed il valore 2 se vi e` stato un errore nell'indirizzamento del canale.

BRWV - Effettua l'interfacciamento tra il programma basic e le routine vere e proprie di lettura o scrittura di vettori sui canali seriali. La chiamata e` cosi` strutturata:

```
CALL BRWV(P1,P2(I),P3(K))
```

dove:

P1 (integer*2) definisce il tipo di richiesta



bit

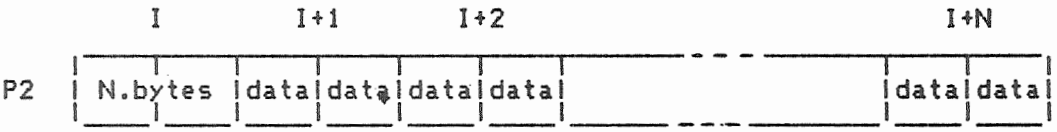
0-7 - indirizzo del canale dati su cui si vuole effettuare l'operazione di I/O.
Il canale A e` situato all'indirizzo 40H
Il canale B e` situato all'indirizzo 41H

8 - operazione da effettuare sul canale CH.
= 0 richiesta di lettura
= 1 richiesta di scrittura

9 - tipo di interruzione richiesta.
= 0 abilita l'interruzione della richiesta di I/O in corso dalla consolle con CTRL-G
= 1 abilita l'interruzione della richiesta di I/O in corso per time out.

10 - inizializzazione del canale.
= 0 NOP
= 1 viene inizializzato il canale su cui si vuole effettuare l'operazione di I/O

P2(I) (integer*2) - buffer dei dati da trasferire dimensionato nel programma chiamante a P2(I)+1 per un max di 32767 bytes. Nei primi due bytes di P2(I) deve essere contenuto il numero di bytes (N) da trasferire a partire da P2(I+2).



P3(K) (integer*2) 2 words - Flag di ritorno con informazioni sull'esito dell'operazione richiesta.

word 1: = 0 operazione completata correttamente
= 1 operazione incompleta per avvenuto time out
= 2 operazione non eseguita per indirizzo di canale errato
= 3 parametro illegale: P2(I) <= 0
= 7 operazione incompleta per interruzione da consolle (e` stato premuto CTRL-G).

word 2: contiene il numero di bytes trasferiti; se word 1=0 conterra` il numero di bytes richiesti P2(I); se word 1=1 o 7 conterra` il numero di bytes trasferiti fino alla richiesta di interruzione; se word 1=2 o 3 il contenuto sara` uguale a 0.

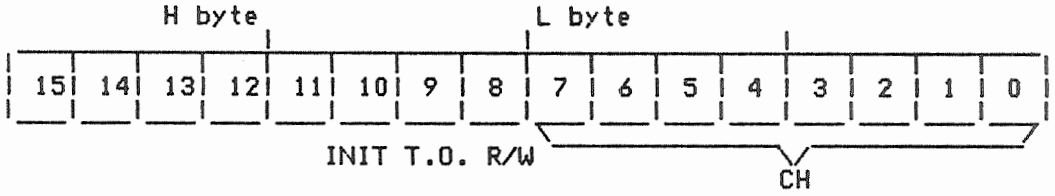
Questa routine fara` uso di altri sottoprogrammi individuati dal tipo di richiesta che vedremo in seguito.

BRWS - Effettua l'interfacciamenro tra il programma basic e le routine vere e proprie di lettura o scrittura di stringhe sui canali seriali. La chiamata e` cosi` strutturata:

CALL BRWS(P1,P2,P3)

dove:

P1 (integer*2) definisce il tipo di richiesta



bit

0-7 - indirizzo del canale dati su cui si vuole effettuare l'operazione di I/O.
Il canale A e` situato all'indirizzo 40H
Il canale B e` situato all'indirizzo 41H

8 - operazione da effettuare sul canale CH.
= 0 richiesta di lettura
= 1 richiesta di scrittura

9 - tipo di interruzione richiesta.
= 0 abilita l'interruzione della richiesta di I/O in corso dalla consolle con CTRL-G
= 1 abilita l'interruzione della richiesta di I/O in corso per time out.

10 - inizializzazione del canale.
= 0 NOP
= 1 viene inizializzato il canale su cui si vuole effettuare l'operazione di I/O

P2 (stringa) - stringa di caratteri da trasferire

P3 (intero su 2 bytes) - flag di ritorno con informazione sull'esito dell'operazione richiesta.

= 0 operazione completata correttamente
= 1 operazione incompleta per avvenuto time out
= 2 operazione non eseguita per individuazione di canale errato
= 3 errore richiesta di scrittura: la stringa e` vuota
= 7 operazione incompleta per richiesta di interruzione da consolle (e` stato premuto CTRL-G).

Con P3 = 1 o 7 si puo` conoscere il numero di caratteri trasferiti fino al momento della richiesta di interruzione usando la funzione basic LEN.

BRWS come BRWV, fa uso di altre routines individuate dal tipo di richiesta che esaminiamo rapidamente:

BIN - routine interna per la lettura di vettori dai canali seriali utilizzata da BRWV.

BOUT - routine interna per la scrittura di vettori o stringhe sui canali seriali utilizzata da BRWV e BRWS.

RCAR - routine interna per la lettura di stringhe dai canali seriali utilizzata da BRWS.

La parte di libreria che ora esamiamo e` specifica di operazioni legate alle funzioni del sistema ETIS-1 e del sistema SVP 2000. Queste routines sono state scritte in linguaggio macchina per velocizzarne l'esecuzione.

PAK - Utilizzata per compattare dati su 16 bits in dati su 8 bits. Ovviamente i valori su 16 bits devono essere compresi tra 0 e 255.

Tipica chiamata:

```
CALL PAK(P1(I),P2(K),P3)
```

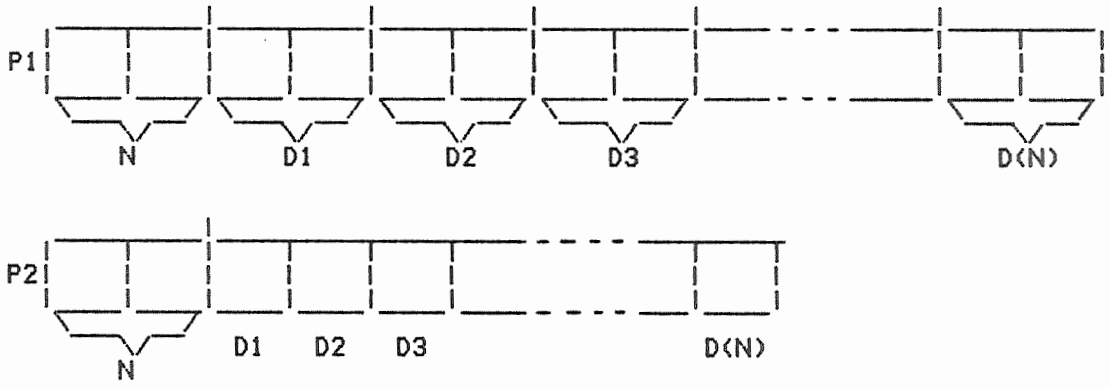
dove:

P1(I) vettore dei dati su 16 bits: nei due bytes indirizzati da P1(I) deve essere contenuto il numero di dati N da compattare (max 32767)

P2(K) vettore dei dati su 8 bits: nei primi due bytes sara` contenuto il numero dei dati compattati N.

P3 Flag di ritorno con informazioni sull'esito della chiamata:
= 0 operazione conclusa correttamente
= 3 errore: parametro N <= 0

Es.



SPAK - Utilizzata per espandere dati da 8 bits in dati a 16 bits.

