

```

1 3 ; =====
2 ; --- EP382
3 ; --- NEZ80 - DISASSEMBLY
4 ; --- Commenti : Antonio Giuliana
5 ; --- 002.2019
6 ; =====
7
8 ; DISPLAY
9 ;
10 ; DSP8 DSP7 DSP6 DSP5 DSP4 DSP3 DSP2 DSP1
11 ; $0007 $0006 $0005 $0004 $0003 $0002 $0001 $0000
12 ; BDISP8 BDISP7 BDISP6 BDISP5 BDISP4 BDISP3 BDISP2 BDISP1
13 ;
14 ; SYMBOL ----- ADDRESS AREA ----- SYMBOL == DATA AREA =
15
16 BDISP1 EQU $0000 ; Immagine in memoria display 1
17 BDISP2 EQU $0001 ; Immagine in memoria display 2
18 BDISP3 EQU $0002 ; Immagine in memoria display 3
19 BDISP4 EQU $0003 ; Immagine in memoria display 4
20 BDISP5 EQU $0004 ; Immagine in memoria display 5
21 BDISP6 EQU $0005 ; Immagine in memoria display 6
22 BDISP7 EQU $0006 ; Immagine in memoria display 7
23 BDISP8 EQU $0007 ; Immagine in memoria display 8
24
25 PORTD0 EQU $00F0 ; Porta per il display 0
26 PORTD8 EQU $00F7 ; Porta per il display 8
27
28 IMGREGA EQU $0008 ; Immagine in memoria registro A
29 IMGREGF EQU $0009 ; Immagine in memoria registro F
30 IMGREGB EQU $000A ; Immagine in memoria registro B
31 IMGREGC EQU $000B ; Immagine in memoria registro C
32 IMGREGD EQU $000C ; Immagine in memoria registro D
33 IMGREGE EQU $000D ; Immagine in memoria registro E
34 IMGREGL EQU $000E ; Immagine in memoria registro L
35 IMGREGH EQU $000F ; Immagine in memoria registro H
36 IMGSPLE EQU $0010 ; Immagine in memoria parte bassa SP
37 IMGSPHE EQU $0011 ; Immagine in memoria parte alta SP
38 IMGPCLE EQU $0012 ; Immagine in memoria parte bassa PC
39 IMGPCHE EQU $0013 ; Immagine in memoria parte alta PC
40
41 DELYTL EQU $0020 ; Parte bassa valore ritardo in ms
42 DELYTH EQU $0021 ; Parte alta valore ritardo in ms
43 TAPECL EQU $0022 ; Contatore parte bassa memoria per
Tape
44 TAPECH EQU $0023 ; Contatore parte alta memoria per
Tape
45
46 EPINT EQU $0038 ; Entry point INT
47 EPNMI EQU $0066 ; Entry point NMI
48
49 CTRL0 EQU $10 ; Codice tasto Control 0
50 CTRL1 EQU $11 ; Codice tasto Control 1
51 CTRL2 EQU $12 ; Codice tasto Control 2
52 CTRL3 EQU $13 ; Codice tasto Control 3
53 CTRL4 EQU $14 ; Codice tasto Control 4
54 CTRL5 EQU $15 ; Codice tasto Control 5
55 CTRL6 EQU $16 ; Codice tasto Control 6
56 CTRL7 EQU $17 ; Codice tasto Control 7
57 CTRLF EQU $1F ; Codice tasto Control F
58
59
60 ; RESET $8000
61 ; Main reset entry point
62 ;
63 8000 RESET C3 AB 81 JMP MAIN ; Passa alla MAIN ($81AB)
64
65
66 ; SHFDAT $8003
67 ; IN : A=<nuovo valore>

```

```

68 ; REG SAVED : HL BC
69 ;
70 ; DSP2 DSP1
71 ; $0001 $0000
72 ; BDISP2 BDISP1
73 ; == DATA AREA =
74 ;
75 ; Shift a sinistra dei valori da BDISP1 a BDISP2 (Area Dati)
76 ; Inserisce nuovo dato presente in A in BDISP1
77 ; I dati non sono visualizzati immediatamente
78 ;
79 8003 SHFDAT E5 PUSH HL ; Salva nello stack HL
80 8004 C5 PUSH BC ; Salva nello stack BC
81 8005 21 01 00 LD HL,BDISP2 ; Carica in HL BDISP2 ($0001)
82 8008 0E 02 LD C,$02 ; Totale display da shiftare=$02
83 800A 18 0B JR SHFTXB ; Passa alla SHFTXB ($8017)
84
85
86 ; SHFADD $800C
87 ; IN : A=<nuovo valore>
88 ; REG SAVED : HL BC
89 ;
90 ; DSP7 DSP6 DSP5 DSP4
91 ; $0006 $0005 $0004 $0003
92 ; BDISP7 BDISP6 BDISP5 BDISP4
93 ; ----- ADDRESS AREA -----
94 ;
95 ; Shift a sinistra dei valori da BDISP4 a BDISP7 (Area Indirizzi)
96 ; Inserisce nuovo dato presente in A in BDISP4
97 ; I dati non sono visualizzati immediatamente
98 ;
99 800C SHFADD E5 PUSH HL ; Salva nello stack HL
100 800D C5 PUSH BC ; Salva nello stack BC
101 800E 21 06 00 LD HL,BDISP7 ; Carica in HL BDISP7 ($0006)
102 8011 0E 04 LD C,$04 ; Totale display da shiftare=$04
103 8013 18 02 JR SHFTXB ; Passa alla SHFTXB ($8017)
104
105
106 ; SHFTX $8015
107 ; IN : <A=nuovo valore> HL=<ultima locazione> C=<numero locazioni>
108 ; REG SAVED : HL BC
109 ; Shift a sinistra dei valori delle locazioni passate come parametri
110 ; Inserisce nuovo dato presente in A nella locazione più bassa
111 ;
112 8015 SHFTX E5 PUSH HL ; Salva nello stack HL
113 8016 C5 PUSH BC ; Salva nello stack BC
114 ; <Continua eseguendo la SHFTXB>
115
116
117 ; SHFTXB $8017
118 ; IN : <A=nuovo valore> HL=<ultima locazione> C=<numero locazioni>
119 ; REG SAVED : NONE (RELOAD HL BC FROM STACK)
120 ; Shift a sinistra dei valori delle locazioni passate come parametri
121 ; Inserisce nuovo dato presente in A nella locazione più bassa
122 ;
123 8017 SHFTXB 0D DEC C ; Decrementa contatore totale display
124 8018 LAA001 2D DEC L ; Decrementa puntatore
125 ; a buffer display
126 8019 46 LD B,(HL) ; Carica in B valore puntato
127 ; nel buffer da HL
128 801A 2C INC L ; Incrementa puntatore
129 ; a buffer display
130 801B 70 LD (HL),B ; Salva B nel buffer puntato da HL
131 801C 2D DEC L ; Decrementa puntatore
132 ; a buffer display
133 801D 0D DEC C ; Decrementa contatore totale display
134 801E 20 F8 JR NZ,LAA001 ; Se diverso da zero,
135 ; passa a LAA001 ($8018) e ripete
136 ; per tutti i display rimanenti

