

零死角玩转STM32



LTDC—液晶显示

淘宝：firestm32.taobao.com

论坛：www.chuxue123.com



扫描进入淘宝店铺

主讲内容



01

显示器简介

02

液晶控制原理

03

LTDC液晶控制器

04

DMA2D图形加速器

05

LTDC及DMA2D结构体

06

LTDC/DMA2D—液晶显示实验

参考资料:《零死角玩转STM32》

“LTDC/DMA2D—液晶显示” 章节

LTDC—液晶显示



LTDC初始化结构体

控制LTDC涉及到非常多的寄存器，利用LTDC初始化结构体可以减轻开发和维护的工作量。这个结构体大部分成员都是用于定义LTDC的时序参数的，包括信号有效电平及各种时间参数的宽度。

```
1 /**
2  * @brief LTDC Init structure definition
3  */
4 typedef struct
5 {
6     uint32_t LTDC_HSPolarity;           /*配置行同步信号 HSYNC 的极性 */
7     uint32_t LTDC_VSPolarity;           /*配置垂直同步信号 VSYNC 的极性 */
8     uint32_t LTDC_DEPolarity;           /*配置数据使能信号 DE 的极性*/
9     uint32_t LTDC_PCPolarity;           /*配置像素时钟信号 CLK 的极性 */
10    uint32_t LTDC_HorizontalSync;        /*配置行同步信号 HSYNC 的宽度 (HSW-1)
11 */
12    uint32_t LTDC_VerticalSync;          /*配置垂直同步信号 VSYNC 的宽度 (VSW-1)
13 */
14    uint32_t LTDC_AccumulatedHBP;        /*配置 (HSW+HBP-1) 的值*/
15    uint32_t LTDC_AccumulatedVBP;        /*配置 (VSW+VBP-1) 的值*/
16    uint32_t LTDC_AccumulatedActiveW;    /*配置 (HSW+HBP+有效宽度-1) 的值*/
17    uint32_t LTDC_AccumulatedActiveH;    /*配置 (VSW+VBP+有效高度-1) 的值*/
18    uint32_t LTDC_TotalWidth;            /*配置 (HSW+HBP+有效宽度+HFP-1) 的值
19 */
20    uint32_t LTDC_TotalHeigh;            /*配置 (VSW+VBP+有效高度+VFP-1) 的值
21 */
22    uint32_t LTDC_BackgroundRedValue;    /*配置背景的红色值*/
23    uint32_t LTDC_BackgroundGreenValue;  /*配置背景的绿色值*/
24    uint32_t LTDC_BackgroundBlueValue;   /*配置背景的蓝色值*/
25 } LTDC_InitTypeDef;
```

LTDC初始化结构体

- **LTDC_HSPolarity**

本成员用于设置行同步信号HSYNC的极性，即HSYNC有效时的电平，该成员的值可设置为高电平(LTDC_HSPolarity_AH)或低电平(LTDC_HSPolarity_AL)。

- **LTDC_VSPolarity**

本成员用于设置垂直同步信号VSYNC的极性，可设置为高电平(LTDC_VSPolarity_AH)或低电平(LTDC_VSPolarity_AL)。

- **LTDC_DEPolarity**

本成员用于设置数据使能信号DE的极性，可设置为高电平(LTDC_DEPolarity_AH)或低电平(LTDC_DEPolarity_AL)。

LTDC初始化结构体

- **LTDC_PCPolarity**

本成员用于设置像素时钟信号CLK的极性，可设置为上升沿(LTDC_DEPolarity_AH)或下降沿(LTDC_DEPolarity_AL)，表示RGB数据信号在CLK的哪个时刻被采集。

- **LTDC_HorizontalSync**

本成员设置行同步信号HSYNC的宽度HSW，它以像素时钟CLK的周期为单位，实际写入该参数时应写入(HSW-1)，参数范围为0x000- 0xFFFF。

- **LTDC_VerticalSync**

本成员设置垂直同步信号VSYNC的宽度VSW，它以“行”为位，实际写入该参数时应写入(VSW-1)，参数范围为0x000- 0x7FF。

LTDC初始化结构体

- **LTDC_AccumulatedHBP**

本成员用于配置“水平同步像素HSW”加“水平后沿像素HBP”的累加值，实际写入该参数时应写入 $(HSW+HBP-1)$ ，参数范围为0x000- 0xFFFF。

