

i51 Application Programming Interface

Reference Manual

Version 1.0.0

Editor : Otto Li

2011-11-30

Chapter 1

Summary of data define

1.1 数据结构

结构名称	成员名称	成员类型	说明
iRECT	X	iS16	左上角 X 坐标
	Y	iS16	左上角 Y 坐标
	Width	iS16	矩形区域宽度
	Height	iS16	矩形区域高度

1.2 宏定义

宏名称	定义详情	说明
i51KitG2COLOR(R, G, B)	(((R) >> 3) << 11) (((G) >> 2) << 5) (((B) >> 3)	将 RGB888 的颜色值转为 RGB565
iS8	char	
iU8	unsigned char	
iS16	signed short	
iU16	unsigned short	
iS32	signed int	
iU32	unsigned int	
iS64	signed long	
iU64	unsigned long	
iBOOL	char	
iTRUE	1	
iFALSE	0	
iNULL	0	
iFILE	iU32	文件指针
iHANDLE	void*	句柄
iCOLOR	iU16	系统颜色类型
iALPHA	iU8	系统 alpha 值类型

Chapter 2

Summary of i51KitG2 Interfaces

2.1 i51KitG2 Interfaces

[i51KitG2DrawPoint](#)

[i51KitG2DrawLine](#)

[i51KitG2DrawPolygon](#)

[i51KitG2DrawArc](#)

[i51KitG2DrawCircle](#)

[i51KitG2FillPolygon](#)

[i51KitG2FillArc](#)

[i51KitG2FillCircle](#)

[i51KitG2CreateImageContainer](#)

[i51KitG2CloseImageContainer](#)

[i51KitG2CheckImageContainer](#)

[i51KitG2OpenImageContainer](#)

[i51KitG2DestroyImageContainer](#)

[i51KitG2SetContainerContentVersion](#)

[i51KitG2UnsetImageMemoryCopy](#)

[i51KitG2GetImageAttr](#)

[i51KitG2DrawImage](#)

[i51KitG2MirrorImage](#)

[i51KitG2FlipImage](#)

[i51KitG2RotateImage](#)

[i51KitG2ImageFillRect](#)

[i51KitG2FontCachEnable](#)

[i51KitG2FontCachDisable](#)

[i51KitG2FontGetDemension](#)

[i51KitG2FontDraw](#)

[i51KitG2VPEnable](#)

[i51KitG2VPDisable](#)

[i51KitG2VPCreate](#)

[i51KitG2VPDestroy](#)

[i51KitG2VPSetCoord](#)

[i51KitG2VPSetActivity](#)

[i51KitG2VPGetActivity](#)

[i51KitG2VPClean](#)

[i51KitG2VPGetAttribute](#)

[i51KitG2VPiSFull](#)

[i51KitG2VPRender](#)

[i51KitG2Update](#)

[i51KitG2CleanScreen](#)

[i51KitG2SetAlpha](#)

[i51KitG2ScreenZoom](#)

[i51KitG2UpdataContainerFromImage](#)

[i51KitG2UpdataContainerFromRe](#)

[i51KitG2ContainerImageIdExist](#)

[i51KitG2GetContainerAttr](#)

[i51KitG2ContainerDeleteImage](#)

[i51KitG2StopDecode](#)

Chapter 2.2

i51KitG2 Interfaces Reference Pages

Name

i51KitG2DrawPoint– 绘制一个点

C Specification

iBOOL i51KitG2DrawPoint(iS32 X, iS32 Y, iCOLOR Color)

Parameters

X 绘制点的 X 坐标

Y 绘制点的 Y 坐标

Color 绘制点的颜色

Returns

成功 iTRUE ,失败 iFALSE。

Notes

Name

i51KitG2DrawLine— 绘制一条直线

C Specification

```
iBOOL i51KitG2DrawLine( iS32 Point[][2], iCOLOR Color)
```

Parameters

textid 字符串资源 id 的 id

Point[][2] :

