Hi everyone !

I am struggling with DMA2D configuration (using HAL library) to display anti-aliased fonts. The fonts I have are A4 coded. Every character has its own array. For example, for the exclamation mark :

```
uint8 t ExclamationMark[57] = {
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x05, 0xF7, 0x00,
  0x05, 0xF7, 0x00,
  0 \times 05, 0 \times F7, 0 \times 00,
  0 \times 05, 0 \times F7, 0 \times 00,
  0x05, 0xF7, 0x00,
  0x03, 0xE5, 0x00,
  0x03, 0xE5, 0x00,
  0x02, 0xC3, 0x00,
  0x02, 0xC3, 0x00,
  0x02, 0xC3, 0x00,
  0x00, 0x00, 0x00,
  0 \times 05, 0 \times F7, 0 \times 00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x00
```

```
};
```

Since it is A4 formatted, every byte in this array allows to draw 2 pixels. Thus, the exclamation mark occupies 6x19 pixels.

Here is what I have done so far using the HAL library (simplified) :

```
HAL StatusTypeDef DrawEM(uint16 t x, uint16 t y) {
    uint8 t ExclamationMark[57] = {
      0x00, 0x00, 0x00,
      0x00, 0x00, 0x00,
      0x00, 0x00, 0x00,
      0x05, 0xF7, 0x00,
      0x03, 0xE5, 0x00,
      0x03, 0xE5, 0x00,
      0x02, 0xC3, 0x00,
      0x02, 0xC3, 0x00,
      0x02, 0xC3, 0x00,
      0x00, 0x00, 0x00,
      0 \times 05, 0 \times F7, 0 \times 00,
      0x00, 0x00, 0x00,
      0x00, 0x00, 0x00,
      0x00, 0x00, 0x00,
      0x00, 0x00, 0x00
    };
    DMA2D HandleTypeDef charDMA2D;
```

```
31/07/2018
                                      171512[PostContent].html
     uint32 t destination = ((uint32 t)backBuffer + (4*(y*LCD X SIZE +
 x)));
     charDMA2D.Instance = DMA2D;
     charDMA2D.Init.Mode = DMA2D M2M BLEND;
     charDMA2D.Init.ColorMode = DMA2D OUTPUT ARGB8888;
     charDMA2D.Init.OutputOffset = 800 - 6;
     charDMA2D.LayerCfq[0].AlphaMode = DMA2D NO MODIF ALPHA;
     charDMA2D.LayerCfg[0].InputColorMode = DMA2D INPUT ARGB8888;
     charDMA2D.LayerCfg[0].InputAlpha = 0;
     charDMA2D.LayerCfg[0].InputOffset = 0;
     charDMA2D.LayerCfg[1].AlphaMode = DMA2D COMBINE ALPHA;
     charDMA2D.LayerCfg[1].InputColorMode = DMA2D INPUT A4;
     charDMA2D.LayerCfg[1].InputAlpha = 0;
     charDMA2D.LayerCfg[1].InputOffset = 0;
     HAL DMA2D Init (&charDMA2D);
     HAL DMA2D ConfigLayer(&charDMA2D, 0);
     HAL DMA2D ConfigLayer(&charDMA2D, 1);
     HAL DMA2D BlendingStart(&charDMA2D, (uint32 t)EclamationMark,
 (uint32 t) 0xFF000000, destination, 6, 19);
     HAL DMA2D PollForTransfer(&charDMA2D, 1000);
```

}

Of course, it doesn't work. And I know that it doesn't work because there are too many parameters I don't quite understand. To set some parameters, I used adn read some notes and presentations made by ST for the DMA2D. Yet, when it comes to blending, I am just lost.

So, since every character I have is A4 formatted, it means that every 4-bit describes a pixel of the character by its Alpha value. Correct ? This is my FG layer.

My BG layer is supposed to be a fixed color (black for instance).

Thus, the blending between those two layers is supposed to display an anti-aliased character (in relation with the fixed color used). In one of those presentation made by ST, there was this picture :



So I don't know why, but I can't achieve this result (adapted to A4). Where am I supposed to set this "color register" ?

Is anyone can help me out and guide me through this darkness ? I have tried to study the STCube blending example but it doesn't seem to work with my STM32F469-DISCOVERY.

Thanks !

EDIT : Ok so I managed to make it work, barely work. Here are my new parameters for both layers :

```
charDMA2D.LayerCfg[0].AlphaMode = DMA2D_NO_MODIF_ALPHA;
charDMA2D.LayerCfg[0].InputColorMode = DMA2D_INPUT_ARGB8888;
charDMA2D.LayerCfg[0].InputAlpha = 0x00;
charDMA2D.LayerCfg[0].InputOffset = 0;
charDMA2D.LayerCfg[1].AlphaMode = DMA2D_NO_MODIF_ALPHA;
charDMA2D.LayerCfg[1].InputColorMode = DMA2D_INPUT_A4;
charDMA2D.LayerCfg[1].InputAlpha = 0x00; // Set the color register
charDMA2D.LayerCfg[1].InputOffset = 0;
```

Plus the fact that I didn't understand back then that it was necessary to have a kind of "background buffer" that contains pixel to blend with the FG layer.

Anyway, even if it barely works, I have the feeling that it doesn't work completely. Here is why :

I draw the "1" number on my display (100px height). Here is what I am supposed to get :







Sorry for the quality of the picture. As you can see (not easy I know), it is... strange when it comes to draw the border of the character. For example, at the beginning of it (left, beginning of the "1" curve), file:///C:/Work/STM/data/1-Faical/171512[PostContent].html

there is on my screen one vertical black line (1px), then one white (1px) and black gain. Even on the right of the character, even if we can't see there easily, there is :

- All black on the vertical path ;

- A vertical white line (1px);

- A kind of grey vertical line (1px);

Do you have a solution ? This result is from the blending operation, which is not possible to see for the user. But I don't understand why I have white pixel where I am supposed to have black ones (or black ones where I am supposed to have white ones).

EDIT 2 : Sooo, after some thinking I figure it out why my characters are not quite drawn as I expected. The issue seems to be the way the DMA2D peripheral fetches data. As I explained before, my characters are A4 formatted. For example to draw two pixels (side by side on the same line) in A4 format I need one byte. Let's say 0x0F : 0 is the alpha channel of the first pixel. F the alpha channel of the second pixel. At least that's what I thought. Actually, when the DMA2D fetches data in memory, it associates F to the first pixel and 0 to the second pixel. This way, there is an inversion. Every two vertical lines are inverted. The result if this inversion is the picture above. I tried manually on a character to "reverse" every 4-bit in a byte and my character was as I expected.

So my new question is : is there any way to change the way DMA2D fetch A4 formatted data ?

Here is what the Reference Manual indicates :

A4	A ₇ [3:0]A ₆ [3:0]	A ₅ [3:0]A ₄ [3:0]	A ₃ [3:0]A ₂ [3:0]	A ₁ [3:0]A ₀ [3:0]
----	--	--	--	--

What I want to achieve is :

Format	<u>a</u> + 3	(a) + 2	<i>(a)</i> + 1	(a) + 0
A4	A6[3:0]A7[3:0]	A4[3:0]A5[3:0]	A2[3:0]A3[3:0]	A0[3:0]A1[3:0]

Is this possible ? Or should I wrote some kind of script of my own to "rewrite" every file character ?

Thanks again !