Fancy OSD Engine Design

目录

[1. Overview 2](#_Toc508286096)

[1.1 OSD Definition 2](#_Toc508286097)

[1.2 Hardware memory layout 3](#_Toc508286098)

[1.3 Entity relationship 4](#_Toc508286099)

[1.4 Features 4](#_Toc508286100)

[1.5 Design consideration 4](#_Toc508286101)

[1.6 Data path 5](#_Toc508286102)

[2. Entity Attribute Definition 5](#_Toc508286103)

[Engine(帧) 5](#_Toc508286104)

[Palette(调色板) 5](#_Toc508286105)

[Window(窗口) 6](#_Toc508286106)

[Block(区块) 6](#_Toc508286107)

[Ingredient(素材 – 抽象类) 6](#_Toc508286108)

[Bitmap(位图 – 继承自Ingredient) 6](#_Toc508286109)

[Glyph(字体 -继承自Ingredient) 7](#_Toc508286110)

[Plot(绘图 – 抽象类 -继承自Ingredient) 7](#_Toc508286111)

[Pixel(点 -继承自Plot) 7](#_Toc508286112)

[Line(线 -继承自Plot) 8](#_Toc508286113)

[Rectangle(矩形 - 继承自Plot) 8](#_Toc508286114)

[Circle(矩形 - 继承自Plot) 8](#_Toc508286115)

[Arc(弧 - 继承自Plot) 9](#_Toc508286116)

[Oval(椭圆 - 继承自Plot) 9](#_Toc508286117)

[Modifier(修改器) 9](#_Toc508286118)

[Move(移动 – 继承自Modifier) 10](#_Toc508286119)

[Flip(图片更换 – 继承自Modifier) 10](#_Toc508286120)

[Fade(淡入淡出 – 继承自Modifier) 10](#_Toc508286121)

[3. Bitmap data compression 11](#_Toc508286122)

[Multi Palette 11](#_Toc508286123)

[Color index compress 11](#_Toc508286124)

[Fixed Palette 11](#_Toc508286125)

[4. Software Design 12](#_Toc508286126)

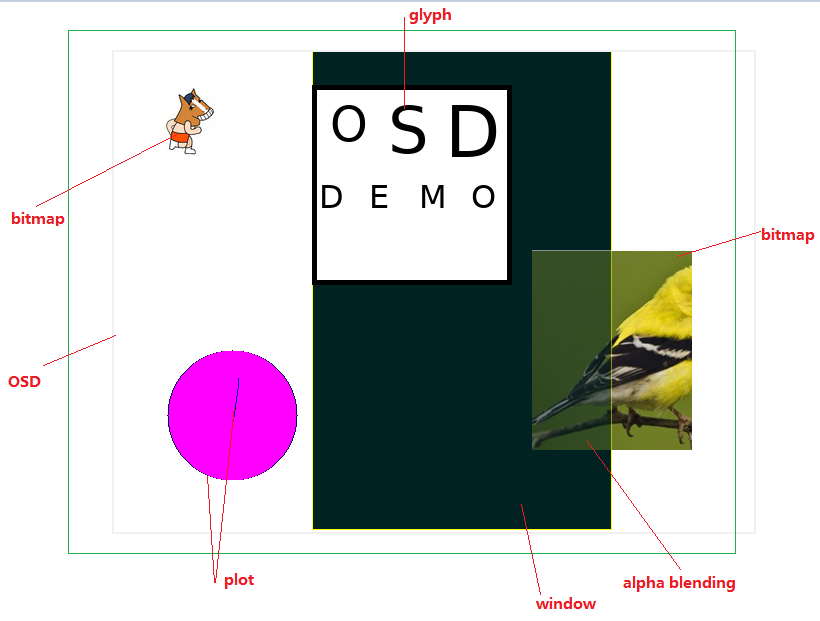
[Software Archtecture 12](#_Toc508286127)

[OML格式 13](#_Toc508286128)

[寄存器设计 13](#_Toc508286129)

# Overview

## OSD Definition



OSD: 屏幕上面的显示

Line: 每次刷新的行数据

Window: 窗口系统,一个独立的绘图范围,多个窗口可以叠加,Blending

Block: 窗口内部的一个显示单元, Block= Ingredient + position(x, y)