```

```

137 8020          77          LD (HL),A          ; Salva nuovo valore in A nel
138                                     ; buffer puntato da HL
139 8021          C1          POP BC           ; Recupera BC dallo stack
140 8022          E1          POP HL           ; Recupera HL dallo stack
141 8023          C9          RET              ; Termina
142
143
144 ; LDADAT $8024
145 ; OUT: A=<valore compattato per 2 display>
146 ; REG SAVED : HL
147 ;
148 ; DSP2   DSP1
149 ; $0001 $0000
150 ; BDISP2 BDISP1
151 ; == DATA AREA =
152 ;
153 ; Carica nel registro A il valore delle locazioni BDISP2 BDISP1 (Area Dati)
154 ; compattati nel nibble alto e basso
155 ;
156 8024 LDADAT E5          PUSH HL           ; Salva nello stack HL
157 8025      21 01 00     LD HL,BDISP2     ; HL punta a DISP2 ($0001)
158 8028      18 01      JR LDAX2B         ; Passa alla LDAX2B ($802B)
159
160
161 ; LDAX2 $802A
162 ; IN : HL=<locazione più alta>
163 ; OUT: A=<valore compattato locazioni>
164 ; REG SAVED : HL
165 ; Carica nel registro A il valore delle locazioni (HL-1)..(HL)
166 ; compattati nel nibble alto e basso
167 ;
168 802A LDAX2 E5          PUSH HL           ; Salva nello stack HL
169                                     ; <Continua eseguendo la LDAX2B>
170
171
172 ; LDAX2B $802B
173 ; IN : HL=<locazione più alta>
174 ; OUT: A=<valore compattato locazioni>
175 ; REG SAVED : NONE (RELOAD HL FROM STACK)
176 ; Carica nel registro A il valore delle locazioni (HL-1)..(HL)
177 ; compattati nel nibble alto e basso
178 ;
179 802B LDAX2B 7E         LD A,(HL)        ; Legge in A il valore puntato da HL
180 802C      07          RLCA              ; Ruota di 4 bit a sinistra
181 802D      07          RLCA              ; il valore di A
182 802E      07          RLCA              ; per occupare il
183 802F      07          RLCA              ; nibble alto
184 8030      2D          DEC L             ; decrementa il puntatore HL
185 8031      B6          OR (HL)          ; Esegue OR con valore puntato da
HL
186                                     ; HL e posiziona il dato nel
187                                     ; nibble basso
188 8032          E1          POP HL           ; Recupera HL dallo stack
189 8033          C9          RET              ; Termina
190
191
192 ; SVADAT $8034
193 ; IN : A=<valore compattato>
194 ; REG SAVED : AF HL BC
195 ; Salva il valore del registro A nell'area dati
196 ; separando i nibble alto e basso
197 ;
198 8034 SVADAT F5          PUSH AF           ; Salva nello stack AF
199 8035      E5          PUSH HL           ; Salva nello stack HL
200 8036      C5          PUSH BC           ; Salva nello stack BC
201 8037      21 01 00     LD HL,BDISP2     ; Carica in HL il valore
202                                     ; BDISP2 ($0001)
203 803A      18 03      JR SVAXB         ; Passa a SVAXB ($803F)
204

```

```

205
206 ; SVAX $803C
207 ; IN : HL=<locazione più alta> A=<valore compattato>
208 ; REG SAVED : AF HL BC
209 ; Salva il valore del registro A nelle locazioni (HL-1)..(HL)
210 ; separando i nibble alto e basso
211 ;
212 803C SVAX F5 PUSH AF ; Salva nello stack AF
213 803D E5 PUSH HL ; Salva nello stack HL
214 803E C5 PUSH BC ; Salva nello stack BC
215 ; <Continua eseguendo la SVAXB>
216
217
218 ; SVAXB $803F
219 ; IN : HL=<locazione più alta> A=<valore compattato>
220 ; REG SAVED : NONE (RELOAD BC HL AF FROM STACK)
221 ; Salva il valore del registro A nelle locazioni (HL-1)..(HL)
222 ; separando i nibble alto e basso
223 ;
224 803F SVAXB 0E 02 LD C,$02 ; Carica $02 in C, due display
225 8041 F5 PUSH AF ; Salva nello stack AF
226 8042 E6 F0 AND $F0 ; Isola il nibble alto di A
227 8044 0F RRCA ; Ruota a destra di 4 bit
228 8045 0F RRCA ; il valore di A
229 8046 0F RRCA ; per spostare il nibble alto
230 8047 0F RRCA ; nel nibble basso
231 8048 CD 15 80 CALL SHFTX ; Esegue la SHFTX ($8015)
232 ; Shift a sx di due display e
; inserimento A
233 804B F1 POP AF ; Recupera AF dallo stack
234 804C E6 0F AND $0F ; Isola il nibble basso di A
235 804E CD 15 80 CALL SHFTX ; Esegue la SHFTX ($8015)
236 ; Shift a sx di due display e
; inserimento A
237 8051 C1 POP BC ; Recupera BC dallo stack
238 8052 E1 POP HL ; Recupera HL dallo stack
239 8053 F1 POP AF ; Recupera AF dallo stack
240 8054 C9 RET ; Termina
241
242
243 ; LDDEADD $8055
244 ; OUT: DE=<valore compattato area indirizzi>
245 ; REG SAVED : AF HL
246 ; Carica nel registro DE il valore ottenuto da BDISP4 a BDISP7 (Area Indirizzi)
247 ; compattati nel nibble alti e bassi
248 ;
249 8055 LDDEADD F5 PUSH AF ; Salva AF nello stack
250 8056 E5 PUSH HL ; Salva HL nello stack
251 8057 21 04 00 LD HL,BDISP5 ; Carica BDISP5 in HL ($0004)
252 805A CD 2A 80 CALL LDAX2 ; Chiama la LDAX2 ($802A)
253 ; Restituisce in A il valore di
BDISP5 e BDISP4
254 ; parte bassa area indirizzi
255 805D 5F LD E,A ; Copia A in E
256 805E 2C INC HL ; Incrementa HL
257 805F 2C INC HL ; Incrementa HL
258 8060 CD 2A 80 CALL LDAX2 ; Chiama la LDAX2 ($802A)
259 ; Restituisce in A il valore di
BDISP7 e BDISP6
260 ; parte alta area indirizzi
261 8063 57 LD D,A ; Copia A in D
262 8064 E1 POP HL ; Recupera HL dallo stack
263 8065 F1 POP AF ; Recupera AF dallo stack
264 8066 C9 RET ; Termina
265
266
267 ; CLS $8067
268 ; REG SAVED : HL BC
269 ; Salva il valore costante $11 nelle locazioni

```

```

270 ; $0000..$0007 (per spegnere display)
271 ;
272 8067 CLS 3E 11 LD A,$11 ; Codice carattere 'blank' in A
273 ; <Continua eseguendo la SAVEALL>
274
275
276 ; SAVEALL $8069
277 ; IN : A=<valore assegnato>
278 ; REG SAVED : HL BC
279 ; Salva il valore di A nelle locazioni $0000..$0007 (buffer display)
280 ;
281 8069 SAVEALL E5 PUSH HL ; Salva HL nello stack
282 806A C5 PUSH BC ; Salva BC nello stack
283 806B 21 08 00 LD HL,BDISP7+1 ; Carica BDISP+1 in HL ($0008)
284 806E 0E 08 LD C,$08 ; Numero display = 8 in C
285 8070 LBB001 2D DEC L ; Decrementa L
286 8071 0D DEC C ; Decrementa C, numero display
287 8072 77 LD (HL),A ; Salva A in memoria puntata da HL
288 8073 20 FB JR NZ,LBB001 ; Se C diverso da zero,
289 ; passa a LBB001 ($8070)
290 8075 C1 POP BC ; Recupera BC dallo stack
291 8076 E1 POP HL ; Recupera HL dallo stack
292 8077 C9 RET ; Termina
293
294
295 ; SVDEADD $8078
296 ; IN : DE=<valore compattato locazioni>
297 ; REG SAVED : AF HL BC
298 ; Salva il valore del registro DE in memoria da BDISP4 a BDISP7 (Area Indirizzi)
299 ; separando i nibble alti e bassi
300 ; I dati non sono visualizzati immediatamente
301 ;
302 8078 SVDEADD F5 PUSH AF ; Salva AF nello stack
303 8079 E5 PUSH HL ; Salva HL nello stack
304 807A C5 PUSH BC ; Salva BC nello stack
305 807B 21 06 00 LD HL,BDISP7 ; Carica BDISP7 in HL ($0006)
306 807E 7A LD A,D ; Copia D in A
307 807F 0E 02 LD C,$02 ; Numero di display C=$02
308 8081 CD 3C 80 CALL SVAX ; Esegue la SVAX ($803C)
309 ; Salva D in parte alta
310 ; area indirizzi
311 8084 2D DEC L ; Decrementa L
312 8085 2D DEC L ; Decrementa L
313 8086 7B LD A,E ; Copia E in A
314 8087 CD 3C 80 CALL SVAX ; Esegue la SVAX ($803C)
315 ; Salva E in parte bassa
316 ; area indirizzi
317 808A C1 POP BC ; Recupera BC dallo stack
318 808B E1 POP HL ; Recupera HL dallo stack
319 808C F1 POP AF ; Recupera AF dallo stack
320 808D C9 RET ; Termina
321
322
323 ; DELY20 $808E
324 ; IN : (DELYTL/DELYTH)=Valore ritardo in ms
325 ; REG SAVED : HL
326 ; Ritardo nell'esecuzione di un numero di ms contenuto in ($0020..$0021)
327 ;
328 808E DELY20 E5 PUSH HL ; Salva HL nello stack
329 808F 2A 20 00 LD HL,(DELYTL) ; Carica in HL ritardo
330 ; in ms da DELYTL/DELYTH
331 8092 18 01 JR DELYHLX ; Passa alla DELYHLX ($8095)
332
333
334 ; DELYHL $8094
335 ; IN : HL=Valore ritardo in ms
336 ; REG SAVED : HL
337 ; Ritardo nell'esecuzione di un numero di ms contenuto in HL
338 ;