Tipica chiamata:

```
CALL SPAK(P1(I),P2(K),P3)
```

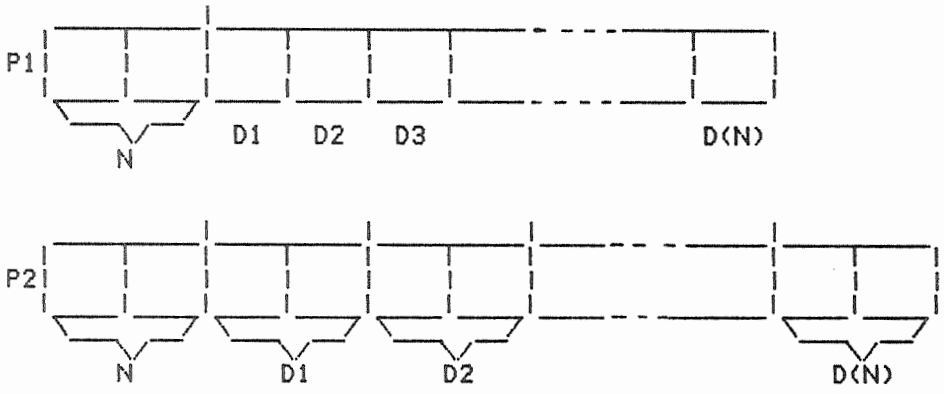
dove:

P1(I) vettore dei dati su 8 bits: nei primi due bytes indirizzati da P1(I) deve essere contenuto il numero di dati N da espandere (max 32767)

P2(K) vettore dei dati su 16 bits: nei primi due bytes sara` contenuto il numero dei dati espansi N.

P3 Flag di ritorno con informazioni sull'esito della chiamata:
= 0 operazione conclusa correttamente
= 3 errore: parametro N <= 0

Es.



INV - Utilizzata per invertire l'ordine dei dati su 8 bits all'interno di un vettore.

Tipica chiamata:

```
CALL INV(P1(I),P2)
```

dove:

P1(I) vettore dei dati da invertire: i primi due bytes indirizzati da P1(I) devono contenere il numero di dati N da invertire (max 32767)

P2 Flag di ritorno con informazioni sull'esito della chiamata:
= 0 operazione conclusa correttamente
= 3 errore: parametro N <= 0

EMU - Permette al PC di emulare una telescrivente. La configurazione del canale A usato, puo` essere effettuata utilizzando il programma CONFE che permette qualsiasi configurazione. Da tenere presente che la configurazione del collegamento col S.O. MPX permette una velocita` max di 1200 baud.

Tipica chiamata:

```
CALL EMU
```

Per rendere il controllo al programma chiamante e` sufficiente inviare CTRL-G dalla keyboard.

```
.Z80
        TITLE DARTE.MAC
        SUBTTL Driver DART A/B (RS232) rev. 2.4 (Cerri Sirio IEI-CNR Pisa)
;
;*************************************************************
;
;     Driver per porte DART canali A/B (RS232)
;     sviluppato su MINUS KYBER (Rev. 2.4 del 12/3/87)
;
;*************************************************************
;
;
;
; Autore :
; Cerri Sirio
; Istituto di Elaborazione della Informazione del C.N.R.
; Via S.Maria 46
; 56100 Pisa
; Tel. 050/500159
;
;
;
        PAGE
```

```
;
;*************************************************************
;
; Il package e' composto dalle seguenti routines :
;
; INIZ - configura il DART-CTC canali A o B :
;        19200 baud, 8 bits , 1 stop bit , no parita'.
;
; BRWV - interfaccia la chiamata Basic per lettura o scrittura
;        di vettori (max 32767 bytes)
;
; BRWS - interfaccia la chiamata Basic per lettura o scrittura
;        di stringhe (max 255 bytes)
;
; BIN  - routine di lettura vettori
;
; BOUT - routine di scrittura vettori o stringhe
;
; RCAR - routine di lettura stringhe
;
; EMU  - routine di emulazione TTY (max 1200 baud)
;
; PAK  - impacca valori (0/255) da 16 bits a 8 bits
;
; SPAK - spacca valori (0/255) da 8 bits a 16 bits
;
; INV  - scambia l'ordine di dati su 8 bit di un vettore
;
;-------------------------------------------------------------
;
; Le routine prevedono l'interruzione della richiesta
; per Time Out (256*256 tests su dato non pronto) o da
; consolle con CTRL-G.
;
;*************************************************************
;
        PAGE
```

```
                                ;
                                ;************************************************
                                ;
                                ; Routine di inizializzazione DART-CTC
                                ;
                                ; DART canale A = 42H , CTC canale 0 = 50H
                                ; DART canale B = 43H , CTC canale 2 = 52H
                                ;
                                ; 19200 baud, 8 bits , 1 stop/bit , no parita'
                                ;
                                ;------------------------------------------------
                                ;
                                ; Sequenza di chiamata e parametri :
                                ;
                                ;  CALL INIZ(P1%,P2%)
                                ;
                                ; dove : INIZ - indirizzo della routine assembler
                                ;
                                ;        P1%  - canale Dart da inizializzare
                                ;               42H = canale A CTC ch. 0 (50H)
                                ;               43H = canale B CTC ch. 2 (52H)
                                ;
                                ;        P2%  - word di ritorno errore
                                ;               =0 OK
                                ;               =2 canale richiesto errato
                                ;
                                ;************************************************
                                ;
0000' 22 033A'                  INIZ:   LD (IHL),HL     ; salva indirizzo di P1%
0003' ED 53 033C'               .       LD (IDE),DE     ; salva indirizzo di P2%
0007' 7E                        .       LD A,(HL)       ; A=canale da configurare
0008' 21 0318'                  .       LD HL,DARTAC
000B' BE                        .       CP (HL)         ; 42H = ch. A
000C' 28 0F                     .       JR Z,INIZA
000E' 21 0319'                  .       LD HL,DARTBC
0011' BE                        .       CP (HL)         ; 43H = ch. B
0012' 28 22                     .       JR Z,INIZB
0014' 2A 033C'                  .       LD HL,(IDE)     ; errore indirizzo canale
0017' 36 02                     .       LD (HL),2       ; Lbyte di P2%=2
0019' 23                        .       INC HL
001A' 36 00                     .       LD (HL),0       ; Hbyte di P2%=0
001C' C9                        .       RET
                                ;
001D' 3A 031A'                  INIZA:  LD A,(CTC0)     ; configura canale CTC0 Dart A
```

