TSconf: Tiles

Development

Yes, Multimatograf close, you need to push!

So, tiles.

To build tiles, we need to do the following:

- prepare graphics *(place it on the page for tile graphics)*
- build a tile map *(with placement on the map page)*
- program ports *(specify which pages are used for graphics, for a map, and enable the display layers)*

> Tao says: Tiles are a combination of graphics and a map of its location on the screen.
>
> There are two layers of tiles that are visually located one above the other: T0 (below) & T1 (above).
>
> The map is stored in the memory page indicated by the TMPage port (#16af) as follows: the first line of the map of the T0 layer is located from #c000 to #c07f, the first line of the map of the T1 layer is from #c080 to #c0ff. The next row of tiles in both maps is located at #100 below.
>
> The tile descriptor consists of two bytes and indicates the number of an element of graphics with the size of 8x8 dots, allowing you to display it vertically and / or horizontally reflected, in one of 4 palettes.
>
> The position of the tile display window on the screen is set by four ports T0XOffs / T0YOffs (#40af-43af) for layer 0, and T1XOffs / T1YOffs for layer 1 (#44af-47af).

Preparing graphics:

To prepare the graphics, we need to consider that the tile is an element of 8x8 pixels, displayed in 16 colors with transparency, which is specified by the first (0th) color of the palette chosen for the tiles.

For use in tiles, the schedule should be placed in separate pages, the first of which should be a multiple of 8 (0, 8, 192, 240, etc.).

We have a graphics size of 256 * 256 pixels, 16 colors, located in the glasspat_page page with #c000 addresses.

Copy this graphic with the help of DMA into the page used to store Tile0_spr_page prepared for output graphics.

```
glasspat_page:
  equ #22
Tile0_spr_page:
  equ #50

  ld hl,glasspat_copy
  call set_ports
  ret
```

```
glasspat_copy:
  db #1a,0
  db #1b,0
  db #1c,glasspat_page
glasspat_copy_adr:
  db #1d,0
  db #1e,0
  db #1f,Tile0_spr_page
  db #26,256/4-1
  db #28,256-1
  db #27,DMA_RAM + DMA_DALGN
  db #ff
```

Graphics copied with alignment by placement in the memory of the receiver. The original image lies in a linear fashion.

Mapping:

Now you need to create a map of the location of tiles on the screen. To do this, in the page with a tile map, place information about the location of tiles in a page with graphics for them. To do this, we use the following tile

numbering scheme in the graphic: To fill the card with data, let's take a look at which bits are used in the tile descriptor:

```
TILE Reg.16 7 6 5 4 3 2 1 0
0 ROL TNUM[7:0]
1 ROH YF XF TPAL[5:4] TNUM[11:8]
```

TNUM is the number of the tile in the graphic, occupies 12 bits (all 8 low and 4 most high), which enables us to address 4096 tiles.

The YF bit is responsible for the vertical flip (flip) when the tile is displayed, the XF bit is responsible for the horizontal flip.

The TPAL bits are responsible for the choice of the palette and indicate its two lower digits, its higher two bits are indicated in the port of PalSel.

Accordingly, for one layer only 4 palettes from all of them arranged in series and multiples of 4: 0-3, 4-7, 8-11, 12-15 can be used.

So, let's assume that the image uses the 0th palette, and it will be located from the upper left corner of the tile layer T0.

Accordingly, the data is placed with #c000 layer T0 (and for the location in the layer T1 - from the address #c080).

Begin to fill in the map data located in the Tile_page page:

```
Tile_page:
  equ #c0

  ld hl,#c000
  ld de,#0000
  ld bc,#2020
```

```
  ld a,Tile_page
  call tile_filler
  ret

tile_filler:
  exx
  ld bc,PAGE3    ;turn on the tile map page
  out (c),a
  exx
  ld a,l
  ld (rfil3+1),a ; save the initial position of the address in the map
  ld a,#40
  sub b
  ld (rfil4+1),a ; calculate and save height
rfil1:
  push bc
rfil2:
  ld (hl),e      ; save the tile number
  inc l
  ld (hl),d      ; save number + attributes
  inc l
  inc de         ; next tile number
  djnz rfil2
  inc h
rfil3:
  ld l,0         ; go to a new line in the map
  ex de,hl
rfil4:
  ld bc,0
  add hl,bc      ; calculate the next line of tiles
  ex de,hl
  pop bc
  dec c
  jr nz,rfil1
  ret
```

Map is built.

To enable the display of tiles, you need to specify in the system where what is and what layers will be included.

We program ports:

T0GPage (#17af) - 1st page number of bitmap (graphics) for T0 layer

T1GPage (#18af) - 1st page number of bitmap for T1 layer

TMPage (#16af) - tile map number of both layers

TSConfig (#06af) - port controls for displaying tiles and sprites (1 - on, 0 - off):

```
bit 7 S_EN - display sprites
```

```
bit 6 T1_EN - display tile layer 1
bit 5 T0_EN - display tile layer 0
bit 3 T1Z_EN - enable display of tile with number 0 for layer 1
bit 2 T0Z_EN - enable display of tile with number 0 for layer 0
bits 4,1,0 are not used yet.
```

From all this magnificence, we need to include T0_EN and T0Z_EN in order for the first image tile to be displayed.

```
  ld hl,t0init
  call set_ports
  ret

t0init:
  db high TSCONFIG,TSU_T0ZEN+TSU_T0EN
  db high TMPAGE, Tile_page
  db high T0GPAGE,Tile0_spr_page
  db high PALSEL,0
  db #ff
```

Everything, tiles on the screen.

Questions and Answers

:? Why is there color noise on the screen besides the image?

! Noise is associated with the initial general initialization of memory in the system, and the Tile0_spr_page / Tile_page pages, in which the memory has a random state.

We take into account that the map was built only for our image, respectively, the other descriptors in the tile map have an unknown random state, which we see on the screen. It is necessary to clear it by specifying a tile suitable for this.

? It seems everything brought out, visually the size is that - and the image does not understand what

! The first byte of the tile is always even- 0,2,4, etc. Those. to display 8 tiles, we start to form a map from the address #c010.

Fiction: FAQ