```

```

339 8094 DELYHL E5 PUSH HL ; Salva HL nello stack
340 ; <Continua eseguendo la DELYHLX>
341
342
343 ; DELYHLX $8095
344 ; IN : HL=Valore ritardo in ms
345 ; REG SAVED : AF (RELOAD HL FROM STACK)
346 ; Ritardo nell'esecuzione di un numero di ms contenuto in HL
347 ;
348 ;
349 8095 DELYHLX F5 PUSH AF ; Salva AF nello stack
350 8096 LCC001 CD D6 80 CALL DELY1 ; Esegue la DELY1 ($80D6)
351 8099 2B DEC HL ; Decrementa HL
352 809A 7C LD A,H ; Carica H in A
353 809B B5 OR L : Esegue OR con L
354 809C 20 F8 JR NZ,LCC001 ; Se HL è diverso da zero,
355 ; passa LCC001 ($8096)
356 809E F1 POP AF ; Recupera AF dallo stack
357 809F E1 POP HL ; Recupera HL dallo stack
358 80A0 C9 RET ; Termina
359
360
361 ; VISK $80A1
362 ; OUT: A=<codice tasto>
363 ; REG SAVED : NONE
364 ; Aggiorna i display e attende la pressione di un tasto, confrontandolo con Ctrl-0
365 ;
366 80A1 VISK CD EC 80 CALL VISUAL ; Chiama la VISUAL ($80EC),
367 ; visualizza contenuto buffer
368 80A4 CD AA 80 CALL GETKEY ; Esegue la GETKEY ($80AA),
369 ; attende un tasto
370 80A7 FE 10 CP CTRL0 ; Confronta con Control 0 ($10)
371 80A9 C9 RET ; Termina
372
373
374 ; GETKEY $80AA
375 ; OUT: A=<codice tasto>
376 ; REG SAVED : BC HL
377 ; Attende la pressione di un tasto, lo decodifica e lo restituisce
378 ;
379 80AA GETKEY C5 PUSH BC ; Salva BC nello stack
380 80AB E5 PUSH HL ; Salva HL nello stack
381 80AC LDD001 21 56 04 LD HL,$0456 ; Carica in HL costante ($0456)
382 80AF LDD002 DB F0 IN A,($F0) ; Input in A dato da
383 ; porta $F0 (tastiera)
384 80B1 CB 7F BIT 7,A ; Controlla STROBE (bit 7 di A)
385 ; Se uguale a 1,
386 ; è stato attivato un tasto
387 80B3 28 F7 JR Z,LDD001 ; Se uguale a zero,
388 ; passa a LDD001 ($80AC)
389 ; per ripetere la lettura da tastiera
390 80B5 2B DEC HL ; Decrementa costante in HL
391 80B6 7C LD A,H ; Copia H in A/B
392 80B7 B5 OR L ; esegue OR con L
393 80B8 20 F5 JR NZ,LDD002 ; Se HL diverso da zero,
394 ; passa a LDD002 ($80AF)
395 ; STROBE deve essere valido
396 ; per un certo tempo
397 ; Eventuali falsi contatti
398 ; sono filtrati
399 80BA LDD003 DB F0 IN A,($F0) ; Input in A dato da
400 ; porta $F0 (tastiera)
401 80BC CB 7F BIT 7,A ; Controlla STROBE (bit 7 di A)
402 ; Se uguale a 1, è stato attivato
un tasto
403 80BE 20 FA JR NZ,LDD003 ; Se diverso da zero passa
404 ; a LDD003 ($80BA)
405 ; Attende che lo STROBE sia
disattivato

```

```

406 80C0          47          LD B,A          ; Copia A in B, codice tasto premuto
407 80C1          21 05 00    LD HL,$0005    ; Carica $0005 in HL per ritardo 5 ms
408 80C4          CD 94 80    CALL DELYHL    ; Chiama la DELYHL ($8094)
409 80C7          LDD004 DB F0    IN A,($F0)    ; Input in A dato
410                                     ; da porta $F0 (tastiera)
411 80C9          4F          LD C,A          ; Copia A in C, codice tasto premuto
412 80CA          E6 7F          AND $7F        ; Maschera i primi 7 bit
413                                     ; del codice tasto
414 80CC          B8          CP B            ; Confronta A con B
415 80CD          20 F8          JR NZ,LDD004   ; Se diversi passa a LDD004 ($B0C7)
416                                     ; Attende che il tasto sia uguale
417 80CF          CB 79          BIT 7,C        ; Test STROBE contenuto in C (bit 7)
418 80D1          20 D9          JR NZ,LDD001   ; Se è diverso da zero passa
419                                     ; a LDD001 ($80AC)
420                                     ; Lettura non valida, ripete
421 80D3          E1          POP HL         ; Recupera HL dallo stack
422 80D4          C1          POP BC         ; Recupera BC dallo stack
423 80D5          C9          RET             ; Termina con il codice tasto in A
424
425
426 ; DELY1 $80D6
427 ; REG SAVED : AF DE
428 ; Ritardo nell'esecuzione di un ms a 1,920 MHz
429 ;
430 80D6          DELY1 F5          PUSH AF        ; Salva AF nello stack
431 80D7          D5          PUSH DE        ; Salva DE nello stack
432 80D8          11 47 00    LD DE,$0047   ; Carica costante $0047 in
433                                     ; DE per ritardo
434 80DB          LEE001 1B          DEC DE         ; Decrementa DE
435 80DC          7A          LD A,D         ; Copia D in A
436 80DD          B3          OR E            ; Esegue OR con E
437 80DE          20 FB          JR NZ,LEE001   ; Se DE diverso da zero passa a
438                                     ; LEE001 ($80DB), loop
439 80E0          D1          POP DE         ; Recupera DE dallo stack
440 80E1          F1          POP AF         ; Recupera AF dallo stack
441 80E2          C9          RET             ; Termina
442
443
444 ; DELY5S $80E3
445 ; REG SAVED : HL
446 ; Ritardo nell'esecuzione di 5 sec (5000 ms)
447 ;
448 80E3          DELY5S E5          PUSH HL        ; Salva HL nello stack
449 80E4          21 88 13    LD HL,$1388   ; Carica la costante $1388 (5000) ms
450 80E7          CD 94 80    CALL DELYHL    ; Chiama la DELYHL ($8094)
451                                     ; Esegue ritardo di HL ms
452 80EA          E1          POP HL         ; Recupera HL dallo stack
453 80EB          C9          RET             ; Termina
454
455
456 ; VISUAL $80EC
457 ; IN : ($0000..$0007)=Valori da visualizzare su display
458 ; REG SAVED : BC DE HL
459 ; Visualizza sui display i caratteri attuali dalle locazioni $0000..$0007
460 ;
461 80EC          VISUAL C5          PUSH BC        ; Salva BC nello stack
462 80ED          D5          PUSH DE        ; Salva DE nello stack
463 80EE          E5          PUSH HL        ; Salva HL nello stack
464 80EF          F5          PUSH AF        ; Salva AF nello stack
465 80F0          01 F7 00    LD BC,PORTD8  ; Carica PORTD8 in BC ($00F7)
466 80F3          11 07 00    LD DE,BDISP8  ; Carica BDISP7 in DE ($0007)
467 80F6          LFF001 21 87 81    LD HL,DTABLE  ; Carica DTABLE in HL ($8187),
468                                     ; indirizzo tabella display
469 80F9          1A          LD A,(DE)      ; Carica in A il valore puntato
470                                     ; da DE in A il
471                                     ; codice
472 80FA          85          ADD A,L        ; carattere nel buffer display
473                                     ; Somma L ad A

