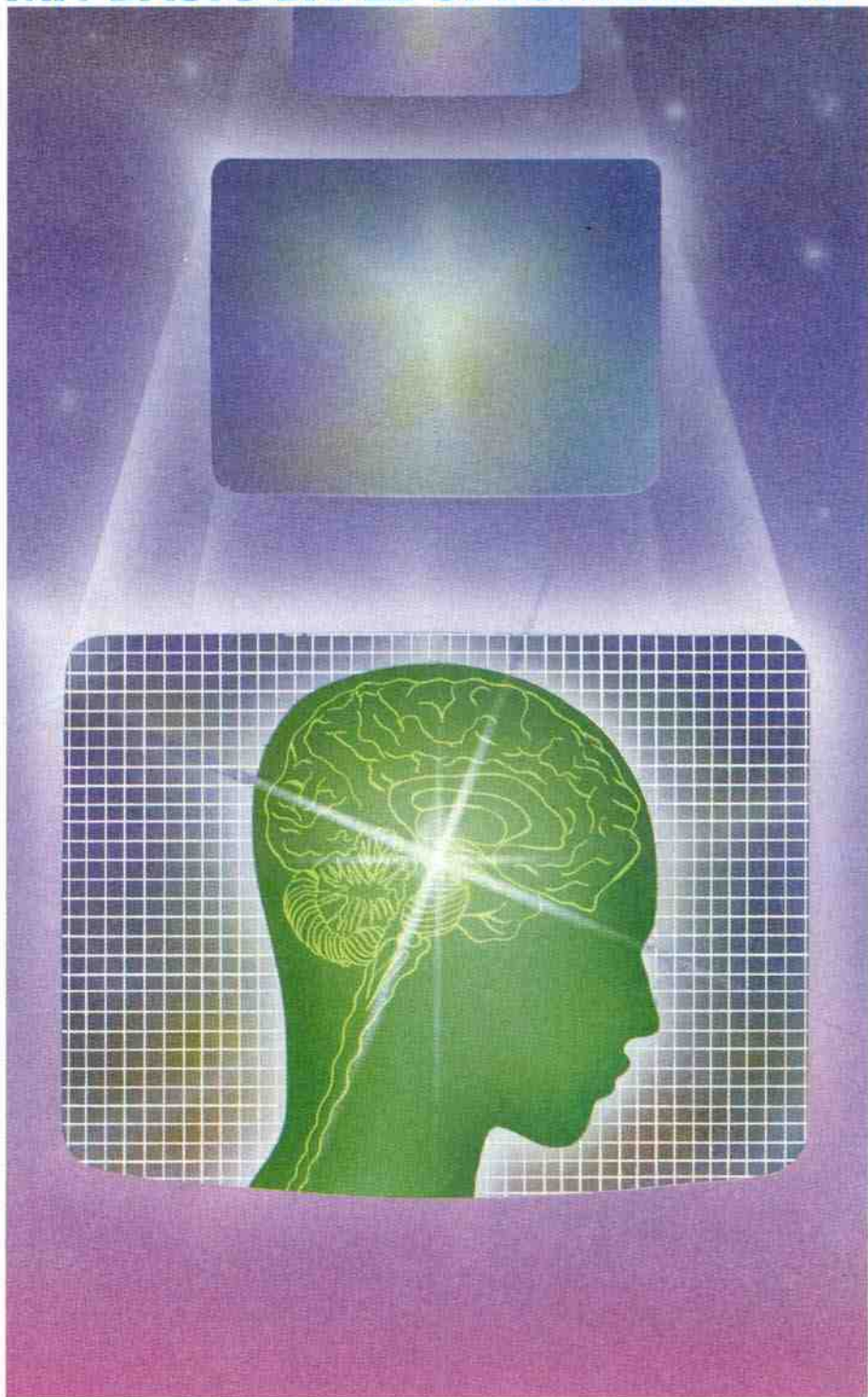


Carlos BELLVER

Ampliación del Basic

ESTE PROGRAMA, ESCRITO INTEGRAMENTE EN CODIGO MAQUINA, PERMITE USAR DOCE NUEVOS COMANDOS DENTRO DE UN PROGRAMA BASIC EN EL SPECTRUM 48 K.