- **LTDC_AccumulatedVBP**

本成员用于配置“垂直同步行VSW”加“垂直后沿行VBP”的累加值，实际写入该参数时应写入 $(VSW+VBP-1)$ ，参数范围为0x000- 0x7FF。

- **LTDC_AccumulatedActiveW**

本成员用于配置“水平同步像素HSW”加“水平后沿像素HBP”加“有效像素”的累加值，实际写入该参数时应写入 $(HSW+HBP+有效宽度-1)$ ，参数范围为0x000- 0xFFFF。

LTDC—液晶显示



LTDC初始化结构体

- **LTDC_AccumulatedActiveH**

本成员用于配置“垂直同步行VSW”加“垂直后沿行VBP”加“有效行”的累加值，实际写入该参数时应写入 $(VSW+VBP+有效高度-1)$ ，参数范围为0x000-0x7FF。

- **LTDC_TotalWidth**

本成员用于配置“水平同步像素HSW”加“水平后沿像素HBP”加“有效像素”加“水平前沿像素HFP”的累加值，即总宽度，实际写入该参数时应写入 $(HSW+HBP+有效宽度+HFP-1)$ ，参数范围为0x000- 0xFFF。

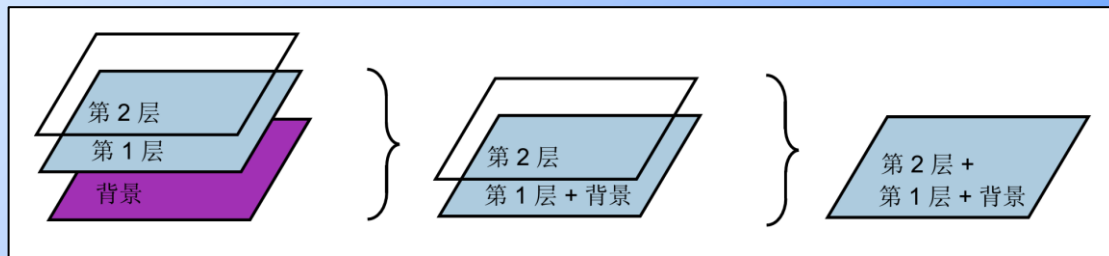
- **LTDC_TotalHeigh**

本成员用于配置“垂直同步行VSW”加“垂直后沿行VBP”加“有效行”加“垂直前沿行VFP”的累加值，即总高度，实际写入该参数时应写入 $(HSW+HBP+有效高度+VFP-1)$ ，参数范围为0x000- 0x7FF

LTDC初始化结构体

- **LTDC_BackgroundRedValue/ GreenValue/ BlueValue**

这三个结构体成员用于配置背景的颜色值，这里说的背景层与前面提到的“前景层/背景层”概念有点区别，它们对应下图中的“第2层/第1层”，而在这两层之外，还有一个最终的背景层，当第1第2层都透明时，这个背景层就会被显示，而这个背景层是一个纯色的矩形，它的颜色值就是由这三个结构体成员配置的，各成员的范围为0x00- 0xFF。



对这些LTDC初始化结构体成员赋值后，调用库函数**LTDC_Init**可把这些参数写入到LTDC的各个配置寄存器，LTDC外设根据这些配置控制时序。

LTDC—液晶显示



LTDC 层级初始化结构体

LTDC初始化结构体只是配置好了与液晶屏通讯的基本时序，还有像素格式、显存地址等诸多参数需要使用LTDC层级初始化结构体完成。

```
1 /**
2  * @brief LTDC Layer structure definition
3  */
4 typedef struct
5 {
6     uint32_t LTDC_HorizontalStart;           /*配置窗口的行起始位置 */
7     uint32_t LTDC_HorizontalStop;            /*配置窗口的行结束位置 */
8     uint32_t LTDC_VerticalStart;             /*配置窗口的垂直起始位置 */
9     uint32_t LTDC_VerticalStop;              /*配置窗口的垂直束位置 */
10    uint32_t LTDC_PixelFormat;                /*配置当前层的像素格式*/
11    uint32_t LTDC_ConstantAlpha;              /*配置当前层的透明度Alpha 常量
值*/
12    uint32_t LTDC_DefaultColorBlue;           /*配置当前层的默认蓝色值 */
13    uint32_t LTDC_DefaultColorGreen;          /*配置当前层的默认绿色值 */
14    uint32_t LTDC_DefaultColorRed;            /*配置当前层的默认红色值 */
15    uint32_t LTDC_DefaultColorAlpha;          /*配置当前层的默认透明值 */
16    uint32_t LTDC_BlendingFactor_1;           /*配置混合因子 BlendingFactor1
*/
17    uint32_t LTDC_BlendingFactor_2;           /*配置混合因子
BlendingFactor2 */
18    uint32_t LTDC_CFBStartAddress;            /*配置当前层的显存起始位置*/
19    uint32_t LTDC_CFBLineLength;             /*配置当前层的行数据长度 */
20    uint32_t LTDC_CFBPitch;                  /*配置从某行的起始到下一行像素起始处的增
量*/
21    uint32_t LTDC_CFBLineNumber;             /* 配置当前层的行数*/
22 } LTDC_Layer_InitTypeDef;
```