Ingredient:图像单元, ,可能是字体位图,一般位图或Plot绘图单元

Bitmap: 一副或者多副位图, 默认显示第一张

Glyph: 某个字对应的位图

Plot: 绘图单元: 动态计算产生的绘图单元, 不存储位图,直接画图到Line Buffer

Rectangle: Plot子类型, 左上角坐标 + 右下角坐标

Line: Plot子类型,起始点 + 终止点

Circle: Plot子类型, 中心点 +半径

Modifier: 修改器, 可以作用与Window或者Block结合, 产生动画效果

Animation: 动画, 在每一帧对Bitmap当前是图片Index进行修改,产生动画效果,不需要软件参与

Palette: 所有的颜色都存在LUT中, 有多个Palette, 每个Window或者Block都对应一个Palette

## Hardware memory layout

* All data structure generate by software tool
* HW need read it back and draw line using the data



## Entity relationship



## Features

* 多窗口管理
* 字体管理
* Bitmap绘制
* 绘图Plot单元引入, 通过动态作图避免存储位图数据,节省内存
* 引入Modifier, 可以由硬件在每一个Frame显示之前修改window或者ingredient的某些属性, 产生动画效果

## Design consideration

* + 1. No Frame Buffer
    2. 尽量少的内存占用
    3. 硬件容易实现,效率高
    4. 主要的软件工作尽量在设计OSD的时候在Tool里面完成,Firmware执行时软件尽量少参与
    5. 软件Model与硬件同步, 容易验证硬件功能
    6. 考虑以后的存在Frame Buffer的设计, 软件工具和数据保持不变

## Data path



# Entity Attribute Definition

## Engine(帧)

全局OSD对象, 负责绘制OSD

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| width, height | UINT16 | OSD大小 |
| Palettes | Array(Palette) | 所有调色板的集合 |
| Modifies | Array(Modifier) | 所有修改器的集合 |
| Ingredients | Array(Ingredient) | 所有素材集合 |
| Windows | Array(Window) | 所有窗口集合 |
| draw\_frame() | Operation | 画一帧OSD |

## Palette(调色板)

OSD的所有颜色都存放在LookupTable, 使用的颜色需要使用调色板来获取真正的RGB值

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| bit\_width | 4 BIT | 调色板索引值位数 |
| count | UINT16 | 项目数 |
| lut | Array(A, R,G,B) | 查找表内容 |
| get\_color(index) | Operation | 根据颜色索引值获取颜色 |

## Window(窗口)

表示一个窗口, 内部成员的坐标都是相对于窗口的坐标

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| x, y | UINT16 | 所在位置 |
| width, height | UINT16 | 大小 |
| Alpha | UIN8 | Alpha blending取值 |
| z-order | 4BITS | 窗口所在的前后关系 |
| Blocks | Array(Block) | 所有Block对象数组 |
| draw\_line(y) | Operation | 画一行OSD |

## Block(区块)

表示窗口内部的一个显示单元块

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| ingredient | Ingredient | 显示素材 |
| x, y | UINT16 | 在窗口内部的显示位置 |
| z-order | 4 BITS | 窗口内部的前后关系 |
| palette | Palette | Ingredient的默认调色板 |
| color | Palette | Ingredient的默认颜色 |
| angle | 4 BITS | 旋转的角度, 默认为0 |
| flip | 4 BITS | 翻转的角度, 默认为0 |
| fade | 4 BITS | 当前fade in/out的程度 |
| fill\_line(window\_line\_buf, y) | Operation | 填充Window的Line buffer |

## Ingredient(素材 – 抽象类)

表示一个绘图素材,可能是Bitmap, Glyph或者Plot

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| width, height | UINT16 | 大小 |
| fill\_line(line\_buf, y, offset\_x) | Operation | 将需要绘制的点写入到buffer中 |

## Bitmap(位图 – 继承自Ingredient)

一般图片对象, 可以包含一张或者多张图片,

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *width, height* | *UINT16* | *继承自Ingredient* |
| *fill\_line(line\_buf, y, offset\_x)* | *Operation* | *继承自Ingredient* |
| palette | Palette | 位图使用的Palette |
| count | UINT8 | 位图数量 |
| index | 4 BITS | 当前显示的位图index |
| data | Array(byte) | 位图数据  len = count \* width \* height \* bits\_per\_pixel/8 |