Driver DART A/B (RS232) rev. 2.4 (Cerri Sirio IEI-CNR Pisa)

```
0020'  32 002A'                    LD (INIZ1+1),A
0023'  32 002F'                    LD (INIZ2+1),A
0026'  3A 031C'                    LD A,(CTCOW1)    ; reset, carica costante di tempo (ch. A)
0029'  D3 00               INIZ1:  OUT (00),A
002B'  3A 031D'                    LD A,(CTCOW2)    ; costante di tempo RS232
002E'  D3 00               INIZ2:  OUT (00),A       ; inizializza CTC
0030'  4E                          LD C,(HL)        ; C = canale Dart/A comandi
0031'  21 0320'                    LD HL,WRA
0034'  18 17                       JR FUORI
                           ;
0036'  3A 031B'            INIZB:  LD A,(CTC2)      ; configura canale CTC2 Dart B
0039'  32 0043'                    LD (INIZ3+1),A
003C'  32 0048'                    LD (INIZ4+1),A
003F'  3A 031E'                    LD A,(CTC2W1)    ; carica costante di tempo (ch. B)
0042'  D3 00               INIZ3:  OUT (00),A
0044'  3A 031F'                    LD A,(CTC2W2)    ; costante di tempo ch. B
0047'  D3 00               INIZ4:  OUT (00),A
0049'  4E                          LD C,(HL)        ; C = canale DART/B comandi
004A'  21 0329'                    LD HL,WRB
                           ;
004D'  06 09               FUORI:  LD B,9
004F'  ED B3                       OTIR
0051'  AF                          XOR A
0052'  2A 033C'                    LD HL,(IDE)
0055'  77                          LD (HL),A        ; Lbyte di P2%=0
0056'  23                          INC HL
0057'  77                          LD (HL),A        ; Hbyte di P2%=0
0058'  C9                          RET
                                   PAGE
```

```
;
;************************************************
;
; Routine di interfaccia chiamate BASIC per
; scrittura/lettura di vettori su DART ch. A/B
;
;------------------------------------------------
;
; Sequenza di chiamata e parametri :
;
;  CALL BRWV(P1%,P2%(I),P3%(K))
;
; dove : BRWV - indirizzo della routine assembler
;
;        P1%  - Hbyte = tipo di richiesta
;
;               bit 0 = 0 read
;                     = 1 write
;
;               bit 1 = 0 abilita l'interruzione da consolle
;                     = 1 abilita l'interruzione per T.O.
;
;               bit 2 = 0 NOP
;                     = 1 inizializza il canale di I/O
;
;               Lbyte = canale I/O (40H=ch. A , 41H=ch. B)
;
;        P2%(I)  - buffer da leggere o scrivere : la prima
;                  word deve contenere il num. di bytes da
;                  trasferire
;
;        P3%(K)  - 2 words di ritorno con informazioni sulla
;                  operazione eseguita :
;
;                  word 1 = 0 trasferimento OK
;                         = 1 interruzione per T.O.
;                         = 2 indirizzo canale errato
;                         = 3 num. bytes da trasferire (=0
;                         = 7 richiesta d'interruzione da consolle
;
;                  word 2 = numero di bytes trasferiti
;
; N.B. - i parametri devono essere definiti nel programma
;        BASIC prima di effettuare la chiamata.
```

```
                                ;
                                ;*************************************************
                                ;
0059'  22 0334'                 BRWV:   LD (RHL),HL     ;salva P1%
005C'  ED 53 0336'                      LD (RDE),DE     ;salva P2%
0060'  ED 43 0338'                      LD (RBC),BC     ;salva P3%
0064'  7E                               LD A,(HL)       ;A=canale
0065'  21 0316'                         LD HL,DARTAD
0068'  BE                               CP (HL)
0069'  28 0A                            JR Z,BRWV7      ;canale A , ok
006B'  21 0317'                         LD HL,DARTBD
006E'  BE                               CP (HL)
006F'  28 04                            JR Z,BRWV7      ;canale B , ok
0071'  3E 02                            LD A,2          ;A=2 codice errore canale errato
0073'  18 59                            JR BRWV2
0075'  2A 0334'                 BRWV7:  LD HL,(RHL)     ;indirizza Lbyte di P1%
0078'  23                               INC HL          ;indirizza Hbyte di P1% : tipo di richiesta
0079'  CB 56                            BIT 2,(HL)      ;test su richiesta inizializzazione
007B'  28 0E                            JR Z,BRWV4      ;se Z no
007D'  3C                               INC A           ;A+2 canale comandi
007E'  3C                               INC A
007F'  32 0333'                         LD (BYTE),A
0082'  21 0333'                         LD HL,BYTE      ;prepara i parametri per INIZ
0085'  11 033E'                         LD DE,SERV
0088'  CD 0000'                         CALL INIZ
008B'  2A 0336'                 BRWV4:  LD HL,(RDE)     ;indirizza Lbyte del num. caratteri
008E'  46                               LD B,(HL)       ;B=Lbyte del num. caratteri
008F'  23                               INC HL          ;indirizza Hbyte del num. caratteri
0090'  7E                               LD A,(HL)       ;A=Hbyte del num. caratteri
0091'  32 0333'                         LD (BYTE),A     ;salva in BYTE
0094'  CB 7F                            BIT 7,A         ;test se A<0
0096'  20 03                            JR NZ,BRWV6     ;se NZ errore
0098'  B0                               OR B            ;test se A e B =0
0099'  20 04                            JR NZ,BRWV5     ;se NZ ok
009B'  3E 03                    BRWV6:  LD A,3          ;codice errore 3 : num. bytes <=0
009D'  18 2F                            JR BRWV2
009F'  21 0333'                 BRWV5:  LD HL,BYTE
00A2'  AF                               XOR A
00A3'  B8                               CP B            ;test su Lbyte : se=0 decrementa Hbyte
00A4'  20 01                            JR NZ,BRWV0     ;Lbyte # 0 ,Hbyte buono
00A6'  35                               DEC (HL)        ;Lbyte = 0 decremento Hbyte
00A7'  2A 0334'                 BRWV0:  LD HL,(RHL)     ;indirizza Lbyte di P1%
00AA'  4E                               LD C,(HL)       ;C=Lbyte di P1% (canale di I/O)
00AB'  23                               INC HL          ;indirizza Hbyte di P1%
00AC'  7E                               LD A,(HL)       ;A=Hbyte di P1% : tipo di richiesta
00AD'  32 0332'                         LD (FTO),A      ;salva in FTO
```

