Programmable Characters Generator (PCG) for Sharp MZ-700
Version Extension BUS

Specifications:

- 4 complete sets of 256 characters of 8x8 bits or 1024 characters on a graphical plane
- 4 switchable graphic designs, ie 4096 characters possible but not simultaneous, possibility of animation on 4 motifs, 1 motif per plan
- 4-byte addressing from E010 to E013 replacing PCG-700 from HAL Laboratory reprogrammable in the GAL20V8

Programming :

E010 : Contains 8-bit data to encode a character line : D0-D7

E011 : Contains the A0-A7 address of the 8-bit data line

E012 : Bit 0 : A8 address
Bit 1 : A9 address
Bit 2 : A10 address
Bit 3 : Mode 0 : CGROM / Mode 1 : RAM
Bit 4 : Data exchange impulse
Bit 5 : Read (0)/Write (1) data
Bit 6 : A0 of the graphic plane
Bit 7 : A1 of the graphic plane

E013 : Port of control. Upon initialization sending the 0x80 byte, i.e., all ports out.

With this interface, basic, we can code on each of the 4 graphic planes, a set of 256 characters, a total 4x256 = 1024 characters but not simultaneous.
To access the total of 4096 characters, I used bits 7 and 3 of the color coding byte of a character.
Bit 7 (CG32) represents A11, and bit 3 (D3 VRAM) represents A12.

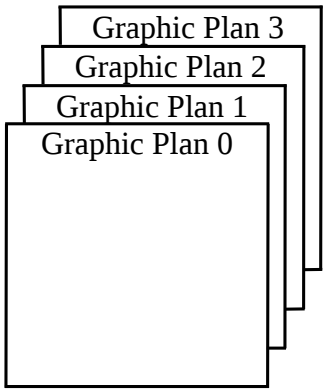
<table>
<tr><td>D7</td><td colspan="2">A11 address (CG32)</td></tr>
<tr><td>D6</td><td rowspan="3">Character Color</td><td>Green</td></tr>
<tr><td>D5</td><td>Red</td></tr>
<tr><td>D4</td><td>Blue</td></tr>
<tr><td>D3</td><td colspan="2">A12 address (D3 VRAM)</td></tr>
<tr><td>D2</td><td rowspan="3">Background color</td><td>Green</td></tr>
<tr><td>D1</td><td>Red</td></tr>
<tr><td>D0</td><td>Blue</td></tr>
</table>

Thus, one can encode 1024 characters per graphic plane.
Knowing that the screen represents 40x25 characters or 1000 characters, we can represent 4 images of 320x200 on 8 colors of characters and 8 colors of background by block of 8x8.

Programmable Characters Generator (PCG) for Sharp MZ-700
Version Extension BUS

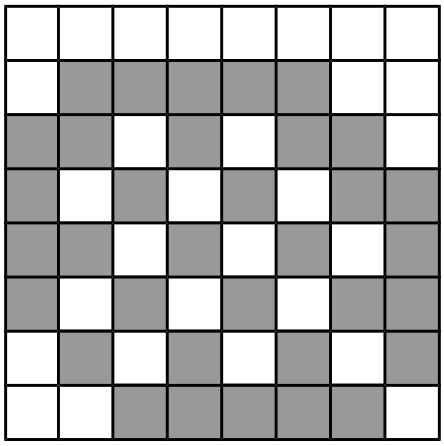
Summary :

Number of Graphic Plan	E012	
	Bit 6	Bit 7
0	0	0
1	1	0
2	0	1
3	1	1



Number character N (0 ≤ N ≤ 1023)	address A12 ... A3	A2	A1	A0	
Ligne 0	N	0	0	0	D0
Ligne 1	N	0	0	1	D1
Ligne 2	N	0	1	0	D2
Ligne 3	N	0	1	1	D3
Ligne 4	N	1	0	0	D4
Ligne 5	N	1	0	1	D5
Ligne 6	N	1	1	0	D6
Ligne 7	N	1	1	1	D7

Example : Asteroid



Programming Character assembler in the graphic plane 0, character N = 1 (0x008):

```
; Ligne 0
LD A, 0x00
LD (0xE010), A
LD A, 0x08
LD (0xE011), A
LD A, 0x10
LD (0xE012), A
LD A, 0x00
LD (0xE012), A
; Ligne 1
LD A, 0x7C
LD (0xE010), A
LD A, 0x09
LD (0xE011), A
LD A, 0x10
LD (0xE012), A
LD A, 0x00
LD (0xE012), A
```

```
; Ligne 2
LD A, 0xD6
LD (0xE010), A
LD A, 0x0A
LD (0xE011), A
LD A, 0x10
LD (0xE012), A
LD A, 0x00
LD (0xE012), A
; Ligne 3
LD A, 0xAB
LD (0xE010), A
LD A, 0x0B
LD (0xE011), A
LD A, 0x10
LD (0xE012), A
LD A, 0x00
LD (0xE012), A
```

```
; Ligne 4
LD A, 0xD5
LD (0xE010), A
LD A, 0x0C
LD (0xE011), A
LD A, 0x10
LD (0xE012), A
LD A, 0x00
LD (0xE012), A
; Ligne 5
LD A, 0xAB
LD (0xE010), A
LD A, 0x0D
LD (0xE011), A
LD A, 0x10
LD (0xE012), A
LD A, 0x00
LD (0xE012), A
```

```
; Ligne 6
LD A, 0x55
LD (0xE010), A
LD A, 0x0E
LD (0xE011), A
LD A, 0x10
LD (0xE012), A
LD A, 0x00
LD (0xE012), A
; Ligne 7
LD A, 0x3E
LD (0xE010), A
LD A, 0x0F
LD (0xE011), A
LD A, 0x10
LD (0xE012), A
LD A, 0x00
LD (0xE012), A
```