```

```

473 80FB          6F          LD L,A          ; Copia A in L
474                                     ; Ottiene in HL indirizzo
475                                     ; in tabella a partire da A
476 80FC          7E          LD A,(HL)       ; Carica in A valore puntato da HL
477                                     ; Ottiene codice a 7 segmenti
478                                     ; da tabella
479 80FD          ED 79       OUT C,A         ; Out di A su porta corrente
480                                     ; su porta contenuta in C
481 80FF          1D          DEC E           ; Decrementa E, punta a
482                                     ; buffer display precedente
483 8100          0D          DEC C           ; Decrementa C, punta a porta
484                                     ; display precedente
485 8101          79          LD A,C         ; Copia C in A
486 8102          FE EF       CP PORTD0-1     ; Confronta con PORTD0-1 ($EF)
487 8104          20 F0       JR NZ,LFF001       ; Se diverso passa a LFF001
488                                     ; ($80F6), display precedente
489 8106          F1          POP AF          ; Recupera AF dallo stack
490 8107          E1          POP HL          ; Recupera HL dallo stack
491 8108          D1          POP DE          ; Recupera DE dallo stack
492 8109          C1          POP BC          ; Recupera BC dallo stack
493 810A          C9          RET             ; Termina
494
495
496 ;   CKKEYS $810B
497 ;   IN : A=Valore codice tasto
498 ;   REG SAVED : NONE
499 ;   Controlla codice tasto presente in A.
500 ;   Se nessun tasto valido, ritorna senza operare, altrimenti
501 ;   scarta indirizzo di rientro e passa a codice specifico
502 ;
503 810B          CKKEYS 33          INC SP          ; Incrementa SP (Stack Pointer)
504 810C          33          INC SP          ; Incrementa SP (Stack Pointer)
505                                     ; Scarta indirizzo rientro per JP
506 810D          FE 10       CP CTRL0       ; Confronta con Control 0 ($10)
507 810F          38 28       JR C,LGG001     ; Se uguale passa a LGG001 ($8139)
508                                     ; Premuto Ctrl 0,
509                                     ; ritorna al chiamante
510 8111          FE 11       CP CTRL1       ; Confronta con Control 1 ($11)
511 8113          CA D4 81     JP Z,HDLC1     ; Se uguale passa a HDLC1 ($81D4)
512 8116          FE 12       CP CTRL2       ; Confronta con Control 2 ($12)
513 8118          CA 16 82     JP Z,HDLC2     ; Se uguale passa a HDLC2 ($8216)
514 811B          FE 14       CP CTRL4       ; Confronta con Control 4 ($14)
515 811D          CA 96 82     JP Z,HDLC4     ; Se uguale passa a HDLC4 ($8296)
516 8120          FE 13       CP CTRL3       ; Confronta con Control 3 ($13)
517 8122          CA FB 82     JP Z,HDLC3     ; Se uguale passa a HDLC3 ($82FB)
518 8125          FE 17       CP CTRL7       ; Confronta con Control 7 ($17)
519 8127          CA 00 84     JP Z,HDLC7     ; Se uguale passa a HDLC7 ($8400)
520 812A          FE 15       CP CTRL5       ; Confronta con Control 5 ($15)
521 812C          CA 0A 83     JP Z,HDLC5     ; Se uguale passa a HDLC5 ($830A)
522 812F          FE 16       CP CTRL6       ; Confronta con Control 6 ($16)
523 8131          CA 5F 83     JP Z,HDLC6     ; Se uguale passa a HDLC6 ($835F)
524 8134          FE 1F       CP CTRLF       ; Confronta con Control F ($1F)
525 8136          CA BD 83     JP Z,HDLCF     ; Se uguale passa a HDLCF ($83BD)
526 8139          LGG001 3B          DEC SP          ; Decrementa SP (Stack Pointer)
527 813A          3B          DEC SP          ; Decrementa SP (Stack Pointer)
528                                     ; Recupera indirizzo rientro
529 813B          C9          RET             ; Termina
530
531
532 ;   SVHLADD $813C
533 ;   IN : HL=<valore compattato>
534 ;   REG SAVED : NONE
535 ;   Salva il valore del registro HL nell'area indirizzi
536 ;   separando i nibble alti e bassi
537 ;
538 813C          SVHLADD EB          EX DE,HL        ; Scambia i registri HL e DE
539 813D          CD 78 80       CALL SVDEADD    ; Chiama la SVDEADD ($8078)
540                                     ; Carica il valore di DE nell'area
541                                     ; indirizzi

```

```

541 8140          CD EC 80          CALL VISUAL          ; Chiama la VISUAL ($80EC),
542                                     ; visualizza contenuto buffer
543 8143          EB                EX DE,HL                ; Scambia i registri HL e DE
544 8144          C9                RET                    ; Termina
545
546
547 ; TAPEO      $8145
548 ;           IN : C=<byte inviato a tape>
549 ;           REG SAVED : DE
550 ;           Invia il byte contenuto in C verso l'interfaccia tape per scriverlo
551 ;
552 8145          TAPEO      D5          PUSH DE                ; Salva registri DE nello stack
553 8146          11 D7 0A          LD DE,$0AD7            ; Carica in DE valore per timeout
554 8149          LHH002  DB EF          IN A,($EF)            ; Legge status da porta $EF
555 814B          CB 5F          BIT 3,A                ; Test bit 3
556                                     ; TRANSMITTER HOLDING REGISTER
557                                     ; EMPTY (THRE)
558                                     ; 0=Busy 1=Ready (Negato in hw)
559 814D          20 0D          JR NZ,LHH001            ; Se diverso da zero, passa
560                                     ; a LHH001 ($815C)
561 814F          1B                DEC DE                ; Decrementa contatore timeout
562 8150          7A                LD A,D                ; Controllo contatore timeout
563 8151          B3                OR E                ; a 16 bit
564 8152          20 F5          JR NZ,LHH002            ; Se diverso da zero, ritenta e
565                                     ; passa a LHH002 ($8149)
566 8154          3E 08          LD A,$08                ; Codice d'errore $08 (Timeout) in A
567 8156          D1                POP DE                ; Recupera DE dallo stack
568 8157          33                INC SP                ; Elimina indirizzo
569                                     ; di rientro
570 8158          33                INC SP                ; dallo stack
571 8159          C3 38 83        JP ERR                ; Passa alla ERR ($8338) per
572                                     ; visualizzare codice
573                                     ; errore in A
574 815C          LHH001  79          LD A,C                ; Copia il byte da trasmettere
575                                     ; da C in A
576 815D          D3 EE          OUT ($EE),A            ; Out del dato da trasmettere
577                                     ; su porta $EE
578 815F          D1                POP DE                ; Recupera registri DE dallo stack
579 8160          C9                RET                    ; Termina
580
581
582 ; TAPEI      $8161
583 ;           IN : D=<00010001 per TAPE A, 00001001 per TAPE B>
584 ;           OUT: A=<byte letto da tape>
585 ;           REG SAVED : NONE
586 ;           Legge un byte dall'interfaccia tape
587 ;
588 8161          TAPEI      DB EF          IN A,($EF)            ; Legge status da porta $EF
589 8163          CB 67          BIT 4,A                ; Test bit 4
590                                     ; DATA AVAILABLE (DA)
591                                     ; 0=Busy 1=Received (Negato in hw)
592 8165          28 FA          JR Z,TAPEI            ; Se uguale a zero,
593                                     ; passa a TAPEI ($8161)
594                                     ; e blocca il sistema
595                                     ; fino alla ricezione
596 8167          E6 07          AND $07                ; Maschera i bit 2 1 e 0
597 8169          28 05          JR Z,LII001            ; Se tutti e tre i bit a zero,
598                                     ; passa a LII001 ($8170)
599 816B          33                INC SP                ; Elimina indirizzo di
rientro
600 816C          33                INC SP                ; dallo stack
601 816D          C3 38 83        JP ERR                ; Passa alla ERR ($8338)
602                                     ; per visualizzare errore in A
603 8170          LII001  7A          LD A,D                ; Passa parametro in A
604                                     ; 00010001 per TAPE A, 00001001 per
TAPE B
605                                     ; Motore A/B acceso e reset DA
606 8171          D3 EF          OUT ($EF),A            ; Out del comando su porta $EF
607 8173          DB EE          IN A,($EE)            ; Input del dato ricevuto da porta