Driver DART A/B (RS232) rev. 2.4 (Cerri Sirio IEI-CNR Pisa)

```
00B0'  2A 0336'                    LD HL,(RDE)      ;HL=indirizzo del buffer
00B3'  23                          INC HL
00B4'  23                          INC HL
00B5'  CB 47                       BIT 0,A          ;test su bit 0
00B7'  28 0C                       JR Z,BRWV1       ;=0 read
00B9'  79                          LD A,C           ;#0 write , A=canale dati
00BA'  3C                          INC A
00BB'  3C                          INC A            ;A=canale comandi
00BC'  32 01B1'                    LD (BOUT1+1),A   ;configura canale comandi
00BF'  CD 01AD'                    CALL BOUT
00C2'  C3 00CE'                    JP BRWV2         ;
00C5'  79                  BRWV1:  LD A,C           ;=0 read , A=canale dati
00C6'  3C                          INC A
00C7'  3C                          INC A            ;A=canale comandi
00C8'  32 016D'                    LD (BIN1+1),A    ;configura canale comandi
00CB'  CD 0169'                    CALL BIN
00CE'  2A 0338'            BRWV2:  LD HL,(RBC)      ;indirizza Lbyte di P3%
00D1'  77                          LD (HL),A        ;salva A in Lbyte di P3%
00D2'  23                          INC HL           ;indirizza Hbyte di P3%
00D3'  AF                          XOR A
00D4'  77                          LD (HL),A        ;Hbyte di P3% = 0
00D5'  2A 0336'                    LD HL,(RDE)
00D8'  7E                          LD A,(HL)        ;Lbyte dei bytes da trasferire
00D9'  23                          INC HL           ;indirizza Hbytes
00DA'  90                          SUB B            ;sottrae i bytes rimasti da trasferire
00DB'  57                          LD D,A           ;salva in D
00DC'  30 01                       JR NC,BRWV3      ;A>=B se NC=0
00DE'  35                          DEC (HL)         ;A<B ,decremento Hbyte
00DF'  7E                  BRWV3:  LD A,(HL)        ;Hbyte dei bytes da trasferire
00E0'  21 0333'                    LD HL,BYTE
00E3'  46                          LD B,(HL)        ;Hbyte dei bytes rimasti da trasferire
00E4'  90                          SUB B
00E5'  2A 0338'                    LD HL,(RBC)      ;indirizza P3%
00E8'  23                          INC HL
00E9'  23                          INC HL
00EA'  72                          LD (HL),D        ;Lbyte dei bytes trasferiti
00EB'  23                          INC HL
00EC'  77                          LD (HL),A        ;Hbyte dei bytes trasferiti
00ED'  C9                          RET
                                   PAGE
```

```
;
;*******************************************************
;
; Routine di interfaccia chiamate BASIC per scrittura
; o lettura di una stringa (max 255 bytes) su
; DART ch. A/B .
;
; In lettura legge fino a CR o fino al 255' carattere
; se abilitata l'interruzione da consolle,
; e fino al 255' carattere se abilitata l'interruzione
; per T.O.
;
;-------------------------------------------------------
;
; Sequenza di chiamata e parametri :
;
; CALL BRWS(P1%,P2$,P3%)
;
; dove : BRWS - indirizzo routine assembler
;
;        P1%  - Hbyte = tipo di richiesta
;
;               bit 0 = 0 read
;                     = 1 write
;
;               bit 1 = 0 abilita interruzione da consolle
;                     = 1 abilita interruzione per T.O.
;
;               bit 2 = 0 NOP
;                     = 1 inizializzazione canale di I/O
;
;               Lbyte = canale di I/O (40H=ch. A, 41H=ch. B)
;
;        P2$  - stringa dei caratteri da trasferire
;
;        P3%  - parametro di ritorno :
;
;               =0 trasferimento OK
;               =1 interruzione per T.O.
;               =2 canale richiesto errato
;               =3 num. bytes richiesto =0
;               =7 richiesta di interruzione da consolle
;
; N.B. - i parametri devono essere definiti nel programma
```

```
                                ;       BASIC prima di effettuare la chiamata
                                ;     - in lettura, prima di effettuare la CALL, P2$ deve
                                ;       essere definito per almeno il num. di caratteri
                                ;       da leggere : si consiglia di usare l'istruzione
                                ;       P2$=SPACE$(255) che ne prevede il massimo.
                                ;
                                ;*******************************************************
                                ;
00EE'    22 0334'               BRWS:   LD (RHL),HL     ;salva P1%
00F1'    ED 53 0336'                    LD (RDE),DE     ;salva P2$
00F5'    ED 43 0338'                    LD (RBC),BC     ;salva P3%
00F9'    7E                             LD A,(HL)       ;A=canale richiesto
00FA'    21 0316'                       LD HL,DARTAD
00FD'    BE                             CP (HL)
00FE'    28 0A                          JR Z,BRWS5      ;se Z canale A, ok
0100'    21 0317'                       LD HL,DARTBD
0103'    BE                             CP (HL)
0104'    28 04                          JR Z,BRWS5      ;se Z canale B, ok
0106'    3E 02                          LD A,2          ;errore canale
0108'    18 57                          JR BRWS2
010A'    2A 0334'               BRWS5:  LD HL,(RHL)     ;indirizza Lbyte di P1%
010D'    23                             INC HL          ;indirizza Hbyte di P1% : tipo di richiesta
010E'    CB 56                          BIT 2,(HL)      ;test su richiesta inizializzazione
0110'    28 0E                          JR Z,BRWS4      ;se Z no
0112'    3C                             INC A           ;A+2 canale comandi
0113'    3C                             INC A
0114'    32 0333'                       LD (BYTE),A
0117'    21 0333'                       LD HL,BYTE      ;prepara i parametri per INIZ
011A'    11 033E'                       LD DE,SERV
011D'    CD 0000'                       CALL INIZ
0120'    2A 0334'               BRWS4:  LD HL,(RHL)
0123'    4E                             LD C,(HL)       ;C=Lbyte P1% canale di I/O
0124'    23                             INC HL          ;indirizza Hbyte di P1%
0125'    7E                             LD A,(HL)       ;A=tipo di richiesta
0126'    32 0332'                       LD (FTO),A      ;salva in FTO
0129'    2A 0336'                       LD HL,(RDE)     ;HL=indirizzo di P2$
012C'    46                             LD B,(HL)       ;B=num. bytes da trasferire
012D'    AF                             XOR A
012E'    B8                             CP B            ;test se B=0
012F'    20 04                          JR NZ,BRWS3     ;se NZ ok
0131'    3E 03                          LD A,3          ;codice errore num. bytes=0
0133'    18 2C                          JR BRWS2
0135'    23                     BRWS3:  INC HL          ;indirizza Lbyte della stringa
0136'    56                             LD D,(HL)       ;D=Lbyte
0137'    23                             INC HL          ;indirizza Hbyte della stringa
0138'    66                             LD H,(HL)       ;H=Hbyte
```

```
0139'  6A                        LD L,D          ;L=Lbyte
013A'  3A 0332'                  LD A,(FTO)
013D'  CB 47                     BIT 0,A         ;test su bit 0 (tipo di richiesta)
013F'  28 10                     JR Z,BRWS1      ;=0 read
0141'  79                        LD A,C          ;=1 write , A=canale dati
0142'  3C                        INC A
0143'  3C                        INC A
0144'  32 01B1'                  LD (BOUT1+1),A  ;configura canale comandi
0147'  AF                        XOR A
0148'  32 0333'                  LD (BYTE),A     ;Hbyte di num. caratteri=0
014B'  CD 01AD'                  CALL BOUT       ;
014E'  C3 0161'                  JP BRWS2
0151'  79                BRWS1:  LD A,C
0152'  32 01FD'                  LD (RCAR3+1),A  ;configura canale dati
0155'  3C                        INC A
0156'  3C                        INC A
0157'  32 01F7'                  LD (RCAR2+1),A  ;configura canale comandi
015A'  CD 01F1'                  CALL RCAR
015D'  2A 0336'                  LD HL,(RDE)
0160'  70                        LD (HL),B       ;salva il num. dei caratteri in Lbyte di P2$
0161'  2A 0338'          BRWS2:  LD HL,(RBC)     ;indirizza Lbyte di P3%
0164'  77                        LD (HL),A       ;salva A in Lbyte di P3%
0165'  23                        INC HL          ;indirizza Hbyte di P3%
0166'  AF                        XOR A
0167'  77                        LD (HL),A       ;Hbyte di P3%=0
0168'  C9                        RET
                                 PAGE
```

```
                        ;
                        ;*****************************************************