## Glyph(字体 -继承自Ingredient)

一般字体对象, 可以包含一个字符的code, style和大小信息

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *width, height* | *UINT16* | *继承自Ingredient* |
| *fill\_line(line\_buf, y, offset\_x)* | *Operation* | *继承自Ingredient* |
| code | UINT16 | 字符的Unicode编码 |
| style | 2 BITS | Normal/Italic//Bold |
| left, top | 4 BITS | BITMAP的开始位置 |
| data | Array(byte) | 字符的位图数据 |

## Plot(绘图 – 抽象类 -继承自Ingredient)

一般字体对象, 可以包含一个字符的code, style和大小信息

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *width, height* | *UINT16* | *继承自Ingredient* |
| *fill\_line(line\_buf, y, offset\_x)* | *Operation* | *继承自Ingredient* |
| palette | Palette | 点的调色板, 如果没有定义,则取自当前Block |
|  |  |  |

## Pixel(点 -继承自Plot)

一般字体对象, 可以包含一个字符的code, style和大小信息

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *width, height* | *UINT16* | *继承自Plot* |
| *fill\_line(line\_buf, y, offset\_x)* | *Operation* | *继承自Plot* |
| *palette* | *Palette* | *继承自Plot* |
| color | UINT8 | 点的颜色 |
| x, y | UINT16 | 点的位置 |

## Line(线 -继承自Plot)

一般字体对象, 可以包含一个字符的code, style和大小信息

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *width, height* | *UINT16* | *继承自Plot* |
| *fill\_line(line\_buf, y, offset\_x)* | *Operation* | *继承自Plot* |
| *palette* | *Palette* | *继承自Plot* |
| color | UINT8 | 线段颜色 |
| x1, y1,x2,y2 | UINT16 | 线的起始和终止 |
| weight | 4 BITS | 线的宽度 |
| style | 4 BITS | 线的类型  SOLID  DASH  DOT  DASHDOT  DASHDOTDOT |

## Rectangle(矩形 - 继承自Plot)

一般字体对象, 可以包含一个字符的code, style和大小信息

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *width, height* | *UINT16* | *继承自Plot* |
| *fill\_line(line\_buf, y, offset\_x)* | *Operation* | *继承自Plot* |
| *palette* | *Palette* | *继承自Plot* |
| color | UINT8 | 矩形边框颜色 |
| x1,y1,x2,y2 | UINT16 | 矩形的起始和终止点 |
| fill\_color1, fill\_color2 | UINT16 | 填充颜色, 两个颜色实现渐变色 |
| fill\_pattern | 4 BITS | 填充类型 |
| border\_style | 4 BITS | Same as Line.style |
| border\_weight | 4 BIT | Same as Line.weight |
| border\_pattern | 4 BITS |  |
| border\_color | UINT16 |  |
|  |  |  |

## Circle(矩形 - 继承自Plot)

一般字体对象, 可以包含一个字符的code, style和大小信息

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *width, height* | *UINT16* | *继承自Plot* |
| *fill\_line(line\_buf, y, offset\_x)* | *Operation* | *继承自Plot* |
| *Palette* | *Palette* | *继承自Plot* |
| color | UINT8 | 边框颜色 |
| center,radius | UINT16 | 中心点和半径 |
| fill\_color | UINT16 | 填充颜色, 两个颜色实现渐变色 |
| fill\_pattern | 4 BITS | 填充类型 |
| border\_style | 4 BITS | Same as Line.style |
| border\_weight | 4 BIT | Same as Line.weight |
| border\_pattern | 4 BITS |  |
| border\_color | UINT16 |  |
|  |  |  |

## Arc(弧 - 继承自Plot)

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *width, height* | *UINT16* | *继承自Plot* |
| *fill\_line(line\_buf, y, offset\_x)* | *Operation* | *继承自Plot* |
| *Palette* | *Palette* | *继承自Plot* |
| color | UINT8 | 颜色 |
| …. |  |  |

## Oval(椭圆 - 继承自Plot)

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *width, height* | *UINT16* | *继承自Plot* |
| *fill\_line(line\_buf, y, offset\_x)* | *Operation* | *继承自Plot* |
| *Palette* | *Palette* | *继承自Plot* |
| color | UINT8 | 颜色 |
| …. |  |  |