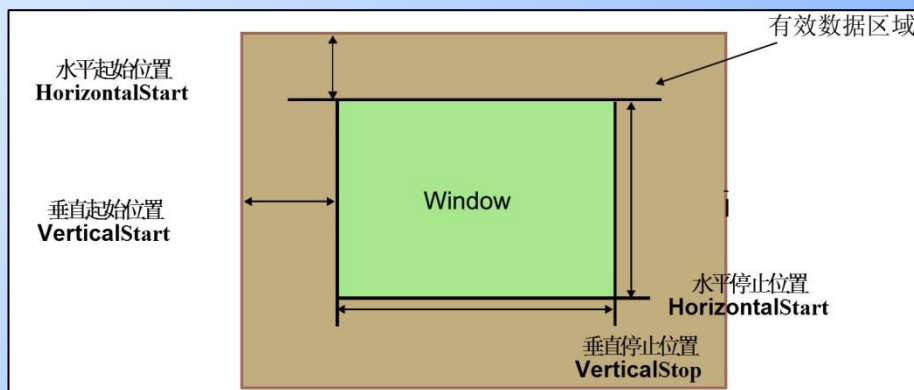
LTDC—液晶显示



LTDC 层级初始化结构体

- **LTDC_HorizontalStart /HorizontalStop/ VerticalStart/ VerticalStop**

这些成员用于确定该层显示窗口的边界，分别表示行起始、行结束、垂直起始及垂直结束的位置，注意这些参数包含同步HSW/VSW、后沿大小HBP/VBP和有效数据区域的内部时序发生器的配置，表中的是各个窗口配置成员应写入的数值。



LTDC层级窗口配置成员	等效于LTDC时序参数配置成员的值	实际值
LTDC_HorizontalStart	(LTDC_AccumulatedHBP+1)	HBP + HSW
LTDC_HorizontalStop	LTDC_AccumulatedActiveW	HSW+HBP+LCD_PIXEL_WIDTH-1
LTDC_VerticalStart	(LTDC_AccumulatedVBP+1)	VBP + VSW
LTDC_VerticalStop	LTDC_AccumulatedActiveH	VSW+VBP+LCD_PIXEL_HEIGHT-1

LTDC 层级初始化结构体

- **LTDC_PixelFormat**

本成员用于设置该层数据的像素格式，可以设置为

LTDC_Pixelformat_ARGB8888/ RGB888/ RGB565/ ARGB1555/ ARGB4444/
L8/ AL44/ AL88格式。

- **LTDC_ConstantAlpha**

本成员用于设置该层恒定的透明度常量Alpha，称为恒定Alpha，参数范围为0x00-0xFF，在图层混合时，可根据后面的BlendingFactor成员的配置，选择是只使用这个恒定Alpha进行混合运算还是把像素本身的Alpha值也加入到运算中。

LTDC 层级初始化结构体

- **LTDC_DefaultColorBlue/Green/Red/Alpha**

这些成员用于配置该层的默认颜色值，分别为蓝、绿、红及透明分量，该颜色在定义的层窗口外或在层禁止时使用。

- **LTDC_BlendingFactor_1/2**

本成员用于设置混合系数 BF1 和 BF2。每一层实际显示的颜色都需要使用透明度参与运算，计算出不包含透明度的直接RGB颜色值，然后才传输给液晶屏(因为液晶屏本身没有透明的概念)。混合的计算公式为：