                        ; Routine di lettura (max 32767 bytes) da DART ch. A/B
                        ; Prevede l'interruzione da consolle (CTRL-G)
                        ; o per T.O.
                        ;
                        ; N.B. - le istruzioni di I/O vengono configurate
                        ;        dalla routine BRWV.
                        ;
                        ;-----------------------------------------------------
                        ;
                        ; Registri usati : A,B,C,DE,HL
                        ;
                        ; A  - parametro di ritorno
                        ;      =0 lettura OK
                        ;      =1 interruzione per T.O.
                        ;      =7 richiesta d'interruzione da consolle
                        ;
                        ; BYTE - Hbyte del num. bytes da leggere
                        ;
                        ; B  - Lbyte del num. bytes da leggere
                        ;
                        ; C  - canale dati : A=40H (ch. A) , B=41H (ch. B)
                        ;
                        ; DE - controllano il T.O.
                        ;
                        ; HL - indirizzo dati
                        ;
                        ; FTO - flag per il tipo d'interruzione
                        ;
                        ;*****************************************************
                        ;
0169'  AF               BIN:    XOR A           ;A=0 inizializza DE per T.O.
016A'  57                       LD D,A
016B'  5F                       LD E,A
016C'  DB 00            BIN1:   IN A,(00)       ; lettura stato del canale
016E'  CB 47                    BIT 0,A         ; test per carattere pronto
0170'  28 11                    JR Z,BIN2       ; =0 non pronto
0172'  ED A2                    INI             ; legge il carattere
0174'  20 F3                    JR NZ,BIN       ; test se fine caratteri
0176'  3A 0333'                 LD A,(BYTE)     ; A=Hbyte del num. caratteri
0179'  B7                       OR A            ; test su fine caratteri
017A'  28 30                    JR Z,BIN4       ; =0 si, ritorna con A=0
```

```
017C'  3D                          DEC A           ; #0 no, decrementa A
017D'  32 0333'                    LD (BYTE),A     ; aggiorna Hbyte
0180'  C3 0169'                    JP BIN          ; continua
0183'  3A 0332'            BIN2:   LD A,(FTD)      ; A = tipo d'interruzione
0186'  CB 4F                       BIT 1,A         ; test su bit 1
0188'  20 1A                       JR NZ,BIN3      ; =1 controlla T.O.
018A'  C5                          PUSH BC         ; =0 controlla consolle
018B'  E5                          PUSH HL
018C'  0E 0B                       LD C,0BH
018E'  CD 0005                     CALL 5          ; test su tasto premuto a consolle
0191'  E1                          POP HL
0192'  C1                          POP BC
0193'  B7                          OR A
0194'  28 D6                       JR Z,BIN1       ; A=0 nessun tasto premuto, continua
0196'  C5                          PUSH BC         ; #0 tasto premuto
0197'  E5                          PUSH HL
0198'  0E 01                       LD C,01H
019A'  CD 0005                     CALL 5          ; lettura carattere in A
019D'  E1                          POP HL
019E'  C1                          POP BC
019F'  FE 07                       CP 07H          ; e' CTRL-G ?
01A1'  20 C9                       JR NZ,BIN1      ; no, ignora e continua
01A3'  C9                          RET             ; si, ritorna con A=7
01A4'  14                  BIN3:   INC D           ; aggiorna D per T.O.
01A5'  20 C5                       JR NZ,BIN1      ; D#0 continua
01A7'  1C                          INC E           ; aggiorna E per T.O.
01A8'  20 C2                       JR NZ,BIN1      ; E#0 continua
01AA'  AF                          XOR A           ; E=0 T.O. avvenuto
01AB'  3C                          INC A           ; A=1 segnala errore
01AC'  C9                  BIN4:   RET
                                   PAGE
```

```
                        ;
                        ;*********************************************************
                        ;
                        ; Routine di scrittura (max 32767 bytes) su DART ch. A/B
                        ; Prevede l'interruzione da consolle (CTRL-G)
                        ; o per T.O.
                        ;
                        ; N.B. - le istruzioni di I/O vengono configurate dalla
                        ;        routine BRWV o BRWS.
                        ;
                        ;---------------------------------------------------------
                        ;
                        ; Registri usati : A,B,C,DE,HL
                        ;
                        ; A - parametro di ritorno
                        ;     =0 trasferimento OK
                        ;     =1 interruzione per T.O.
                        ;     =7 interruzione da consolle
                        ;
                        ; BYTE - Hbyte del num. bytes da trasferire
                        ;
                        ; B - Lbyte del num. bytes da trasferire
                        ;
                        ; C - canale dati : A=40H (ch. A) , B=41H (ch. B)
                        ;
                        ; DE - controllano il T.O.
                        ;
                        ; HL - indirizzo dati
                        ;
                        ; FTO - flag per il tipo d'interruzione
                        ;
                        ;*********************************************************
01AD'  AF               BOUT:   XOR A           ; inizializza per T.O.
01AE'  57                       LD D,A
01AF'  5F                       LD E,A
01B0'  DB 00            BOUT1:  IN A,(00)       ; leggo lo stato del canale
01B2'  CB 57                    BIT 2,A
01B4'  28 11                    JR Z,BOUT2      ; =0 non pronto
01B6'  ED A3                    OUTI            ; #0 pronto, manda carattere
01B8'  20 F3                    JR NZ,BOUT      ; test su B=Lbyte #0 continua
01BA'  3A 0333'                 LD A,(BYTE)     ; =0 controlla Hbyte
01BD'  B7                       OR A
01BE'  28 30                    JR Z,BOUT4      ; test su A=Hbyte, se = 0 fine, ritorna
01C0'  3D                       DEC A           ; decrementa A
```

```
01C1'  32 0333'                    LD (BYTE),A     ; aggiorna Hbyte
01C4'  C3 01AD'                    JP BOUT         ; continua
01C7'  3A 0332'          BOUT2:    LD A,(FTD)      ; A=tipo d'interruzione
01CA'  CB 4F                       BIT 1,A         ; test su bit 1
01CC'  20 1A                       JR NZ,BOUT3     ; #0 test per T.O.
01CE'  C5                          PUSH BC         ; =0 test su consolle
01CF'  E5                          PUSH HL
01D0'  0E 0B                       LD C,0BH
01D2'  CD 0005                     CALL 5          ; test su tasto premuto a consolle
01D5'  E1                          POP HL
01D6'  C1                          POP BC
01D7'  B7                          OR A
01D8'  28 D6                       JR Z,BOUT1      ; A=0 nessun tasto premuto, continua
01DA'  C5                          PUSH BC         ; A#0 tasto premuto
01DB'  E5                          PUSH HL
01DC'  0E 01                       LD C,01H
01DE'  CD 0005                     CALL 5          ; lettura carattere in A
01E1'  E1                          POP HL
01E2'  C1                          POP BC
01E3'  FE 07                       CP 07H          ; e' CTRL-G ?
01E5'  20 C9                       JR NZ,BOUT1     ; no, ignora e continua
01E7'  C9                          RET             ; si, ritorna con A=7
01E8'  14                BOUT3:    INC D           ; aggiorna D per T.O.
01E9'  20 C5                       JR NZ,BOUT1     ; D#0 continua
01EB'  1C                          INC E           ; aggiorna E per T.O.
01EC'  20 C2                       JR NZ,BOUT1     ; E#0 continua
01EE'  AF                          XOR A           ; E=0 T.O. avvenuto
01EF'  3C                          INC A           ; A=1 segnala errore
01F0'  C9                BOUT4:    RET
                                   PAGE
```

```
                        ;
                        ;*********************************************************
                        ;
                        ; Routine di lettura caratteri (max 255).
                        ;
                        ; Prevede l'interruzione da consolle (CTRL-G)
                        ; o per T.O.
                        ;