```

```

$EE
608 8175          C9          RET          ; Termina
609
610
611 8176          UNUSED DF A4
612
613
614 8178          ; Tabella usata nell'handler di Control 2
615          ;
616 8178          TCTRL2 0A 0F          ; 'AF'
617 817A          0B 0C          ; 'BC'
618 817C          0D 0E          ; 'DE'
619 817E          16 15          ; 'LH'
620 8180          01 05 17        ; 'SP'
621 8183          01 17 0C        ; 'PC'
622 8186          00
623
624 8187          ; Tabella conversione display sette segmenti in C1
625          ;
626 8187          DTABLE C0          ; '0' COD $00
627 8188          F9          ; '1' COD $01
628 8189          A4          ; '2' COD $02
629 818A          B0          ; '3' COD $03
630 818B          99          ; '4' COD $04
631 818C          92          ; '5' COD $05
632 818D          82          ; '6' COD $06
633 818E          F8          ; '7' COD $07
634 818F          80          ; '8' COD $08
635 8190          90          ; '9' COD $09
636 8191          88          ; 'A' COD $0A
637 8192          83          ; 'B' COD $0B
638 8193          C6          ; 'C' COD $0C
639 8194          A1          ; 'D' COD $0D
640 8195          86          ; 'E' COD $0E
641 8196          8E          ; 'F' COD $0F
642 8197          BF          ; '-' COD $10
643 8198          FF          ; BLANK COD $11
644 8199          AB          ; ' ' COD $12
645 819A          9D          ; ' ' COD $13
646 819B          B6          ; ' ' COD $14
647 819C          89          ; ' ' COD $15
648 819D          C7          ; ' ' COD $16
649 819E          8C          ; ' ' COD $17
650 819F          A7          ; 'c' COD $18
651 81A0          C2          ; ' ' COD $19
652 81A1          C8          ; ' ' COD $1A
653 81A2          98          ; ' ' COD $1B
654 81A3          AF          ; ' ' COD $1C
655 81A4          87          ; ' ' COD $1D
656 81A5          C1          ; ' ' COD $1E
657 81A6          E3          ; ' ' COD $1F
658 81A7          E1          ; ' ' COD $20
659 81A8          91          ; ' ' COD $21
660 81A9          FF          ; BLANK COD $22
661 81AA          FF          ; BLANK COD $23
662
663
664          ; MAIN $81AB
665          ; REG SAVED : NONE
666          ; Punto di ingresso della ROM
667          ; (from reset entry point)
668          ;
669 81AB          MAIN 21 00 00      LD HL,$0000          ; Azzera HL
670 81AE          22 22 00      LD (TAPECL),HL      ; Azzera contatore tape ($0022/$0023)
671 81B1          2D          DEC L          ; Decrementa L per avere $FF
672 81B2          LJ001 25      DEC H          ; e decrementa H (pagina corrente)
673 81B3          7E          LD A,(HL)       ; Legge in A dal puntatore corrente
674 81B4          2F          CPL          ; Complementa A
675 81B5          77          LD (HL),A       ; Salva A in memoria al puntatore

```

```

corrente
676 81B6 BE CP (HL) ; Confronta A con byte puntato
677 ; da puntatore corrente
678 81B7 20 F9 JR NZ,LJJ001 ; Se diverso, non è memoria
scrivibile
679 ; e passa a LJJ001 ($B1B2)
680 81B9 2F CPL ; RAM trovata, complementa il
valore cella
681 81BA 77 LD (HL),A ; Riscrive vecchio valore
682 ; al puntatore corrente
683 81BB 23 INC HL ; Aumenta HL, per top indirizzo
684 ; dello Stack Pointer
685 81BC F9 LD SP,HL ; Assegna nuovo indirizzo top
686 ; dello Stack Pointer
687 81BD 21 66 00 LD HL,$0066 ; NMI entry point ($0066)
688 81C0 22 10 00 LD (IMGSP),HL ; Salva HL in IMGSP/H ($0010/$0011)
689 81C3 3E C3 LD A,$C3 ; Istruzione JP
690 81C5 32 38 00 LD (EPINT),A ; ($0038) INT
691 81C8 32 66 00 LD (EPNMI),A ; ($0066) NMI
692 81CB 21 B9 82 LD HL,NMIINT ; ($82B9) Punto ingresso NMIINT
693 81CE 22 39 00 LD (EPINT+1),HL ; ($0039) Operando JP $82B9
694 81D1 22 67 00 LD (EPNMI+1),HL ; ($0067) Operando JP $82B9
695 ; <Continua eseguendo la HDLC1>
696
697 ; HDLC1 $81D4
698 ; REG SAVED : NONE
699 ; Gestione selezione tasto Control 1
700 ; Lettura/Modifica valori locazioni di memoria
701 ;
702 81D4 HDLC1 3E 10 LD A,$10 ; Carattere '-' in A
703 81D6 CD 69 80 CALL SAVEALL ; Chiama la SAVEALL ($8069)
704 ; Copia A nel buffer di tutti i
display
705 81D9 21 07 00 LD HL,BDISP8 ; Carica BDISP8 in HL ($0007)
706 81DC 36 12 LD (HL),$12 ; Carica $12 'n' in (HL)
707 81DE LKK002 CD EC 80 CALL VISUAL ; Chiama la VISUAL ($80EC),
708 ; visualizza contenuto buffer
709 81E1 CD AA 80 CALL GETKEY ; Chiama la GETKEY ($80AA),
710 ; attende un tasto
711 81E4 CD 0B 81 CALL CKKEYS ; Chiama la CKKEYS ($810B),
712 ; controlla tasti menu Control
713 81E7 FE 10 CP CTRL0 ; Confronta con Control 0 ($10)
714 81E9 28 05 JR Z,LKK001 ; Se uguale a Control 0 passa a
LKK001 ($81F0)
715 81EB CD 0C 80 CALL SHFADD ; Chiama la SHFADD ($800C)
716 ; Shift a sinistra display area
indirizzi
717 ; e inserisce in display a destra
718 ; il valore di A proveniente dalla
GETKEY
719 81EE 18 EE JR LKK002 ; Passa a LKK002 ($81DE), ripete loop
720 81F0 LKK001 CD 55 80 CALL LDDEADD ; Chiama la LDDEADD ($8055)
721 ; Carica in DE il valore
dell'indirizzo da display
722 81F3 1A LD A,(DE) ; Carica in A il valore puntato da
(DE)
723 81F4 CD 34 80 CALL SVADAT ; Chiama la SVADAT ($8034)
724 ; Salva il valore di A nei display
1 e 2
725 ; per visualizzare il dato relativo
all'indirizzo
726 81F7 LKK004 CD EC 80 CALL VISUAL ; Chiama la VISUAL ($80EC),
visualizza contenuto buffer
727 81FA CD AA 80 CALL GETKEY ; Chiama la GETKEY ($80AA), attende
un tasto
728 81FD CD 0B 81 CALL CKKEYS ; Chiama la CKKEYS ($810B),
controlla tasti menu Control
729 8200 FE 10 CP CTRL0 ; Confronta con Control 0 ($10)
730 8202 28 05 JR Z,LKK003 ; Se è uguale a Control 0 passa a