Estos nuevos comandos deberán escribirse carácter a carácter (no importa si se hace en mayúsculas o en minúsculas) tras el símbolo &. Para usarlos en un programa, la primera línea de éste habrá de ser similar a la siguiente:

```
10 CLEAR 63999: LOAD ""CODE:
RANDOMIZE USR 64000
```

Si posteriormente se hace RUN, NEW o CLEAR, tendremos que ejecutar otro USR 64000 para poder volver a usar las nuevas instrucciones del EXTBASIC.

LOS NUEVOS COMANDOS

Son los siguientes:

&REPEAT

Se usa conjuntamente con &UNTIL para crear un bucle que se ejecutará hasta que se cumpla la condición que sigue a &UNTIL. Por ejemplo:

EJEMPLO 1

```
10&REPEAT
20 PRINT AT RND*21,RND*31;"*"
30&UNTIL, INKEY$="k"
```

Dibujará asteriscos en la pantalla hasta que se pulse la tecla «K».

Se pueden anidar los bucles &REPEAT del mismo modo que se hace con los FOR-NEXT, hasta un límite de ocho anidaciones.

&REPEAT ha de ser la última instrucción en una línea, y sólo se puede usar dentro de un programa, nunca en modo directo, es decir, sin un número de líneas. Si se hiciera esto no se ejecutaría nada.

&CLR

Cuando el usuario pulsa BREAK dentro de un bucle REPEAT, la pila de REPEAT (la zona de memoria en que se almacenan los números de línea a que ha de saltar UNTIL) no se borra, y si esta acción se efectúa varias veces, la pila acabará por llenarse y aparecerá un mensaje de error. Entonces habrá que usar &CLR, que borra la pila de REPEAT.

&SCREEN, num

Esta instrucción pone el BORDER y el PAPER al color indicado por la ex-

presión «num» y el INK al color que mejor contraste con éste. También pone a cero el FLASH y el BRIGHT. Resulta más rápido que BORDER num:PAPER num:INK 9:BRIGHT 0:FLASH 0:CLS, cuyo efecto es equivalente al de &SCREEN, num.

&RECOL, paper, ink

Cambia los atributos a los indicados por las expresiones «paper» e «ink», pero no altera lo que haya dibujado en la pantalla. Ejemplo:

EJEMPLO 2

```
100&SCREEN,0
110 FOR I=0 TO 703: PRINT CHR$(
32+INT(RND*96));: NEXT I
120 FOR I=0 TO 15
130 FOR J=0 TO 7
140&RECOL,0,J
150 NEXT J: NEXT I
160&SCREEN,0
```

&SCROLL

Desplaza la pantalla una línea hacia arriba, lo cual es útil en juegos sencillos o en presentaciones como ésta:

EJEMPLO 3

```
100 PRINT AT 21,0;"EXTBASIC V1
0"
110 FOR I=0 TO 21
120&SCROLL: BEEP .1,I
130 NEXT I
```

&CLSLOW

Borra la parte inferior de la pantalla, normalmente las dos últimas líneas, en las que se puede escribir mediante PRINT #0, o PRINT #1. Por ejemplo:

EJEMPLO 4

```
100 FOR i=32 TO 255
110&CLSLOW: PRINT #0;"Pulsa 'c'
para ver el CHR$ ";i
120&REPEAT
130&UNTIL, INKEY$="c"
140 PRINT CHR$(i);
150 NEXT i
```

&SOUND, f1, f2, step, dur

Produce un sonido de frecuencia «f1» (0-65535) y duración «dur» (0-65535). Suma «step» a «f1» y si el resultado es menor o igual a «f2» repite el proceso. Pueden obtenerse algunos efectos bastante buenos:

```
&SOUND,100,200,1,8
&SOUND,400,500,1,4
&SOUND,100,500,1,16
```

&NOISE, dur

Produce ruido durante un tiempo «dur». Cuando «dur» vale más de 8000 los resultados no son muy aceptables. Ejemplo:

EJEMPLO 5

```
100 BORDER 2:&NOISE,50
110 BORDER 1:&NOISE,100
120 BORDER 4:&NOISE,40
130 BORDER 7
```

&WAIT, dur

Detiene la ejecución del programa, como PAUSE, durante «dur/50» segundos. Al contrario que PAUSE no sigue la ejecución cuando se pulsa una tecla. Por ello &WAIT,0 espera más de veinte minutos...