## Modifier(修改器)

修改器可以作用于window或者Block, 用来修改window或者block的属性来达到动画的目的

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| windows | Array(Window) | 修改器作用的窗口列表 |
| blocks | Array(Block) | 修改器作用的区块列表 |
| frame | UINT8 | 几个Frame调用一次Modifier |
| execute() | Operation | 执行修改器 |

## Move(移动 – 继承自Modifier)

移动窗口或者Block

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *windows* | *Array(Window)* | *继承自Modifier* |
| *blocks* | *Array(Block)* | *继承自Modifier* |
| *frame* | *UINT8* | *继承自Modifier* |
| *execute()* | *Operation* | *继承自Modifier* |
| x\_step, y\_step | UINT8 | 移动的步长 |
| x\_end, y\_end | UINT16 | 移动的终点 |

## Flip(图片更换 – 继承自Modifier)

只能作用于Bitmap对象,用于修改当前显示的图片

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *Windows* | *Array(Window)* | *继承自Modifier* |
| *Blocks* | *Array(Block)* | *继承自Modifier* |
| *Frame* | *UINT8* | *继承自Modifier* |
| *execute()* | *Operation* | *继承自Modifier* |
| index | 4 BITS | 当前显示的图片Index |

## Fade(淡入淡出 – 继承自Modifier)

作用于Window或者Block, 实现淡入淡出效果

|  |  |  |
| --- | --- | --- |
| 属性/操作 | 类型 | 说明 |
| *Windows* | *Array(Window)* | *继承自Modifier* |
| *Blocks* | *Array(Block)* | *继承自Modifier* |
| *Frame* | *UINT8* | *继承自Modifier* |
| *execute()* | *Operation* | *继承自Modifier* |
| duration | 4 BITS | 淡入淡出效果的步骤次数 |
| step | 4 BITS | 当前所在的步骤 |

# Bitmap data compression

## Multi Palette

采用多个Palette的方式, 可以减少每个color所需要存储的颜色位数

比如图片1(100 x 100)有16种颜色, 图片2(100 x 100)有200种颜色, 此时若采用同一个Palette, 都需要用8BIT来存放数据一个Pixel 值, 那么需要使用的BYTE数目为100 \* 100 + 100 \* 100 = 200000 bytes.

如果图片1采用一个4BIT的Palette, 图片2采用8bit的Palette, 则需要的Byte数据为(4 \* 100 \* 100)/8 + 100 \* 100 = 150000 bytes

## Color index compress

可以考虑对较大的BITMAP进行无损压缩,可以采用如下压缩方式,在工具中计算出最优的压缩方式,然后保存到图片数据中. 数据头需要保存每行的BYTE数, 用于在绘图的时候快速定位到某一行所在的BYTE. 在BITMAP数据转换查找LUT之前,先进行解压缩.

1. Huffman Code
2. RLE算法

需要考虑硬件解压缩的成本

## Fixed Palette

可以将常用的Palette用运算电路表示,避免存放所有RGB值, 比如灰度, 还有常用的颜色表

# Software Design

## Software Archtecture



|  |  |
| --- | --- |
| OML | OSD Markup Language  描述OSD数据的文本格式，采用 YAML或者XML格式 |
| GUI Tool | 图形编辑工具，用于生成OSD描述文件(OML) |
| Text Editor | 可以采用普通文本编辑工具来手动编辑OML文件 |
| OML2BIN | 命令行工具，将OSD YAML文件转换为寄存器描述文件+存放在Flash中的OSD Data文件 |
| Register File | OSD Engine寄存器配置，主要是Window的配置信息 |
| OSD Data File | OSD Data所有资源的数据信息,将OML里面描述的OSD转换为二进制格式，供Firmware和HW读取，可能存放在Flash或者RAM中 |
| OSD Simulation | OSD模拟器，运行在Windows上面，可以根据OSD数据模拟最终的运行效果，可以用于RTL的验证和OSD的调试 |