```



778					; il valore di A proveniente dalla GETKEY
779	8240	18 EE	JR LLL003		; Passa a LLL003 (\$8230),
780					; prossimo tasto per dato
781	8242	LLL002 CD 24 80	CALL LDADAT		; Chiama la LDADAT (\$8024)
782					; Carica in A il valore del dato da display
783	8245	02	LD (BC),A		; Carica A in memoria puntata da (BC)
784					; Modifica valore immagine
785					; registro corrente
786	8246	LLL007 23	INC HL		; Incrementa HL, prossimo simbolo
787					; registro in tabella
788	8247	03	INC BC		; Incrementa BC, prossimo
789					; valore immagine registro
790	8248	18 DC	JR LLL004		; Passa a LLL004 (\$8226), ripete
791					; per prossimo registro
792	824A	LLL001 23	INC HL		; Incrementa HL, prossimo simbolo
793					; registro in tabella
794	824B	CD 8D 82	CALL SYMVAL		; Chiama la SYMVAL (\$828D)
795					; Copia valore puntato da (HL)
796					; in buffer display (\$0006)
797					; Copia nei primi due display
798					; valore puntato da (BC)
799					; Parte bassa di SP/PC
800	824E	23	INC HL		; Incrementa HL, prossimo simbolo
801					; registro in tabella
802	824F	03	INC BC		; Incrementa BC, prossimo valore
803					; immagine registro
804	8250	7E	LD A, (HL)		; Carica in A il valore puntato da (HL)
805	8251	32 05 00	LD (BDISP6),A		; Carica A in BDISP6 (\$0005)
806					; Per carattere 'P' di 'SP' oppure 'C' di 'PC'
807	8254	0A	LD A, (BC)		; Carica in A valore puntato da (BC)
808	8255	E5	PUSH HL		; Salva HL nello stack
809	8256	C5	PUSH BC		; Salva BC nello stack
810	8257	21 03 00	LD HL,BDISP4		; Carica in HL BDISP4 (\$0003)
811	825A	0E 02	LD C,\$02		; Carica in C la costante
812					; \$02 (2 display)
813	825C	CD 3C 80	CALL SVAX		; Esegue la SVAX (\$803C)
814					; Salva A nei display BDISP3 BDISP2
815					; Parte alta di SP/PC
816	825F	C1	POP BC		; Recupera BC dallo stack
817	8260	E1	POP HL		; Recupera HL dallo stack
818	8261	LLL006 CD EC 80	CALL VISUAL		; Chiama la VISUAL (\$80EC),
819					; visualizza contenuto buffer
820	8264	CD AA 80	CALL GETKEY		; Esegue la GETKEY (\$80AA),
821					; attende un tasto
822	8267	CD 0B 81	CALL CKKEYS		; Chiama la CKKEYS (\$810B),
823					; controlla tasti menu Control
824	826A	FE 10	CP CTRL0		; Confronta con Control 0 (\$10)
825	826C	28 0E	JR Z,LLL005		; Se uguale a Control 0,
826					; passa a LLL005 (\$827C)
827	826E	E5	PUSH HL		; Salva HL nello stack
828	826F	C5	PUSH BC		; Salva BC nello stack
829	8270	21 03 00	LD HL,BDISP4		; Carica in HL BDISP4 (\$0003)
830	8273	0E 04	LD C,\$04		; Carica in C la costante \$04, per 4 display
831	8275	CD 15 80	CALL SHFTX		; Esegue la SHFTX (\$8015)
832					; Shift di C display da BDISP4
833	8278	C1	POP BC		; Recupera BC dallo stack
834	8279	E1	POP HL		; Recupera HL dallo stack
835	827A	18 E5	JR Z,LLL006		; Se uguale a Control 0,
836					; passa a LLL006 (\$8261), ciclo
837	827C	LLL005 CD 24 80	CALL LDADAT		; Chiama la LDADAT (\$8024)
838					; Carica in A il valore del dato da display
839	827F	0B	DEC BC		; Decrementa BC, punta a

```

840                                     ; parte bassa immagine SP
841 8280          02          LD (BC),A      ; Salva A in memoria puntata da (BC)
842                                     ; aggiorna parte bassa SP/PC
843 8281          E5          PUSH HL        ; Salva HL nello stack
844 8282          21 03 00    LD HL,BDISP4   ; Carica in HL BDISP4 ($0003)
845 8285          CD 2A 80    CALL LDAX2    ; Chiama la LDAX2 ($802A)
846                                     ; Ottiene in A parte
847                                     ; alta immagine SP/PC
848 8288          E1          POP HL         ; Recupera HL dallo stack
849 8289          03          INC BC         ; Incrementa BC, prossimo valore
850                                     ; immagine registro
851 828A          02          LD (BC),A      ; Salva A in memoria puntata da (BC)
852                                     ; aggiorna parte alta SP/PC
853 828B          18 B9      JR LLL007      ; Passa a LLL007 ($8246), ripete loop
854
855
856 ;   SYMVAL $828D
857 ;   IN : HL=<pointer to char, move to BDISP7> BC=<pointer to value display 1 2>
858 ;   REG SAVED : NONE
859 ;   Predisporre visualizzazione simbolo in BDISP7
860 ;   e relativo valore byte in BDISP1 BDISP2
861 ;
862 828D          SYMVAL 7E          LD A,(HL)      ; Carica in A byte puntato da (HL)
863 828E          32 06 00    LD (BDISP7),A    ; Salva A nel buffer display
864                                     ; BDISP7 ($0006)
865 8291          0A          LD A,(BC)      ; Carica in A valore puntato da (BC)
866 8292          CD 34 80    CALL SVADAT    ; Chiama la SVADAT ($8034)
867                                     ; Salve il valore di A nei display
                                     ; 1 e 2
868 8295          C9          RET           ; Termina
869
870
871 ;   HDLC4 $8296
872 ;   REG SAVED : NONE
873 ;   Gestione selezione tasto Control 4
874 ;   Esecuzione programma utente
875 ;
876 8296          HDLC4 2A 10 00    LD HL,(IMGSP) ; Carica in HL da IMGSP/IMGSPH
877                                     ; ($0010/$0011)
878 8299          F9          LD SP,HL       ; Copia HL in SP
879 829A          2A 12 00    LD HL,(IMGPCL) ; Carica in HL da IMGPC/IMGPC
880                                     ; ($0012/$0013)
881 829D          E5          PUSH HL        ; Salva HL nello stack
882                                     ; in modo da copiarlo in PC al RET
883 829E          CD A3 82    CALL RESTREG   ; Chiama la RESTREG ($82A3)
884                                     ; Recupera i valori dei registri HL
                                     ; DE BC AF
885 82A1          FB          EI           ; Abilita interrupt
886 82A2          C9          RET           ; Rientra nel programma utente
887
888
889 ;   RESTREG $82A3
890 ;   REG SAVED : NONE
891 ;   Recupera nei registri HL DE BC AF i valori dalle immagini in memoria
892 ;
893 82A3          RESTREG 21 08 00    LD HL,IMGREGA ; Carica IMGREGA in HL ($0008),
894                                     ; indirizzo iniziale
895 82A6          3E 03      LD A,$03     ; Carica in A la costante $03,
896                                     ; contatore coppie registri
897 82A8          LMM001 56      LD D,(HL)   ; Carica in D il valore puntato da
(HL)
898 82A9          23          INC HL         ; Incrementa HL
899 82AA          5E          LD E,(HL)     ; Carica in E il valore puntato da
(HL)
900 82AB          23          INC HL         ; Incrementa HL
901 82AC          D5          PUSH DE       ; Salva DE nello stack
902                                     ; (Salva immagine registri AF BC DE)
903 82AD          3D          DEC A         ; Decrementa A, contatore
904 82AE          20 F8      JR NZ,LMM001  ; Se diverso da zero,

```



```

966 82F4          CD 0B 81          CALL CKKEYS          ; Chiama la CKKEYS ($810B),
967                                     ; controlla tasti menu Control
968 82F7          FE 10          CP CTRL0            ; Confronta con Control 0 ($10)
969 82F9          20 F3          JR NZ,LNN001        ; Se diverso da Control 0,
970                                     ; passa a LNN001 ($82EE)
971                                     ; (attende in loop il Control 0)
972                                     ; <Continua eseguendo la HDLC3>
973
974 ; HDLC3 $82FB
975 ; REG SAVED : NONE
976 ; Gestione selezione tasto Control 3
977 ; Esecuzione codice utente passo passo
978 ;
979 82FB          HDLC3  2A 10 00      LD HL,(IMGSPH)      ; Carica in HL valore puntato
980                                     ; da IMGSPH/IMGSPH ($0010/$0011)
981 82FE          F9              LD SP,HL            ; Copia HL in SP
982 82FF          2A 12 00      LD HL,(IMGPCL)     ; Carica in HL valore puntato
983                                     ; da IMGPC/IMGPC ($0012/$0013)
984 8302          E5              PUSH HL             ; Salva indirizzo di rientro nello
stack
985                                     ; in modo da copiarlo in PC al RETN
986 8303          CD A3 82          CALL RESTREG        ; Chiama la RESTREG ($82A3)
987                                     ; Recupera i valori dei registri HL
DE BC AF
988 8306          D3 F8          OUT ($F8),A         ; Out di A su porta $F8 per execute
step
989 8308          ED 45          RETN                ; Rientra nel programma utente
990
991
992 ; HDLC5 $830A
993 ; REG SAVED : NONE
994 ; Gestione selezione tasto Control 5
995 ; Scrittura dati su tape
996 ;
997 830A          HDLC5  0E 14          LD C,$14           ; C=00010100
998                                     ; motore A on
999 830C          06 05          LD B,$05           ; 'S' per (S)ave
1000 830E          CD 8D 83          CALL TAPEIC        ; Chiama la TAPEIC ($838D)
1001 8311          CD E3 80          CALL DELY5S        ; Chiama la DELY5S ($80E3)
1002 8314          CD E3 80          CALL DELY5S        ; Chiama la DELY5S ($80E3)
1003                                     ; Attende 10 sec per avvio motore
1004 8317          2A 22 00      LD HL,(TAPECL)     ; Carica in HL da TAPECL/TAPECH
1005                                     ; ($0022/$0023)
1006                                     ; In HL puntatore ad inizio memoria
destinazione
1007 831A          11 00 00      LD DE,$0000        ; Azzerare DE
1008 831D          ED 53 22 00     LD ($0022),DE      ; Copia DE in TAPECL/TAPECH
1009                                     ; ($0022/$0023)
1010                                     ; Azzerare inizio indirizzo memoria
destinazione
1011 8321          4C              LD C,H              ; Copia H in C
1012 8322          CD 45 81          CALL TAPEO         ; Chiama la TAPEO ($8145)
1013                                     ; Scrive C su tape selezionato,
parte alta indirizzo
1014 8325          4D              LD C,L              ; Copia L in C
1015 8326          CD 45 81          CALL TAPEO         ; Chiama la TAPEO ($8145)
1016                                     ; Scrive C su tape selezionato,
1017                                     ; parte bassa indirizzo
1018 8329          LOO001 CB 52          BIT 2,D            ; Test bit 2 di B
1019 832B          20 22          JR NZ,TAPEEC       ; Se diverso da zero,
1020                                     ; passa a TAPEEC ($834F)
1021                                     ; Quando bit 2 è uguale a 1,
termina il primo K
1022                                     ; di memoria (indirizzi da $0000 a
$3FFF)
1023 832D          4E              LD C,(HL)          ; Copia in C byte puntato da (HL)
1024                                     ; Ottiene locazione di memoria
corrente da salvare
1025 832E          CD 45 81          CALL TAPEO         ; Chiama la TAPEO ($8145)