                        ; Nel primo caso l'input termina con un CR (codice 13) oppure
                        ; al 255' carattere (oltre che con CRTL-G da consolle).
                        ;
                        ; Nel secondo caso termina con il 255' carattere (oltre che
                        ; per T.O.).
                        ;
                        ; N.B. - le istruzioni di I/O vengono configurate dalla
                        ;        routine BRWS.
                        ;
                        ;---------------------------------------------------------
                        ;
                        ; Registri usati : A,B,DE,HL
                        ;
                        ; A  - registro di lavoro
                        ;
                        ; B  -  conta caratteri
                        ;
                        ; DE - controllano il T.O.
                        ;
                        ; HL - indirizzo buffer caratteri
                        ;
                        ; FTO - flag per il tipo d'interruzione
                        ;
                        ;*********************************************************
                        ;
01F1'  06 00            RCAR:   LD B,00H        ; inizializza il conta caratteri
01F3'  AF               RCAR1:  XOR A           ; inizializza per T.O.
01F4'  57                       LD D,A
01F5'  5F                       LD E,A
01F6'  DB 00            RCAR2:  IN A,(00)       ; lettura dello stato del canale
01F8'  CB 47                    BIT 0,A         ; test per carattere pronto
01FA'  28 18                    JR Z,RCAR4      ; =0 non pronto
01FC'  DB 00            RCAR3:  IN A,(00)       ; #0 pronto, lo carico in A
01FE'  77                       LD (HL),A       ; salvo il carattere
01FF'  3A 0332'                 LD A,(FTO)      ; test sul tipo d'interruzione
0202'  CB 4F                    BIT 1,A
```

```
0204'  20 05                        JR NZ,RCAR7     ; #0 non controlla il carattere letto
0206'  7E                           LD A,(HL)       ; ultimo carattere letto in A
0207'  FE 0D                        CP 0DH          ; e' CR ?
0209'  28 07                        JR Z,RCAR5      ; si, fine input : ritorno
020B'  23                  RCAR7:   INC HL          ; incremento per carattere successivo
020C'  04                           INC B           ; aggiorno il num. dei caratteri letti
020D'  AF                           XOR A
020E'  2F                           CPL             ; A=255
020F'  B8                           CP B            ; controllo se num. caratteri letti = 255
0210'  20 E1                        JR NZ,RCAR1     ; no, leggo ancora
0212'  AF                  RCAR5:   XOR A           ; si , fine input : ritorna con A=0
0213'  C9                           RET
0214'  3A 0332'            RCAR4:   LD A,(FTD)      ; A=tipo d'interruzione
0217'  CB 4F                        BIT 1,A         ; test su bit 1
0219'  20 1A                        JR NZ,RCAR6     ; #0 test per T.O.
021B'  C5                           PUSH BC         ; =0 test su consolle
021C'  E5                           PUSH HL
021D'  0E 0B                        LD C,0BH
021F'  CD 0005                      CALL 5          ; test su tasto premuto a consolle
0222'  E1                           POP HL
0223'  C1                           POP BC
0224'  B7                           OR A
0225'  28 CF                        JR Z,RCAR2      ; =0 nessun tasto premuto, continua
0227'  C5                           PUSH BC         ; #0 tasto premuto
0228'  E5                           PUSH HL
0229'  0E 01                        LD C,01H
022B'  CD 0005                      CALL 5          ;lettura carattere in A
022E'  E1                           POP HL
022F'  C1                           POP BC
0230'  FE 07                        CP 07H          ; e' CTRL-G ?
0232'  20 C2                        JR NZ,RCAR2     ; no , ignora e continua
0234'  C9                           RET             ; si , ritorna con A=7
0235'  14                  RCAR6:   INC D           ; aggiorna D per T.O.
0236'  20 BE                        JR NZ,RCAR2     ; D#0 continua
0238'  1C                           INC E           ; aggiorna E per T.O.
0239'  20 BB                        JR NZ,RCAR2     ; E#0 continua
023B'  AF                           XOR A           ; E=0 T.O. avvenuto
023C'  3C                           INC A           ; A=1 segnala errore
023D'  C9                           RET
                                    PAGE
```

```
                        ;
                        ;*************************************************************
                        ;
                        ;       Routine di emulazione TTY (max 1200 baud)
                        ;
                        ;       8 bits/carattere, 1 stop/bit, no parita'
                        ;
                        ;-------------------------------------------------------------
                        ;
                        ;       Usa il canale Dart-A (RS232)
                        ;
                        ;       Sequenza di chiamata :
                        ;
                        ;         CALL EMU%
                        ;
                        ;         dove : EMU% - indirizzo della routine assembler
                        ;
                        ;       Per ritornare al programma chiamante usare CTRL-G
                        ;
                        ;  N.B. prima di effettuare la chiamata cambiare la velocita'
                        ;       di  trasmissione scrivendo nella locazione CTCOW2 il
                        ;       valore desiderato come da tabella (vedi manuale per
                        ;       altre configurazioni) :
                        ;
                        ;               hex  dec   baud
                        ;
                        ;                C0  192    600
                        ;                60   96   1200
                        ;
                        ;*************************************************************
                        ;
023E'   21 0318'        EMU:    LD HL,DARTAC    ; configura canale comandi
0241'   7E                      LD A,(HL)
0242'   32 0253'                LD (EMU0+1),A
0245'   32 0274'                LD (WAIT+1),A
0248'   21 0316'                LD HL,DARTAD    ; configura canale dati
024B'   7E                      LD A,(HL)
024C'   32 0259'                LD (EMU1+1),A
024F'   32 027B'                LD (EMU2+1),A
0252'   DB 00           EMU0:  IN A,(00)       ; lettura stato canale comandi Dart-A (RS232)
0254'   CB 47                  BIT 0,A         ; carattere pronto ? 0=no
0256'   28 0A                  JR Z,KYB        ; =0 controlla la keyboard
0258'   DB 00           EMU1:  IN A,(00)       ; =1 lettura carattere da RS232
025A'   5F                      LD E,A
```

```
025B'  0E 02              VID:   LD C,02H
025D'  CD 0005                   CALL 5          ; stampa carattere su video
0260'  18 F0                     JR EMUO
0262'  0E 0B              KYB:   LD C,0BH
0264'  CD 0005                   CALL 5          ; lettura stato keyboard
0267'  B7                        OR A
0268'  28 E8                     JR Z,EMUO       ; A=0 nessun tasto premuto
026A'  0E 01                     LD C,01H        ; A#0 carattere pronto
026C'  CD 0005                   CALL 5          ; lettura carattere
026F'  5F                        LD E,A          ; salva il carattere in E
0270'  FE 07                     CP 07H          ; controlla se CTRL-G
0272'  C8                        RET Z           ; si, ritorna
0273'  DB 00              WAIT:  IN A,(00)       ; lettura stato RS232
0275'  CB 57                     BIT 2,A         ; test se disponibile
0277'  28 FA                     JR Z,WAIT       ; =0 no, aspetta
0279'  7B                        LD A,E
027A'  D3 00              EMU2:  OUT (00),A      ; manda il carattere su RS232
027C'  18 D4                     JR EMUO
                                 PAGE
```

```
;
;*************************************************************
;
; Routine di impaccamento dati.
;
; Dato un vettore con valori interi tra 0 e 255 su 16 bits,
; la routine restituisce un vettore con i dati su 8 bits.
;
;-------------------------------------------------------------
;
; Sequenza di chiamata e parametri :
;
;  CALL PAK(P1%(0),P2%(0),P3%)
;
; dove : PAK   - indirizzo della routine assembler
;
;        P1%(I) - vettore dei dati su 16 bits : la
;                 prima word deve contenere il num.
;                 dei dati.
;
;        P2%(I) - vettore di ritorno con i dati su