&MOV, numbytes, org, dest

Copia un bloque de bytes de longitud «numbytes» en la dirección «org» a la dirección «dest». Su utilidad más inmediata es la de guardar pantallas en memoria y recuperarlas, pero se le pueden encontrar muchas otras.

El ejemplo muestra un caso de almacenamiento de pantallas sin atributos:

EJEMPLO 6

```
100 CLEAR 26999: RANDOMIZE USR
64000
110&SCREEN,5
120 FOR i=0 TO 5: CLS
130 CIRCLE 128,88,10*i+10
140&MOV,6144,16384,27000+6144*i
150 NEXT i
160&SOUND,100,300,1,4
170 FOR i=0 TO 5
180&MOV,6144,27000+6144*i,16384
190&NOISE,20: NEXT i
```

&DEL, line1, line2

Borra las líneas del programa Basic line1 y line2, ambas inclusive. Su utilidad es evidente.

EL PROGRAMA EXTBASIC

Este programa se basa en el hecho de que se puede cambiar la dirección de la rutina de errores a la que se salta con RST 8. Esto se hace así:

```
LD DE, NEWADD
LD HL, (ERRSP)
LD (HL),E
```


INC HL
LD (HL),D
RET

Siendo NEWADD la dirección de la nueva rutina de errores y ERRSP la variable del sistema en 23613d.

Pues bien, el programa EXTBASIC se sitúa en NEWADD (64010) y lo primero que hace es comprobar si el error es «Nonsense in Basic». Si no es así el error, no se puede deber a la introducción de una de las nuevas instrucciones. En tal caso hay que saltar a la rutina de la ROM que presenta un informe de error o a la que presenta un cursor parpadeante para señalar error de sintaxis, según estemos ejecutando una instrucción o comprobando su sintaxis (esto se sabe por el bit 7 de FLAGS, 23611d, puesto a 1 para indicar ejecución).

En caso de que si fuera error «Nonsense in Basic», el programa EXTBASIC lee el carácter que lo ha provocado y si no es «&» pasa el control a las rutinas de la ROM antes comentadas. Si efectivamente es «&» el programa lee los caracteres que hemos escrito a continuación (usando RST 32) y si coinciden con alguno de los nuevos comandos, toma los parámetros que le siguen y, si estamos en tiempo de ejecución (bit 7 de FLAGS a 1), salta a la rutina correspondiente al comando de que se trate.

Tras ello hay que volver a la ROM para comprobar o ejecutar la siguiente instrucción. Esto se hace saltando a la dirección 7030d.



Para utilizar los dos bloques de código máquina debemos teclear el primero y realizar el DUMP. Sin borrarlo de la memoria procederemos a teclear el segundo listado y hacer el DUMP correspondiente, y, por último, grabarlas en conjunto indicando como dirección la 64000 y 860 como número de bytes.

LISTADO 1 DUMP: 64.000 N.º BYTES: 330

```

1 2A3D5C110AFA732372C9 937
2 110AFAD53A3A5CFE0B20 935
3 13CD44FA210313CD3025 857
4 200421CF12E5C37E1BE1 1068
5 3A3A5CFD3600FFCD3025 1060
6 28043CC313132A5D5C22 598
7 5F5C225B5CC3B012FD36 1113
8 00FF2A3D5C5E5ED733D 1321
9 5CE1CD60FAE1223D5CFD 1533
10 CB007EC018C72A5D5C2B 1014
11 225D5C7EFE26C28A1C06 1003
12 1021015B5E5E7E1E6DFFE 1533
13 413810FE5E300C77ED5B 989
14 5D5C122310EAC38A1C3E 911
15 109032005B21DEFA1101 824
16 5BCB7E20ED3A005B4623 943
17 B8200A1ABE2006231310 550
18 F818082310FD232E2318 713
19 DF46230405280F5DFFE 1098
20 2C2001E7C5CD821CC110 1077
21 F3E1E5DFE1FE0D2804FE 1710
22 3A20B3CD3025C85E2356 974

