Fancy OSD Engine Design

目录

[1. Overview 2](#_Toc505062867)

[1.1 OSD Definition 2](#_Toc505062868)

[1.2 Hardware memory layout 3](#_Toc505062869)

[1.3 Entity relationship 4](#_Toc505062870)

[1.4 Features 4](#_Toc505062871)

[1.5 Design consideration 4](#_Toc505062872)

[1.6 Data path 5](#_Toc505062873)

[2. Entity Attribute Definition 5](#_Toc505062874)

[Engine(帧) 5](#_Toc505062875)

[Palette(调色板) 5](#_Toc505062876)

[Window(窗口) 6](#_Toc505062877)

[Block(区块) 6](#_Toc505062878)

[Ingredient(素材 – 抽象类) 6](#_Toc505062879)

[Bitmap(位图 – 继承自Ingredient) 6](#_Toc505062880)

[Glyph(字体 -继承自Ingredient) 7](#_Toc505062881)

[Plot(绘图 – 抽象类 -继承自Ingredient) 7](#_Toc505062882)

[Pixel(点 -继承自Plot) 7](#_Toc505062883)

[Line(线 -继承自Plot) 8](#_Toc505062884)

[Rectangle(矩形 - 继承自Plot) 8](#_Toc505062885)

[Circle(矩形 - 继承自Plot) 8](#_Toc505062886)

[Arc(弧 - 继承自Plot) 9](#_Toc505062887)

[Oval(椭圆 - 继承自Plot) 9](#_Toc505062888)

[Modifier(修改器) 9](#_Toc505062889)

[Move(移动 – 继承自Modifier) 10](#_Toc505062890)

[Resize(改变大小 – 继承自Modifier) 10](#_Toc505062891)

[Rotate(旋转 – 继承自Modifier) 10](#_Toc505062892)

[Flip(翻转 – 继承自Modifier) 11](#_Toc505062893)

[Shift(图片更换 – 继承自Modifier) 11](#_Toc505062894)

[Fade(淡入淡出 – 继承自Modifier) 11](#_Toc505062895)

[3. Bitmap data compression 12](#_Toc505062896)

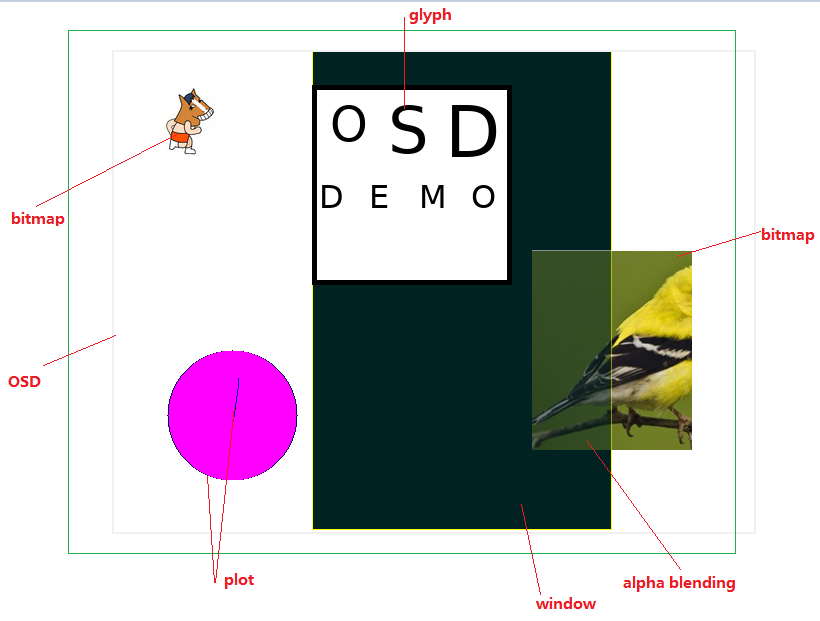
[Multi Palette 12](#_Toc505062897)

[Color index compress 12](#_Toc505062898)

[Fixed Palette 12](#_Toc505062899)

# 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 |

## Resize(改变大小 – 继承自Modifier)

修改窗口或者Block的大小

|  |  |  |
| --- | --- | --- |
| 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 | 大小修改的单位 |
| x\_end, y\_end | UINT16 | 修改的终点 |

## Rotate(旋转 – 继承自Modifier)

旋转窗口或者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* |
| angle | UINT8 | 旋转的角度pi/angle |
| angle\_end | UINT16 | 旋转的终点, 不设置则为无限循环 |

## Flip(翻转 – 继承自Modifier)

修改窗口或者Plot对象的大小

|  |  |  |
| --- | --- | --- |
| 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 | 翻转的步长 |
| x\_end, y\_end | UINT16 | 翻转的终点 |

## Shift(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