;                 8 bits : la prima word conterra'
;                 il num. dei bytes validi
;
;        P3%    - 1 word di ritorno con informazioni
;                 sull'esito della chiamata :
;                 word = 0 Ok
;                      = 3 errore : P1%(0) (= 0
;
; Esempio :
;
;       Dato il vettore A% con dati su 16 bits con
;       valori tra 0 e 255 ,ad esempio
;          A%(1)=25 , A%(2)=38 , A%(3)=56
;       effettuando la chiamata alla routine con
;          CALL PAK(A%(0),B%(0),ST%)
;       dove A%(0) = 3  (num. dei dati su 16 bits)
;       verra' restituito il vettore B% come segue :
;          B%(0) = 3 (num. dei dati su 8 bits)
;          B%(1) = Lbyte 25 , Hbyte 38
;          B%(2) = Lbyte 56 , Hbyte vecchio valore
;
;*************************************************************
;
```

```
027E'  22 0334'          PAK:   LD (RHL),HL     ;salva P1%
0281'  ED 53 0336'              LD (RDE),DE     ;salva P2%
0285'  ED 43 0338'              LD (RBC),BC     ;salva P3%
0289'  7E                       LD A,(HL)       ;A=Lbyte di A%(0)
028A'  12                       LD (DE),A       ;copia A in Lbyte di B%(0)
028B'  4F                       LD C,A          ;lo copia anche in C
028C'  23                       INC HL          ;indirizza Hbyte di A%(0)
028D'  13                       INC DE          ;indirizza Hbyte di B%(0)
028E'  7E                       LD A,(HL)       ;A=Hbyte di A%(0)
028F'  12                       LD (DE),A       ;copia A in Hbyte di B%(0)
0290'  47                       LD B,A          ;lo copia anche in B
0291'  CB 7F                    BIT 7,A         ;controllo se A%(0) < 0
0293'  20 03                    JR NZ,PAK0      ;se NZ errore
0295'  B1                       OR C            ;controllo se A%(0) = 0
0296'  20 04                    JR NZ,PAK1      ;se NZ Ok
0298'  3E 03             PAK0:  LD A,3          ;errore, A=3 : A%(0) (= 0
029A'  18 0B                    JR PAK2
029C'  23                 PAK1: INC HL          ;indirizza Lbyte di A%(I)
029D'  13                       INC DE          ;indirizza Lbyte di B%(I)
029E'  7E                       LD A,(HL)       ;A=dato
029F'  12                       LD (DE),A       ;lo copia in B%(I)
02A0'  23                       INC HL          ;incremento per saltare Hbyte di A%(I)
02A1'  0B                       DEC BC          ;decremento il num. dei dati
02A2'  78                       LD A,B          ;e controllo se = 0
02A3'  B1                       OR C
02A4'  20 F6                    JR NZ,PAK1      ;se NZ continua
02A6'  AF                       XOR A           ;A=0 per operazione Ok
02A7'  2A 0338'           PAK2: LD HL,(RBC)     ;indirizza Lbyte di P3%
02AA'  77                       LD (HL),A       ;e salva A
02AB'  23                       INC HL          ;indirizza Hbyte di P3%
02AC'  AF                       XOR A           ;A = 0
02AD'  77                       LD (HL),A       ;Hbyte di P3%=0
02AE'  C9                       RET
                                PAGE
```

```
;
;*************************************************************
;
; Routine di spaccamento dati.
;
; Dato un vettore con due valori interi tra 0 e 255 su 16 bits
; (uno per byte), la routine restituisce un vettore con un
; valore su 16 bits (uno ogni due bytes).
;
;-------------------------------------------------------------
;
; Sequenza di chiamata e parametri :
;
;  CALL SPAK(P1%(0),P2%(0),P3%)
;
; dove : SPAK   - indirizzo della routine assembler
;
;        P1%(I) - vettore dei dati su 8 bits : la
;                 prima word deve contenere il num.
;                 dei dati.
;
;        P2%(I) - vettore di ritorno con i dati su
;                 16 bits : la prima word conterra'
;                 il num. delle words valide
;
;        P3%    - 1 word di ritorno con informazioni
;                 sull'esito della chiamata :
;                 word = 0 Ok
;                      = 3 errore : P1%(0) (= 0
;
; Esempio :
;
;       Dato il vettore A% con dati su 8 bits, ad esempio
;          A%(1) : Lbyte = 25 , Hbyte = 38
;          A%(2) : Lbyte = 33 , Hbyte = 37
;          A%(3) : Lbyte = 42 , Hbyte = 39
;       effettuando la chiamata alla routine con
;          CALL SPAK(A%(0),B%(0),ST%)
;       dove A%(0) = 6  (num. dei dati su 8 bits)
;       verra' restituito il vettore B% come segue :
;          B%(0) = 6 (num. dei dati su 16 bits)
;          B%(1) : Lbyte = 25 , Hbyte = 0
;          B%(2) : Lbyte = 38 , Hbyte = 0
;          B%(3) : Lbyte = 33 , Hbyte = 0
```

```
                                ;          B%(4) : Lbyte = 37 , Hbyte = 0
                                ;          B%(5) : Lbyte = 42 , Hbyte = 0
                                ;          B%(6) : Lbyte = 39 , Hbyte = 0
                                ;
                                ;*************************************************************
                                ;

02AF'   22 0334'        SPAK:   LD (RHL),HL     ;salva P1%
02B2'   ED 53 0336'             LD (RDE),DE     ;salva P2%
02B6'   ED 43 0338'             LD (RBC),BC     ;salva P3%
02BA'   7E                      LD A,(HL)       ;A=Lbyte di A%(0)
02BB'   12                      LD (DE),A       ;copia A in Lbyte di B%(0)
02BC'   4F                      LD C,A          ;lo copia anche in C
02BD'   23                      INC HL          ;indirizza Hbyte di A%(0)
02BE'   13                      INC DE          ;indirizza Hbyte di B%(0)
02BF'   7E                      LD A,(HL)       ;A=Hbyte di A%(0)
02C0'   12                      LD (DE),A       ;copia A in Hbyte di B%(0)
02C1'   47                      LD B,A          ;lo copia anche in B
02C2'   CB 7F                   BIT 7,A         ;controllo se A%(0) ( 0
02C4'   20 03                   JR NZ,SPAK0     ;se NZ errore
02C6'   B1                      OR C            ;controllo se A%(0) = 0
02C7'   20 04                   JR NZ,SPAK1     ;se NZ Ok
02C9'   3E 03           SPAK0:  LD A,3          ;errore, A=3 : A%(0) (= 0
02CB'   18 0D                   JR SPAK2
02CD'   23              SPAK1:  INC HL          ;indirizza Lbyte di A%(I)
02CE'   13                      INC DE          ;indirizza Lbyte di B%(I)
02CF'   7E                      LD A,(HL)       ;A=dato
02D0'   12                      LD (DE),A       ;lo copia in B%(I)
02D1'   13                      INC DE          ;indirizza Hbyte di B%(I)
02D2'   AF                      XOR A           ;A=0
02D3'   12                      LD (DE),A       ;copia A in Hbyte di B%(I)
02D4'   0B                      DEC BC          ;decremento il num. dei dati
02D5'   78                      LD A,B          ;e controllo se = 0
02D6'   B1                      OR C
02D7'   20 F4                   JR NZ,SPAK1     ;se NZ continua
02D9'   AF                      XOR A           ;A=0 per operazione Ok
02DA'   2A 0338'        SPAK2:  LD HL,(RBC)     ;indirizza Lbyte di P3%
02DD'   77                      LD (HL),A       ;e salva A
02DE'   23                      INC HL          ;indirizza Hbyte di P3%
02DF'   AF                      XOR A           ;A = 0
02E0'   77                      LD (HL),A       ;Hbyte di P3%=0
02E1'   C9                      RET
                                PAGE
```

```
;
;*************************************************************
;
; Routine di inversione posizione dati.
;
; Dato un vettore con due valori interi su 16 bits (uno per
; byte), la routine restituisce lo stesso vettore con i valori
; (byte) scambiati di posizione (primo con ultimo, secondo con
; penultimo , ecc...).
;
;-------------------------------------------------------------
;
; Sequenza di chiamata e parametri :
;
;  CALL INV(P1%(0),P2%)
;
; dove : INV     - indirizzo della routine assembler
;
;        P1%(I) - vettore dei dati su 8 bits : la
;                 prima word deve contenere il num.
;                 dei dati.
;
;        P2%     - 1 word di ritorno con informazioni
;                 sull'esito della chiamata :
;                 word = 0 Ok
;                      = 3 errore : P1%(0) (= 0
;
; Esempio :
;
;       Dato il vettore A% con dati su 8 bits, ad esempio
;          A%(1) : Lbyte = 25 , Hbyte = 38
;          A%(2) : Lbyte = 33 , Hbyte = 37
;          A%(3) : Lbyte = 42 , Hbyte = xx
;       effettuando la chiamata alla routine con
;          CALL INV(A%(0),ST%)
;       dove A%(0) = 5  (num. dei bytes da scambiare)
;       verra' restituito il vettore A% come segue :
;          A%(1) : Lbyte = 42 , Hbyte = 37
;          A%(2) : Lbyte = 33 , Hbyte = 38
;          B%(3) : Lbyte = 25 , Hbyte = xx
;
;*************************************************************
;
```