$$BC = BF1 \times C + BF2 \times Cs$$

LTDC—液晶显示



LTDC 层级初始化结构体

- **LTDC_BlendingFactor_1/2**

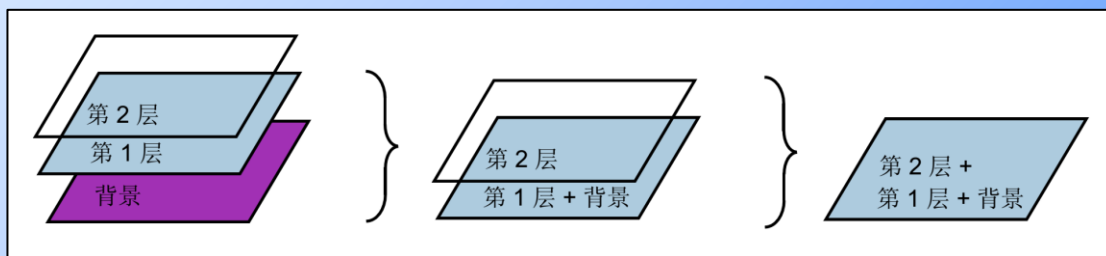
混合的计算公式: $BC = BF1 \times C + BF2 \times Cs$

参数	说明	CA	PAxCA
BC	混合后的颜色(混合结果)	-	-
C	当前层颜色	-	-
Cs	底层混合后的颜色	-	-
BF1	混合系数1	等于(恒定Alpha值)	等于(恒定Alpha x 像素Alpha值)
BF2	混合系数2	等于(1-恒定Alpha)	等于(1-恒定Alpha x 像素Alpha值)

LTDC 层级初始化结构体

- **LTDC_BlendingFactor_1/2**

本结构体成员可以设置BF1/BF2参数使用CA配置 (LTDC_BlendingFactor1/2_CA) 还是PAXCA配置 (LTDC_BlendingFactor1/2_PAXCA)。配置成CA表示混合系数中只包含恒定的Alpha值，即像素本身的Alpha不会影响混合效果，若配置成PAXCA，则混合系数中包含有像素本身的Alpha值，即把像素本身的Alpha加入到混合运算中。其中的恒定Alpha值即前面“LTDC_ConstantAlpha”结构体配置参数的透明度百分比：(配置的Alpha值/0xFF)。



数据源混合时，由下至上，如果使用了2层，则先将第1层与LTDC背景混合，随后再使用该混合颜色与第2层混合得到最终结果。例如，当只使用第1层数据源时，且BF1及BF2都配置为使用恒定Alpha，该Alpha值在LTDC_ConstantAlpha结构体成员值中被配置为240(0xF0)。因此，恒定Alpha值为 $240/255=0.94$ 。若当前层颜色 $C=128$ ，背景色 $C_s=48$ ，那么第1层与背景色的混合结果为：
 $BC = \text{恒定Alpha} \times C + (1 - \text{恒定Alpha}) \times C_s = 0.94 \times C_s + (1 - 0.94) \times 48 = 123$

LTDC 层级初始化结构体

- **LTDC_CFBStartAdress**

本成员用于设置该层的显存首地址，该层的像素数据保存在从这个地址开始的存储空间内。

- **LTDC_CFBLineLength**

本成员用于设置当前层的行数据长度，即每行的有效像素点个数 \times 每个像素的字节数，实际配置该参数时应写入值(行有效像素个数 \times 每个像素的字节数+3)，每个像素的字节数跟像素格式有关，如RGB565为2字节，RGB888为3字节，ARGB8888为4字节。

LTDC 层级初始化结构体

- **LTDC_CFBPitch**

本成员用于设置从某行的有效像素起始位置到下一行起始位置处的数据增量，无特殊情况的话，它一般就直接等于行的有效像素个数x每个像素的字节数。

- **LTDC_CFBLineNumber**

本成员用于设置当前层的显示行数。

配置完LTDC_Layer_InitTypeDef层级初始化结构体后，调用库函数**LTDC_LayerInit**可把这些配置写入到LTDC的层级控制寄存器中，完成初始化。初始化完成后LTDC会不断把显存空间的数据传输到液晶屏进行显示，我们可以直接修改或使用DMA2D修改显存中的数据，从而改变显示的内容。