```

```

1026                                     ; Scrive C su tape selezionato
1027 8331          CD 3C 81      CALL SVHLADD      ; Chiama la SVHLADD ($813C)
1028                                     ; Visualizza l'indirizzo corrente
                                     da HL
1029 8334          23           INC HL           ; Incrementa HL, prossimo indirizzo
1030                                     ; in memoria
1031 8335          13           INC DE           ; Incrementa DE, contatore byte
1032 8336          18 F1       JR LOO001        ; Passa a LOO001 ($8329),
1033                                     ; loop prossimo byte
1034
1035
1036 ;   ERR      $8338
1037 ;   IN : A=<codice errore> BC=<indirizzo d'errore>
1038 ;   REG SAVED : NONE
1039 ;   Gestione errori
1040 ;
1041 8338          F5           PUSH AF          ; Salve AF nello stack
1042 8339          E5           PUSH BC          ; Salve BC nello stack
1043 833A          3E 10       LD A,$10        ; Codice carattere '-' in A
1044 833C          CD 69 80    CALL SAVEALL   ; Chiama la SAVEALL ($8069)
1045                                     ; Copia A nel buffer di tutti i
                                     display
1046 833F          21 07 00    LD HL,BDISP8   ; Carica BDISP8 in HL ($0007)
1047 8342          36 0E       LD (HL),$0E     ; 'E' per (E)rror
1048                                     ; Carica 'E' in BDISP8
1049 8344          D1           POP DE           ; Recupera in DE dallo stack
1050                                     ; il valore di BC
1051                                     ; BC conteneva l'indirizzo d'errore
1052 8345          CD 78 80    CALL SVDEADD   ; Chiama la SVDEADD ($8078)
1053                                     ; Salva DE nell'area indirizzi
1054 8348          F1           POP AF          ; Recupera AF dallo stack
1055 8349          32 00 00    LD (BDISP1),A  ; Carica A in BDISP1 ($0000)
1056 834C          CD EC 80    CALL VISUAL   ; Chiama la VISUAL ($80EC),
1057                                     ; visualizza contenuto buffer
1058                                     ; <Continua eseguendo la TAPEEC>
1059
1060
1061 ;   TAPEEC   $834F
1062 ;   REG SAVED : NONE
1063 ;   Parte comune finale gestione tape
1064 ;
1065 834F          TAPEEC  CD E3 80    CALL DELY5S   ; Chiama la DELY5S ($80E3)
1066                                     ; Attende 5 sec
1067 8352          3E 18       LD A,$18        ; A=00011000, motori off
1068 8354          D3 EF       OUT ($EF),A      ; Out di A su porta $EF
1069                                     ; Spegne motori tape
1070 8356          CD AA 80    CALL GETKEY   ; Esegue la GETKEY ($80AA),
1071                                     ; attende un tasto
1072 8359          AF          XOR A           ; Azzera A
1073 835A          D3 EF       OUT ($EF),A      ; Reset UART
1074 835C          C3 00 80    JP RESET     ; Esegue riavvio del sistema
1075                                     ; e passa a RESET ($8000)
1076
1077
1078 ;   HDLC6    $835F
1079 ;   REG SAVED : NONE
1080 ;   Gestione selezione tasto Control 6
1081 ;   Lettura dati da tape
1082 ;
1083 835F          HDLC6   0E 10       LD C,$10      ; C=00010000
1084                                     ; motore A on
1085 8361          06 16       LD B,$16        ; 'L' per (L)oad
1086 8363          CD 8D 83    CALL TAPEIC   ; Chiama la TAPEIC ($838D)
1087 8366          CD E3 80    CALL DELY5S   ; Chiama la DELY5S ($80E3)
1088                                     ; Attende 5 sec per avvio motore
1089 8369          D3 EF       OUT ($EF),A      ; Out di A su porta $EF
1090 836B          CD 61 81    CALL TAPEI   ; Chiama la TAPEI ($8161)
1091                                     ; Riceve byte in A da Tape
                                     selezionato

```

1092	836E		47	LD B,A	; Copia A in B
1093					; Copia dato ricevuto in B
1094					; (parte alta indirizzo BC)
1095	836F		CD 61 81	CALL TAPEI	; Chiama la TAPEI (\$8161)
1096					; Riceve byte in A da Tape selezionato
1097	8372		4F	LD C,A	; Copia A in C
1098					; Copia dato ricevuto in C
1099					; (parte bassa indirizzo BC)
1100	8373		2A 22 00	LD HL, (TAPECL)	; Carica in HL da
1101					; TAPECL/TAPECH (\$0022/\$0023)
1102					; In HL puntatore ad
1103					; inizio memoria destinazione
1104	8376		09	ADD HL,BC	; Somma BC ad HL
1105					; Ottiene indirizzo finale
1106					; per caricamento dati
1107	8377		01 00 00	LD BC,\$0000	; Azzera BC
1108	837A		ED 43 22 00	LD (TAPECL),BC	; Copia BC in
1109					; TAPECL/TAPECH (\$0022/\$0023)
1110					; Azzera inizio indirizzo
1111					; memoria destinazione
1112	837E	LPP001	CB 50	BIT 2,B	; Test bit 2 di B
1113	8380		20 CD	JR NZ,TAPEEC	; Se diverso da zero,
1114					; passa a TAPEEC (\$834F)
1115					; Quando bit 2 è uguale a 1,
1116					; termina il primo K di
1117					; memoria (indirizzi da \$0000 a \$3FFF)
1118	8382		CD 61 81	CALL TAPEI	; Chiama la TAPEI (\$8161)
1119					; Riceve byte in A da Tape selezionato
1120	8385		77	LD (HL),A	; Copia A in memoria puntata da (HL)
1121					; Byte ricevuto dal tape in memoria
1122	8386		CD 3C 81	CALL SVHLADD	; Chiama la SVHLADD (\$813C)
1123					; Visualizza l'indirizzo corrente da HL
1124	8389		03	INC BC	; Incrementa BC, contatore byte
1125	838A		23	INC HL	; Incrementa HL, prossimo
1126					; indirizzo in memoria
1127	838B		18 F1	JR LPP001	; Passa a LPP001 (\$837E),
1128					; loop prossimo byte
1129					
1130					
1131					; TAPEIC \$838D
1132					; IN : B=<codice carattere S o L> C=<codice motori iniziale>
1133					; REG SAVED : NONE
1134					; Parte comune iniziale gestione tape
1135					;
1136	838D	TAPEIC	3E 18	LD A,\$18	; A=00011000
1137	838F		D3 EF	OUT (\$EF),A	; Spegne motori
1138	8391		3E 10	LD A,\$10	; Codice carattere '-' in A
1139	8393		CD 69 80	CALL SAVEALL	; Chiama la SAVEALL (\$8069)
1140					; Copia A nel buffer di tutti i display
1141	8396		21 07 00	LD HL,BDISP8	; Carica BDISP8 in HL (\$0007)
1142	8399		36 18	LD (HL),\$18	; 'c' in DISP8
1143	839B		21 01 00	LD HL,BDISP2	; Carica BDISP2 in HL (\$0001)
1144	839E		70	LD (HL),B	; 'S' o 'L' in DISP2
1145	839F		CD EC 80	CALL VISUAL	; Chiama la VISUAL (\$80EC),
1146					; visualizza contenuto buffer
1147	83A2		2B	DEC HL	; Punta a DISP1 (\$0000)
1148	83A3		36 0B	LD (HL),\$0B	; 'b' in DISP1
1149					; (Tape B non visualizzato)
1150	83A5		16 09	LD D,\$09	; D=00001001 (per Tape B,
1151					; accende motore e reset invio)
1152	83A7		CD AA 80	CALL GETKEY	; Esegue la GETKEY (\$80AA),
1153					; attende un tasto
1154	83AA		FE 0B	CP \$0B	; Confronta con 'B'
1155	83AC		28 04	JR Z,LQQ001	; Se uguale, passa a LQQ001 (\$83B2)