Driver DART A/B (RS232) rev. 2.4 (Cerri Sirio IEI-CNR Pisa)

```
02E2' 22 0334'          INV:  LD (RHL),HL     ;salva P1%
02E5' ED 53 0336'             LD (RDE),DE     ;salva P2%
02E9' 4E                      LD C,(HL)       ;C=Lbyte di A%(0)
02EA' 23                      INC HL          ;indirizza Hbyte di A%(0)
02EB' 46                      LD B,(HL)       ;B=Hbyte di A%(0)
02EC' 54                      LD D,H
02ED' 5D                      LD E,L
02EE' 09                      ADD HL,BC       ;HL=indirizzo dell'ultimo byte
02EF' 13                      INC DE          ;DE=indirizzo del primo byte
02F0' CB 28                   SRA B           ;BC=int(BC/2) : num. di scambi
02F2' CB 19                   RR C            ;da effettuare
02F4' CB 78                   BIT 7,B         ;controllo se BC < 0
02F6' 20 04                   JR NZ,INV0      ;se NZ errore
02F8' 78                      LD A,B
02F9' B1                      OR C            ;controllo se BC = 0
02FA' 20 04                   JR NZ,INV1      ;se NZ Ok
02FC' 3E 03             INV0: LD A,3          ;errore, A=3 : A%(0) <= 1
02FE' 18 0E                   JR INV2
0300' 7E                INV1: LD A,(HL)       ;A=ultimo dato
0301' 08                      EX AF,AF'       ;salvo A in A'
0302' 1A                      LD A,(DE)       ;A=primo dato
0303' 77                      LD (HL),A       ;salva A al posto dell'ultimo
0304' 08                      EX AF,AF'       ;recupero in A l'ultimo dato
0305' 12                      LD (DE),A       ;salva A al posto del primo
0306' 2B                      DEC HL          ;indirizza l'ultimo scambiato meno 1
0307' 13                      INC DE          ;indirizza l'ultimo scambiato piu' 1
0308' 0B                      DEC BC          ;decremento il num. degli scambi
0309' 78                      LD A,B          ;e controllo se = 0
030A' B1                      OR C
030B' 20 F3                   JR NZ,INV1      ;se NZ continua
030D' AF                      XOR A           ;A=0 per operazione Ok
030E' 2A 0336'          INV2: LD HL,(RDE)     ;indirizza Lbyte di P2%
0311' 77                      LD (HL),A       ;e salva A
0312' 23                      INC HL          ;indirizza Hbyte di P2%
0313' AF                      XOR A           ;A = 0
0314' 77                      LD (HL),A       ;Hbyte di P3%=0
0315' C9                      RET
                              PAGE
```

```
                        ;
                        ;*****************************************************************
                        ;
                        ;       Costanti e variabili
                        ;
                        ;*****************************************************************
                        ;
0316'   40              DARTAD: DEFB 40H        ;Dart A canale dati
0317'   41              DARTBD: DEFB 41H        ;Dart B canale dati
0318'   42              DARTAC: DEFB 42H        ;Dart A canale comandi
0319'   43              DARTBC: DEFB 43H        ;Dart B canale comandi
031A'   50              CTC0:   DEFB 50H        ;CTC canale 0 Dart/A
031B'   52              CTC2:   DEFB 52H        ;CTC canale 2 Dart/B
                        ;
031C'   57              CTC0W1: DEFB 57H        ;Reset, carica costante di tempo ch. A
031D'   06              CTC0W2: DEFB 06H        ;Costante di tempo per 19200 baud ch. A
031E'   45              CTC2W1: DEFB 45H        ;Carica costante di tempo ch. B
031F'   06              CTC2W2: DEFB 06H        ;Costante di tempo per 19200 baud ch. B
                        ;
0320'   18              WRA:    DEFB 18H        ;WR0 : reset del canale
0321'   34                      DEFB 34H        ;WR0 : reset errori, indirizza WR4
0322'   44                      DEFB 44H        ;WR4 : clock x16, 1 stop bit, no parita'
0323'   03                      DEFB 03H        ;WR0 : indirizza WR3
0324'   E1                      DEFB 0E1H       ;WR3 : Rx abilitata, 8 bit, autoenable
0325'   15                      DEFB 15H        ;WR0 : reset ext stat int, indirizza WR5
0326'   EA                      DEFB 0EAH       ;WR5 : Tx abilitata, 8 bit, RTS, DTR
0327'   29                      DEFB 29H        ;WR0 : reset Tx int, indirizza WR1
0328'   00                      DEFB 00H        ;WR1 : int disabilitati
                        ;
0329'   18              WRB:    DEFB 18H        ;WR0 : reset del canale
032A'   34                      DEFB 34H        ;WR0 : reset errori, indirizza WR4
032B'   44                      DEFB 44H        ;WR4 : clock x16, 1 stop bit, no parita'
032C'   31                      DEFB 31H        ;WR0 : reset errori, indirizza WR1
032D'   00                      DEFB 00H        ;WR1 : int disabilitati
032E'   33                      DEFB 33H        ;WR0 : reset errori, indirizza WR3
032F'   C1                      DEFB 0C1H       ;WR3 : Rx abilitata, 8 bit
0330'   35                      DEFB 35H        ;WR0 : reset errori, indirizza WR5
0331'   EA                      DEFB 0EAH       ;WR5 : Tx abilitata, 8 bit, RTS, DTR
                        ;
0332'   00              FTO:    DEFB 00
0333'   00              BYTE:   DEFB 00
0334'   0000            RHL:    DEFW 00
0336'   0000            RDE:    DEFW 00
0338'   0000            RBC:    DEFW 00
```

```
033A'  0000                IHL:   DEFW 00
033C'  0000                IDE:   DEFW 00
033E'  0000                SERV:  DEFW 00
                                  PAGE
```

```
                        ;
                        ;*****************************************************************
                        ;
                        ;  Spostamento delle routines per Save su disco
                        ;
                        ;  Operazioni da effettuare per la creazione del file
                        ;  ridotto caricabile da Basic:
                        ;
                        ;  1 - compilare con M80 DARTE,DARTE=DARTE
                        ;  2 - linkare con L80 /P:C000,DARTE/E
                        ;  3 - caricare MBASIC52 /M:&HC000
                        ;  4 - fare una CALL all'indirizzo MUOVE (C340H)
                        ;  5 - sotto CP/M fare SAVE npag.(256 bytes) nome.COM (SAVE 4 DARTE.COM)
                        ;
                        ;*****************************************************************
                        ;
0340                    NBYTES  EQU $-INIZ
0340'  21 C000          MUOVE:  LD HL,0C000H    ;Sposta le routine da C000 (NBYTES bytes)
0343'  11 0100                  LD DE,0100H     ;a 0100 per Save su disco
0346'  01 0340                  LD BC,NBYTES
0349'  ED B0                    LDIR
034B'  C3 0000                  JP 0
                                END
```

```
DARTE.MAC        MACRO-80 3.37   08-May-80        PAGE     5
Driver DART A/B (RS232) rev. 2.4 (Cerri Sirio IEI-CNR Pisa)

Macros:

Symbols:
BIN      0169'   BIN1     016C'   BIN2     0183'   BIN3     01A4'
BIN4     01AC'   BOUT     01AD'   BOUT1    01B0'   BOUT2    01C7'
BOUT3    01E8'   BOUT4    01F0'   BRWS     00EE'   BRWS1    0151'
BRWS2    0161'   BRWS3    0135'   BRWS4    0120'   BRWS5    010A'
BRWV     0059'   BRWV0    00A7'   BRWV1    00C5'   BRWV2    00CE'
BRWV3    00DF'   BRWV4    008B'   BRWV5    009F'   BRWV6    009B'
BRWV7    0075'   BYTE     0333'   CTC0     031A'   CTC0W1   031C'
CTC0W2   031D'   CTC2     031B'   CTC2W1   031E'   CTC2W2   031F'
DARTAC   0318'   DARTAD   0316'   DARTBC   0319'   DARTBD   0317'
EMU      023E'   EMU0     0252'   EMU1     0258'   EMU2     027A'
FTO      0332'   FUORI    004D'   IDE      033C'   IHL      033A'
INIZ     0000'   INIZ1    0029'   INIZ2    002E'   INIZ3    0042'
INIZ4    0047'   INIZA    001D'   INIZB    0036'   INV      02E2'
INV0     02FC'   INV1     0300'   INV2     030E'   KYB      0262'
MUOVE    0340'   NBYTES   0340    PAK      027E'   PAK0     0298'
PAK1     029C'   PAK2     02A7'   RBC      0338'   RCAR     01F1'
RCAR1    01F3'   RCAR2    01F6'   RCAR3    01FC'   RCAR4    0214'
RCAR5    0212'   RCAR6    0235'   RCAR7    020B'   RDE      0336'
RHL      0334'   SERV     033E'   SPAK     02AF'   SPAK0    02C9'
SPAK1    02CD'   SPAK2    02DA'   VID      025B'   WAIT     0273'
WRA      0320'   WRB      0329'


No  Fatal error(s)


A>
```