## OSD YAML格式

|  |
| --- |
|  |

## 寄存器设计

Ingredient Type

Palette Registers

X, Y

Pixel\_format

Ingredient Registers

Pixel\_bits

Palette0

Weight | Height

Rectangle1

entry\_count

Palette1

Border\_color

Bitmap1

Luts\_addr

…

Line1

Ingredient Type

Data\_size

Bitmap2

Width

Data\_addr

Window registers

Block\_count

Palette\_index

X, Y, W, H

Visible/alpha

Window0

Window1

…

Block\_addr

RAM

R|G|B

…

R|G|B

P2

…

P1

P0

BLOCK\_Y

BLOCK\_X

Ingredient\_index

Block\_index

Block\_index

Ingredient\_index

BLOCK\_X

BLOCK\_Y

Block\_index

Ingredient\_index

BLOCK\_X

BLOCK\_Y

系统需要如下几组寄存器

1. **一组全局寄存器**

|  |  |  |
| --- | --- | --- |
| 名称 | 功能 | BITS |
| OSD\_WIDTH |  | 16 |
| OSD\_HEIGHT |  | 16 |
| OSD\_WINDOW\_DATA\_SIZE | OSD\_WINDOW数据结构大小 | 8 |
| OSD\_INGREDIENT\_DATA\_SIZE |  | 8 |
| OSD\_PALETTE\_DATA\_SIZE |  | 8 |
| OSD\_MODIFIER\_DATA\_SIZE |  | 8 |

1. **多组Palette寄存器**

第一组Palette

|  |  |  |
| --- | --- | --- |
| 名称 | 功能 | BITS |
| FORMAT | 1. 无效PALETTE 2. 24 bit RGB 3. 灰度等级 4. 查找表LUT | 8 |
| PIXEL\_BITS | 每个Color使用的BIT数 | 8 |
| ENTRY\_COUNT | LUT的表格数目 | 16 |
| LUTS\_ADDR | LUT地址 | 32 |

第二组Palette

|  |  |  |
| --- | --- | --- |
| 名称 | 功能 | BITS |
| FORMAT | 1. 无效PALETTE 2. 24 bit RGB 3. 灰度等级 4. 查找表LUT | 8 |
| PIXEL\_BITS | 每个Color使用的BIT数 | 8 |
| ENTRY\_COUNT | LUT的表格数目 | 16 |
| LUTS | LUT地址 | 32 |

…

1. **多组Window寄存器**

**第一组窗口在最底层， 第n组在最上层**

第一组Window寄存器

|  |  |  |
| --- | --- | --- |
| 名称 | 功能 | BITS |
| PALETTE\_INDEX | 当前窗口使用的是第几组PALETTE  0xFF为无效 | 8 |
| VISIBLE | 窗口是否显示 | 8 |
| ALPHA | 窗口的ALPHA值 0 -255  0表示全透明  255表示不透明 | 8 |
| ZORDER | Z Order顺序 | 8 |
| X, Y | 窗口所在位置 | 16, 16 |
| WIDTH, HEIGHT | 窗口大小 | 16, 16 |
| BLOCK\_COUNT | 窗口BLOCK数量 | 32 |
| BLOCKS | 窗口BLOCK描述数组起始地址 | 32 |

BLOCKS指向RAM中的BLOCKS数组，每个BLOCK的数据结构如下

|  |  |  |
| --- | --- | --- |
| 名称 | 功能 | BITS |
| BLOCK\_INDEX | 第几个BLOCK | 8 |
| INGREDIENT\_INDEX | 使用的INGREDIENT编号 | 8 |
| BLOCK\_X, BLOCK\_Y | BLOCK在WINDOW中的位置 | 16,16 |
| BLOCK\_INDEX | 第几个BLOCK | 8 |
| INGREDIENT\_INDEX | 使用的INGREDIENT编号 | 8 |
| BLOCK\_X, BLOCK\_Y | BLOCK在WINDOW中的位置 | 16,16 |
| BLOCK\_INDEX | 第几个BLOCK | 8 |
| INGREDIENT\_INDEX | 使用的INGREDIENT编号 | 8 |
| BLOCK\_X, BLOCK\_Y | BLOCK在WINDOW中的位置 | 16,16 |
| … | … | … |