```

```

23 EBE90653435245454E01 923
24 F4FB055245434F4C0200 875
25 FC0344454C0224FC0653 847
26 43524F4C4C00FE0D0643 720
27 4C534C4F57006E0D054E 607
28 4F495345014EFC05534F 802
29 554E440472FC04574149 830
30 5401ABFC034D4F5603B8 940
31 FC0552455045415400DE 913
32 FC05554E54494C01EDFC 1143
33 03434C520013FD800000 628

```

LISTADO 2 DUMP: 64.500 N.º BYTES: 360

```

1 CD94223A485C328D5CC3 1087
2 6B0DCDD52DFE083018F5 1165
3 CDD52DFE083013070707 813
4 47F1B021005811015801 716
5 BF0277EDB0C9CF13CDA5 1522
6 2D3823B02820606923CD 825
7 6E19E5CDA52D3814B028 1071
8 115069CD6E19545DE1E5 1189
9 A7ED52E1D4E519C9CF19 1610
10 CDA52D381D50593A485C 891
11 0F0F0F4F2100007E5618 537
12 B1D3FE060010FE231B7A 1102
13 B320F0C9CF0ACDA52D38 1340
14 F9C5CDA52D38F3ED4300 1464
15 5BCDA52D38FAED43025B 1193
16 CDA52D38E16069D1E5E5 1564
17 C12A025BA7ED42E1D8E5 1468
18 D5CDB503D1E1ED4E005B 1439
19 0918E7CDA52D38C0FB76 1296
20 0B78B120FAC9CDA52D38 1262
21 B3C5CDA52D38ADC5CDA5 1567
22 2D38A7E1D1EDB0C93A52 1456

```

```

23 FDFE08303F3C3252FD2A 1113
24 555C46234E0000002A53 485
25 FD712370232253FDC93A 1177
26 52FA72827CD052D0A728 1251
27 0D2152FD32A53FD2B2B 898
28 2253FDC92A53FD2B4E2B 1105
29 4ECD2B2DC3671EAF3252 1006
30 FD2120FD2253FDC9CF00 1349
31 0F001400000000000000 35
32 00000000000000000000 0
33 00000000000000000000 0
34 00000000000000000000 0
35 00000000000000000000 0
36 0122FD00000000000000 288

```

LISTADO ENSAMBLADOR

```

18      ORG 64000
20      LD HL,(23613)
30      LD DE,BUCLE1
40      LD (HL),E
50      INC HL
60      LD (HL),D
70      RET
80 BUCLE1 LD DE,BUCLE1
90      PUSH DE
100     LD A,(23610)
110     CP #0B
120     JR NZ,BUCLE2
130     CALL BUCLE3
140     LD HL,4867
150     CALL 9520
160     JR NZ,BUCLE4
170     LD HL,4815
180     PUSH HL
190 BUCLE4 JP 7030
200 BUCLE5 POP HL
210 BUCLE2 LD A,(23610)
220     LD (1Y+0),#FF
230     CALL 9520
240     JR 2,BUCLE6
250     INC A
260     JP 4883
270 BUCLE6 LD HL,(23645)
280     LD (23647),HL
290     LD (23643),HL
300     JP 4797
310 BUCLE3 LD (1Y+0),#FF
320     LD HL,(23613)
330     PUSH HL
340     PUSH HL
350     LD (23613),SP
360     POP HL
370     CALL BUCLE7
380     POP HL
390     LD (23613),HL
400     BIT 7,(1Y+0)
410     RET NZ
420     JR BUCLE5
430 BUCLE7 LD HL,(23645)
440     DEC HL
450     LD (23645),HL