1156					; (Caso Tape B)
1157	83AE		36 0A	LD (HL), \$0A	; 'A' in DISP1 (Tape A
1158					; non visualizzato)
1159	83B0		16 11	LD D, \$11	; D=00010001 (per Tape A,
1160					; accende motore e reset invio)
1161	83B2	LQQ001	79	LD A, C	; Per TAPE A, carica in A=\$14
1162					; per save e A=\$10 per load
1163	83B3		20 02	JR NZ, LQQ002	; Se diverso, passa a LQQ002
1164					; (\$83B7) (caso Tape A)
1165	83B5		D6 08	SUB \$08	; Per TAPE B, sottrae \$08,
1166					; A=\$0C per save
1167					; A=\$08 per load
1168	83B7	LQQ002	D3 EF	OUT (\$EF), A	; Uscita A su \$EF
1169					; >>> TAPE A <<<
1170					; SAVE : 00010100
1171					; LOAD : 00010000
1172					; REG D : 00010001
1173					;
1174					; >>> TAPE A <<<
1175					; SAVE : 00001100
1176					; LOAD : 00010000
1177					; REG D : 00001001
1178	83B9		CD EC 80	CALL VISUAL	; Chiama la VISUAL (\$80EC),
1179					; visualizza contenuto buffer
1180	83BC		C9	RET	; Termina
1181					
1182					
1183	; HDLCF	\$83BD			
1184	; REG SAVED :	NONE			
1185	; Gestione selezione	tasto Control F			
1186	;				
1187	83BD	HDLCF	21 04 00	LD HL, BDISP5	; Carica BDISP5 (\$0004) in HL
1188	83C0		11 33 00	LD DE, \$0033	; Carica \$0033 in DE
1189	83C3		AF	XOR A	; Azzerata A, '0'
1190	83C4		CD 69 80	CALL SAVEALL	; Chiama la SAVEALL (\$8069)
1191					; Copia A nel buffer di tutti i display
1192	83C7	LRR002	CD A1 80	CALL VISK	; Esegue la VISK (\$80A1),
1193					; visualizza e attende tasto
1194	83CA		20 02	JR NZ, LRR001	; Se tasto diverso da Ctrl 0
1195					; passa a LRR001 (\$83CE)
1196	83CC		ED 5F	LD A, R	; Carica R (random) in A
1197	83CE	LRR001	E6 0F	AND \$0F	; Isola i 4 bit bassi
1198	83D0		1C	INC E	; Incrementa E
1199	83D1		2D	DEC L	; Decrementa L
1200	83D2		12	LD (DE), A	; Carica A (random) in (DE)
1201	83D3		36 11	LD (HL), \$11	; Carica \$11 (blank) in (HL)
1202	83D5		20 F0	JR NZ, LRR002	; Se L diverso da zero,
1203					; passa a LRR002 (\$83C7)
1204	83D7	LRR004	21 07 00	LD HL, BDISP8	; Carica BDISP8 in HL (\$0007)
1205	83DA		0E 04	LD C, \$04	; Carica \$04 in C
1206	83DC		CD A1 80	CALL VISK	; Esegue la VISK (\$80A1),
1207					; visualizza e attende tasto
1208	83DF		28 05	JR Z, LRR003	; Se tasto uguale a Ctrl 0,
1209					; passa a LRR003 (\$83E6)
1210	83E1		CD 15 80	CALL SHFTX	; Esegue la SHFTX (\$8015)
1211					; Shift di C display da BDISP8
1212	83E4		18 F6	JR LRR004	; Passa a LRR004 (\$83D7)
1213	83E6	LRR003	21 03 00	LD HL, BDISP4	; Carica BDISP4 (\$0003) in HL
1214	83E9		11 33 00	LD DE, \$0033	; Carica \$0033 in DE
1215	83EC		06 05	LD B, \$05	; Carica \$05 in B
1216	83EE		0E 00	LD C, \$00	; Azzerata C
1217	83F0	LRR005	79	LD A, C	; Copia C in A
1218	83F1		32 00 00	LD (BDISP1), A	; Carica A in BDISP1 (\$0000)
1219	83F4		05	DEC B	; Decrementa B
1220	83F5		28 E0	JR Z, LRR004	; Se uguale a zero,
1221					; passa a LRR004 (\$83D7)
1222	83F7		23	INC HL	; Incrementa HL
1223	83F8		13	INC DE	; Incrementa DE

```

1224 83F9          1A          LD A, (DE)          ; Carica A da puntatore (DE)
1225 83FA          BE          CP (HL)            ; Confronta con valore puntato da
(HL)
1226 83FB          20 F3       JR NZ, LRR005       ; Se diverso, passa a LRR005 ($83F0)
1227 83FD          0C          INC C              ; Incrementa C
1228 83FE          18 F0       JR LRR005          ; Passa a LRR005 ($83F0)
1229
1230

```

```

1231 ; HDLC7 $8400
1232 ; REG SAVED : NONE
1233 ; Gestione selezione tasto Control 7
1234 ;
1235 8400 HDLC7 ; UNINPLEMENTED
1236

```

```

1237 =====

```

```

1238
1239 --- Elenco SUB/*TABLE per indirizzo

```

```

--- Elenco SUB/*TABLE alfabetico

```

1241	8000 RESET	810B CKKEYS
1242	8003 SHFDAT	8067 CLS
1243	800C SHFADD	80D6 DELY1
1244	8015 SHFTX	808E DELY20
1245	8017 SHFTXB	80E3 DELY5S
1246	8024 LDADAT	8094 DELYHL
1247	802A LDAX2	8095 DELYHLX
1248	802B LDAX2B	8187 DTABLE *
1249	8034 SVADAT	8338 ERR
1250	803C SVAX	80AA GETKEY
1251	803F SVAXB	81D4 HDLC1
1252	8055 LDDEADD	8216 HDLC2
1253	8067 CLS	82FB HDLC3
1254	8069 SAVEALL	8296 HDLC4
1255	8078 SVDEADD	830A HDLC5
1256	808E DELY20	835F HDLC6
1257	8094 DELYHL	8400 HDLC7
1258	8095 DELYHLX	83BD HDLCF
1259	80A1 VISK	8024 LDADAT
1260	80AA GETKEY	802A LDAX2
1261	80D6 DELY1	802B LDAX2B
1262	80E3 DELY5S	8055 LDDEADD
1263	80EC VISUAL	81AB MAIN
1264	810B CKKEYS	82B9 NMIINT
1265	813C SVHLADD	8000 RESET
1266	8145 TAPEO	82A3 RESTREG
1267	8161 TAPEI	8069 SAVEALL
1268	8178 TCTRL2 *	800C SHFADD
1269	8187 DTABLE *	8003 SHFDAT
1270	81AB MAIN	8015 SHFTX
1271	81D4 HDLC1	8017 SHFTXB
1272	8216 HDLC2	8034 SVADAT
1273	828D SYMVAL	803C SVAX
1274	8296 HDLC4	803F SVAXB
1275	82A3 RESTREG	8078 SVDEADD
1276	82B9 NMIINT	813C SVHLADD
1277	82FB HDLC3	828D SYMVAL
1278	830A HDLC5	834F TAPEEC
1279	8338 ERR	8161 TAPEI
1280	834F TAPEEC	838d TAPEIC
1281	835F HDLC6	8145 TAPEO
1282	838d TAPEIC	8178 TCTRL2 *
1283	83BD HDLCF	80A1 VISK
1284	8400 HDLC7	80EC VISUAL
1285		