第二组WINDOW寄存器

…

1. **多组Modifier寄存器**

第一组Modifier寄存器

|  |  |  |
| --- | --- | --- |
| 名称 | 功能 | BITS |
| TYPE | 0, 无效  1, Move  2, Flip | 8 |
| INTERVAL | 执行间隔帧数 | 8 |
| LIMIT | 最大执行次数 | 8 |
| ACTIVE | 是否启用 | 8 |
| WINDOW1 | 作用的WINDOW编号, 0xFF为无效 | 8 |
| WINDOW2 | 作用的WINDOW编号, 0xFF为无效 | 8 |
| WINDOW3 | 作用的WINDOW编号, 0xFF为无效 | 8 |
| WINDOW4 | 作用的WINDOW编号, 0xFF为无效 | 8 |
| BLOCK1 | 作用的WINDOW编号+BLOCK编号, 0xFFFF为无效 | 16 |
| BLOCK2 | 作用的WINDOW编号+BLOCK编号, 0xFFFF为无效 | 16 |
| BLOCK3 | 作用的WINDOW编号+BLOCK编号, 0xFFFF为无效 | 16 |
| BLOCK4 | 作用的WINDOW编号+BLOCK编号, 0xFFFF为无效 | 16 |

第二组Modifier寄存器

**…**

1. **多组Ingredient寄存器**

|  |  |  |
| --- | --- | --- |
| 名称 | 功能 | BITS |
| TYPE | 0, 无效  1, Rectangle  2, Line  3, Glyph  4, Bitmap | 8 |
| PALETTE\_INDEX | 当前窗口使用的是第几组PALETTE  0xFF为无效 | 8 |
| flags |  | 16 |
| TYPE == Rectangle时寄存器功能(5 DWORDS) | | |
| WIDTH, HEIGHT | 矩形宽度高度 | 16,16 |
| BORDER\_COLOR[4] | 上下左右边框颜色 | 8,8,8,8 |
| BORDER\_WEIGHT | 边框宽度 | 8 |
| BORDER\_STYLE | 边框线段样式  const u8 OSD\_LINE\_STYLE\_SOLID = 1;  const u8 OSD\_LINE\_STYLE\_DASH = 2;  const u8 OSD\_LINE\_STYLE\_DOT1 = 3;  const u8 OSD\_LINE\_STYLE\_DOT2 = 4;  const u8 OSD\_LINE\_STYLE\_DOT3 = 5;  const u8 OSD\_LINE\_STYLE\_DASH\_DOT = 6;  const u8 OSD\_LINE\_STYLE\_DASH\_DOT\_DOT = 7; | 8 |
| BGCOLOR\_START  BGCOLOR\_END | 矩形背景颜色开始，结束 | 8, 8 |
| GRADIENT\_MODE |  | 8 |
| RESERVED |  | 8,8,8 |
| TYPE == Line时寄存器功能(5 DWORDS) | | |
| X1, Y1, X2, Y2 | 直线坐标 | 16,16,16,16 |
| WEIGHT | 线段宽度 | 8 |
| STYLE | 线段样式  const u8 OSD\_LINE\_STYLE\_SOLID = 1;  const u8 OSD\_LINE\_STYLE\_DASH = 2;  const u8 OSD\_LINE\_STYLE\_DOT1 = 3;  const u8 OSD\_LINE\_STYLE\_DOT2 = 4;  const u8 OSD\_LINE\_STYLE\_DOT3 = 5;  const u8 OSD\_LINE\_STYLE\_DASH\_DOT = 6;  const u8 OSD\_LINE\_STYLE\_DASH\_DOT\_DOT = 7; | 8 |
| COLOR | 线段颜色 | 8 |
| RESERVED |  | 8 |
| RESERVED32 |  | 32 |
| TYPE == GLYPH时寄存器功能(5 DWORDS) | | |
| BITMAP\_LEFT  BITMAP\_TOP | BITMAP有效数据开始位置 | 8,8 |
| BITMAP\_WIDTH  BITMAP\_HEIGHT | BITMAP数据宽度高度 | 8,8 |
| BITMAP\_PITCH | BITMAP\_PICTH | 8 |
| COLOR | 字体颜色 | 8 |
| FONT\_WIDTH | 字体宽度 | 8 |
| RESERVED |  | 8 |
| CHAR\_CODE | 字体编码 | 16 |
| DATA\_SIZE | BITMAP数据长度 | 16 |
| BITMAP\_DATA | BITMAP数据内存地址 | 32 |
| TYPE == BITMAP时寄存器功能(5 DWORDS) | | |
| WIDTH, HEIGHT | BITMAP大小 | 16,16 |
| BITMAP\_COUNT | 包含位图熟练，默认为1 | 8 |
| RESERVED |  | 8 |
| DATA\_SIZE | BITMAP数据长度 | 16 |
| BITMAP\_DATA | BITMAP数据内存地址 | 32 |
| RESERVED32 |  | 32 |