```


| | | | | | | | | | | | |
|-----|--------|-------------|------|-------|-----------|------|------|-----------|------|--------|-----------|
| 460 | LD | A,(HL) | 970 | CP | #2C | 1480 | LD | C,H | 1990 | LD | C,H |
| 470 | CP | #26 | 980 | JR | NZ,DATA8 | 1490 | LD | D,E | 2000 | LD | D,D |
| 480 | JP | NZ,7306 | 990 | RST | #20 | 1500 | LD | C,H | 2010 | NOP | |
| 490 | LD | B,#10 | 1000 | DATA8 | PUSH BC | 1510 | LD | C,A | 2020 | INC | DE |
| 500 | LD | HL,23297 | 1010 | CALL | 7298 | 1520 | LD | D,A | 2030 | NOP | |
| 510 | BUCLE | PUSH HL | 1020 | POP | BC | 1530 | NOP | | 2040 | ADD | A,B |
| 520 | RST | #20 | 1030 | DJNZ | DATA7 | 1540 | LD | L,(HL) | 2050 | ORG | 64500 |
| 530 | POP | HL | 1040 | POP | HL | 1550 | DEC | C | 2060 | DATA8 | CALL 8852 |
| 540 | AND | #DF | 1050 | DATA6 | PUSH HL | 1560 | DEC | B | 2070 | LD | A,(23624) |
| 550 | CP | #41 | 1060 | RST | #18 | 1570 | LD | C,(HL) | 2080 | LD | (23693),A |
| 560 | JR | C,BUCLE9 | 1070 | POP | HL | 1580 | LD | C,A | 2090 | JP | 3435 |
| 570 | CP | #5B | 1080 | CP | #0D | 1590 | LD | C,C | 2100 | CALL | 11733 |
| 580 | JR | NC,BUCLE9 | 1090 | JR | Z,DATA9 | 1600 | LD | D,E | 2110 | DATA12 | CP #08 |
| 590 | LD | (HL),A | 1100 | CP | #3A | 1610 | LD | B,L | 2120 | JR | NC,DATA13 |
| 600 | LD | DE,(23645) | 1110 | JR | NZ,BUCLE8 | 1620 | LD | BC,DATA10 | 2130 | PUSH | AF |
| 610 | LD | (DE),A | 1120 | DATA9 | CALL 9520 | 1630 | DEC | B | 2140 | CALL | 11733 |
| 620 | INC | HL | 1130 | RET | Z | 1640 | LD | D,E | 2150 | CP | #08 |
| 630 | DJNZ | BUCLE | 1140 | LD | E,(HL) | 1650 | LD | C,A | 2160 | JR | NC,DATA13 |
| 640 | BUCLE8 | JP 7306 | 1150 | INC | HL | 1660 | LD | D,L | 2170 | RLCA | |
| 650 | BUCLE9 | LD A,#10 | 1160 | LD | D,(HL) | 1670 | LD | C,(HL) | 2180 | RLCA | |
| 660 | SUB | B | 1170 | EX | DE,HL | 1680 | LD | B,H | 2190 | RLCA | |
| 670 | LD | (23296),A | 1180 | JP | (HL) | 1690 | INC | B | 2200 | LD | B,A |
| 680 | LD | HL,DATA1 | 1190 | DATA1 | LD B,#53 | 1700 | LD | (HL),D | 2210 | POP | AF |
| 690 | DATA2 | LD DE,23297 | 1200 | LD | B,E | 1710 | CALL | M,22276 | 2220 | OR | B |