LTDC—液晶显示



DMA2D结构体

在实际显示时，我们常常采用DMA2D描绘直线和矩形，这个时候会用到DMA2D结构体。

```
1  /**
2   * @brief  DMA2D Init structure definition
3   */
4  typedef struct
5  {
6      uint32_t  DMA2D_Mode;                /*配置 DMA2D 的传输模式*/
7      uint32_t  DMA2D_CMode;              /*配置 DMA2D 的颜色模式 */
8      uint32_t  DMA2D_OutputBlue;         /*配置输出图像的蓝色分量*/
9      uint32_t  DMA2D_OutputGreen;        /*配置输出图像的绿色分量*/
10     uint32_t  DMA2D_OutputRed;           /*配置输出图像的红色分量*/
11     uint32_t  DMA2D_OutputAlpha;         /*配置输出图像的透明度分量
*/
12     uint32_t  DMA2D_OutputMemoryAdd;     /*配置显存地址*/
13     uint32_t  DMA2D_OutputOffset;        /*配置输出地址偏移*/
14     uint32_t  DMA2D_NumberOfLine;        /*配置要传输多少行 */
15     uint32_t  DMA2D_PixelPerLine;        /*配置每行有多少个像素 */
16 } DMA2D_InitTypeDef;
```


DMA2D结构体

- **DMA2D_Mode**

本成员用于配置DMA2D的工作模式，它可以被设置为下表中的值：

宏	说明
DMA2D_M2M	从存储器到存储器（仅限FG获取数据源）
DMA2D_M2M_PFC	存储器到存储器并执行 PFC（仅限 FG PFC 激活时的 FG 获取）
DMA2D_M2M_BLEND	存储器到存储器并执行混合（执行 PFC 和混合时的 FG 和 BG 获取）
DMA2D_R2M	寄存器到存储器（无 FG 和 BG，仅输出阶段激活）

这几种工作模式主要区分数据的来源、是否使能PFC以及是否使能混合器。使用DMA2D时，可把数据从某个位置搬运到显存，该位置可以是DMA2D本身的寄存器，也可以是设置好的DMA2D前景地址、背景地址(即从存储器到存储器)。若使能了PFC，则存储器中的数据源会经过转换再传输到显存。若使能了混合器，DMA2D会把两个数据源中的数据混合后再输出到显存。若使用存储器到存储器模式，需要调用库函数DMA2D_FGConfig，使用初始化结构体DMA2D_FG_InitTypeDef配置数据源的格式、地址等参数。(背景层使用函数DMA2D_BGConfig和结构体DMA2D_BG_InitTypeDef)。

DMA2D结构体

- **DMA2D_CMode**

本成员用于配置DMA2D的输出PFC颜色格式，即它将要传输给显存的格式。

- **DMA2D_OutputBlue/ Green/ Red/ Alpha**

这几个成员用于配置DMA2D的寄存器颜色值，若DMA2D工作在“寄存器到存储器”(DMA2D_R2M)模式时，这个颜色值作为数据源，被DMA2D复制到显存空间，即目标空间都会被填入这一种色彩。

- **DMA2D_OutputMemoryAdd**

本成员用于配置DMA2D的输出FIFO的地址，DMA2D的数据会被搬运到该空间，一般把它设置为本次传输显示位置的起始地址。

DMA2D结构体

- DMA2D_OutputOffset**

本成员用于配置行偏移(以像素为单位)，行偏移会被添加到各行的结尾，用于确定下一行的起始地址。如下表中的黄色格子表示行偏移，绿色格子表示要显示的数据。左表中显示的是一条垂直的线，且线的宽度为1像素，所以行偏移的值 $=7-1=6$ ，即“行偏移的值=行宽度-线的宽度”，右表中的线宽度为2像素，行偏移的值 $=7-2=5$ 。

0	1	2	3	4	5	6	7
		■	■	■	■	■	■
■	■	■					
		■					
		■					
		■					

0	1	2	3	4	5	6	7
		■	■	■	■	■	■
■	■	■	■				
		■	■				
		■	■				
		■	■				

- **DMA2D_NumberOfLine**

- DMA2D PixelPerLine**

0	1	2	3	4	5	6	7

0	1	2	3	4	5	6	7

DMA2D结构体

配置完这些结构体成员，调用库函数**DMA2D_Init**即可把这些参数写入到DMA2D的控制寄存器中，然后再调用DMA2D_StartTransfer函数开启数据传输及转换。

零死角玩转STM32



THANKS

论坛：www.chuxue123.com

淘宝：firestm32.taobao.com



扫描进入淘宝店铺