| 700 | BIT | 7,(HL) | 1210 | LD | D,D | 1720 | LD | B,C | 2230 | LD | HL,22528 |
| 710 | JR | NZ,BUCLE8 | 1220 | LD | B,L | 1730 | LD | C,C | 2240 | LD | DE,22529 |
| 720 | LD | A,(23296) | 1230 | LD | B,L | 1740 | LD | D,H | 2250 | LD | BC,703 |
| 730 | LD | B,(HL) | 1240 | LD | C,(HL) | 1750 | LD | BC,DATA11 | 2260 | LD | (HL),A |
| 740 | INC | HL | 1250 | LD | BC,DATA8 | 1760 | INC | BC | 2270 | LDIR | |
| 750 | CP | B | 1260 | DEC | B | 1770 | LD | C,L | 2280 | RET | |
| 760 | JR | NZ,DATA3 | 1270 | LD | D,D | 1780 | LD | C,A | 2290 | DATA13 | RST 8 |
| 770 | DATA4 | LD A,(DE) | 1280 | LD | B,L | 1790 | LD | D,(HL) | 2300 | INC | DE |
| 780 | CP | (HL) | 1290 | LD | B,E | 1800 | INC | BC | 2310 | CALL | 11685 |
| 790 | JR | NZ,DATA3 | 1300 | LD | C,A | 1810 | CP | B | 2320 | JR | C,DATA14 |
| 800 | INC | HL | 1310 | LD | C,H | 1820 | CALL | M,20998 | 2330 | OR | B |
| 810 | INC | DE | 1320 | LD | (BC),A | 1830 | LD | B,L | 2340 | JR | Z,DATA14 |
| 820 | DJNZ | DATA4 | 1330 | NOP | | 1840 | LD | D,B | 2350 | LD | H,B |
| 830 | JR | DATA5 | 1340 | CALL | M,17411 | 1850 | LD | B,L | 2360 | LD | L,C |
| 840 | DATA3 | INC HL | 1350 | LD | B,L | 1860 | LD | B,C | 2370 | INC | HL |
| 850 | DJNZ | DATA3 | 1360 | LD | C,H | 1870 | LD | D,H | 2380 | CALL | 6510 |
| 860 | INC | HL | 1370 | LD | (BC),A | 1880 | NOP | | 2390 | PUSH | HL |
| 870 | INC | HL | 1380 | INC | H | 1890 | ADC | A,#FC | 2400 | CALL | 11685 |
| 880 | INC | HL | 1390 | CALL | M,21254 | 1900 | DEC | B | 2410 | JR | C,DATA14 |
| 890 | JR | DATA2 | 1400 | LD | B,E | 1910 | LD | D,L | 2420 | OR | B |
| 900 | DATA5 | LD B,(HL) | 1410 | LD | D,D | 1920 | LD | C,(HL) | 2430 | JR | Z,DATA14 |
| 910 | INC | HL | 1420 | LD | C,A | 1930 | LD | D,H | 2440 | LD | H,B |
| 920 | INC | B | 1430 | LD | C,H | 1940 | LD | C,C | 2450 | LD | L,C |
| 930 | DEC | B | 1440 | LD | C,H | 1950 | LD | C,H | 2460 | CALL | 6510 |
| 940 | JR | Z,DATA6 | 1450 | NOP | | 1960 | LD | BC,64749 | 2470 | LD | D,H |
| 950 | PUSH | HL | 1460 | CP | #0D | 1970 | INC | BC | 2480 | LD | E,L |
| 960 | DATA7 | RST #18 | 1470 | LD | B,#43 | 1980 | LD | B,E | 2490 | POP | HL |

| | | | | | | | |
|------|-------------------|------|-------------------|------|----------------|------|-----------------------|
| 2500 | PUSH HL | 3010 | SBC HL,BC | 3290 | POP HL | 3570 | JR Z,DATA22 |
| 2510 | AND A | 3020 | POP HL | 3300 | POP DE | 3580 | LD HL,DATA18 |
| 2520 | SBC HL,DE | 3030 | RET C | 3310 | LDIR | 3590 | DEC (HL) |
| 2530 | POP HL | 3040 | PUSH HL | 3320 | RET | 3600 | LD HL,(DATA20) |
| 2540 | CALL NC,6629 | 3050 | PUSH DE | 3330 | LD A,(DATA18) | 3610 | DEC HL |
| 2550 | RET | 3060 | CALL 949 | 3340 | CP #08 | 3620 | DEC HL |
| 2560 | DATA14 RST 8 | 3070 | POP DE | 3350 | JR NC,DATA19 | 3630 | LD (DATA20),HL |
| 2570 | ADD HL,DE | 3080 | POP HL | 3360 | INC A | 3640 | RET |
| 2580 | DATA18 CALL 11685 | 3090 | LD BC,(23296) | 3370 | LD (DATA18),A | 3650 | DATA22 LD HL,(DATA20) |
| 2590 | JR C,DATA15 | 3100 | ADD HL,BC | 3380 | LD HL,(23637) | 3660 | DEC HL |
| 2600 | LD D,B | 3110 | JR DATA16 | 3390 | LD B,(HL) | 3670 | LD B,(HL) |
| 2610 | LD E,C | 3120 | DATA11 CALL 11685 | 3400 | INC HL | 3680 | DEC HL |
| 2620 | LD A,(23624) | 3130 | JR C,DATA15 | 3410 | LD C,(HL) | 3690 | LD C,(HL) |
| 2630 | RRCA | 3140 | EI | 3420 | NOP | 3700 | CALL 11563 |
| 2640 | RRCA | 3150 | DATA17 HALT | 3430 | NOP | 3710 | JP 7783 |
| 2650 | RRCA | 3160 | DEC BC | 3440 | NOP | 3720 | XOR A |
| 2660 | LD C,A | 3170 | LD A,B | 3450 | LD HL,(DATA20) | 3730 | DATA19 LD (DATA18),A |
| 2670 | LD HL,0 | 3180 | OR C | 3460 | LD (HL),C | 3740 | LD HL,DATA23 |
| 2680 | DATA24 LD A,(HL) | 3190 | JR NZ,DATA17 | 3470 | INC HL | 3750 | DATA21 LD (DATA20),HL |
| 2690 | AND #18 | 3200 | RET | 3480 | LD (HL),B | 3760 | RET |
| 2700 | OR C | 3210 | CALL 11685 | 3490 | INC HL | 3770 | RST 8 |
| 2710 | OUT (#FE),A | 3220 | JR C,DATA15 | 3500 | LD (DATA20),HL | 3780 | NOP |
| 2720 | LD B,#08 | 3230 | PUSH BC | 3510 | RET | 3790 | DATA23 CALL M,DATA12 |
| 2730 | PAUSA DJNZ PAUSA | 3240 | CALL 11685 | 3520 | LD A,(DATA18) | 3800 | INC BC |
| 2740 | INC HL | 3250 | JR C,DATA15 | 3530 | AND A | 3810 | ORG 64851 |
| 2750 | DEC DE | 3260 | PUSH BC | 3540 | JR Z,DATA21 | 3820 | DATA18 NOP |
| 2760 | LD A,D | 3270 | CALL 11685 | 3550 | CALL 11733 | 3830 | DATA20 JR NZ,DATA18 |
| 2770 | OR E | 3280 | JR C,DATA15 | 3560 | AND A | | |
| 2780 | JR NZ,DATA24 | | | | | | |
| 2790 | RET | | | | | | |
| 2800 | DATA15 RST 8 | | | | | | |
| 2810 | LD A,(BC) | | | | | | |
| 2820 | CALL 11685 | | | | | | |
| 2830 | JR C,DATA15 | | | | | | |
| 2840 | PUSH BC | | | | | | |
| 2850 | CALL 11685 | | | | | | |
| 2860 | JR C,DATA15 | | | | | | |
| 2870 | LD (23296),BC | | | | | | |
| 2880 | CALL 11685 | | | | | | |
| 2890 | JR C,DATA15 | | | | | | |
| 2900 | LD (23298),BC | | | | | | |
| 2910 | CALL 11685 | | | | | | |
| 2920 | JR C,DATA15 | | | | | | |
| 2930 | LD H,B | | | | | | |
| 2940 | LD L,C | | | | | | |
| 2950 | POP DE | | | | | | |
| 2960 | DATA16 PUSH HL | | | | | | |
| 2970 | PUSH HL | | | | | | |
| 2980 | POP BC | | | | | | |
| 2990 | LD HL,(23298) | | | | | | |
| 3000 | AND A | | | | | | |