Point[0][0]是需要绘制线段的起点 X 坐标

Point[0][1]是需要绘制线段的起点 Y 坐标

Point[1][0]是需要绘制线段的终点 X 坐标

Point[1][1]是需要绘制线段的终点 Y 坐标

Color：绘制线段的颜色

Returns

成功 iTRUE,失败 iFALSE。

Notes

Name

i51KitG2DrawPolygon— 绘制一个多边形

C Specification

iBOOL i51KitG2DrawPolygon(iU32 VerticeNumber, iS32 Point[][2], iCOLOR Color)

Parameters

VerticeNumber: 绘制多边形的顶点个数

Point[][2]:

Point[0][0]是需要绘制线段的起点 X 坐标

Point[0][1]是需要绘制线段的起点 Y 坐标

Point [Num-1][0]是需要绘制多边形的最后一个顶点 X 坐标

Point [Num-1][1]是需要绘制多边形的最后一个顶点 X 坐标

Color: 绘制多边形的颜色

Returns

成功 iTRUE,失败 iFALSE。

Notes

函数功能是绘制有 VerticeNumber 个顶点,顶点坐标用 Point 存放的多边形
Point 数组中存放的顶点必须是有序的,且顶点个数必须大于等于 3,例如多边形由线段 AB, 线段 BC, 线段 CD, 线段 DE, 线段 EA 连接组成,

则 Point[0][0]存放顶点 A 的 X 坐标,Point[0][1]存放顶点 A 的 Y 坐标

则 Point[1][0]存放顶点 B 的 X 坐标,Point[1][1]存放顶点 B 的 Y 坐标

则 Point[2][0]存放顶点 C 的 X 坐标,Point[2][1]存放顶点 C 的 Y 坐标

则 Point[3][0]存放顶点 D 的 X 坐标,Point[3][1]存放顶点 D 的 Y 坐标

则 Point[4][0]存放顶点 E 的 X 坐标,Point[4][1]存放顶点 E 的 Y 坐标

Name

i51KitG2DrawArc— 绘制一段弧线

C Specification

iBOOL i51KitG2DrawArc(iS32 Point[][2], iCOLOR Color)

Parameters

Point[][2] :

Point[0][0]是需要绘制弧线的起点 X 坐标

Point[0][1]是需要绘制弧线的起点 Y 坐标

Point[1][0]是经过弧线上任意点的 X 坐标

Point[1][1]是经过弧线上任意点的 Y 坐标

Point[2][0]是需要绘制弧线的终点 X 坐标

Point[2][1]是需要绘制弧线的终点 Y 坐标

Color：绘制弧线的颜色

Returns

成功 iTRUE ,失败 iFALSE。

Notes

不在同一直线上的三点才能确定一个圆，请确定给出的三点坐标不在同一直线上

Name

i51KitG2DrawCircle— 绘制一个圆

C Specification

```
iBOOL i51KitG2DrawCircle( iS32 X, iS32 Y, iU32 R, iCOLOR Color )
```

Parameters

X 圆心 X 坐标

Y 圆心 Y 坐标

R 圆半径

Color 绘制圆的颜色

Returns

成功 iTRUE, 失败 iFALSE。

Notes

Name

i51KitG2FillPolygon- 绘制一个填充多边形

C Specification

iBOOL i51KitG2FillPolygon (iU32 VerticeNumber, iS32 Point[][2], iCOLOR Color)

Parameters

VerticeNumber 绘制多边形的顶点个数

Point[][2]

Point[0][0]是需要绘制线段的起点 X 坐标

Point[0][1]是需要绘制线段的起点 Y 坐标

Point [Num-1][0]是需要绘制多边形的最后一个顶点 X 坐标

Point [Num-1][1]是需要绘制多边形的最后一个顶点 X 坐标

Color 填充多边形的填充颜色

Returns

成功 iTRUE,失败 iFALSE。

Notes

函数功能是绘制有 VerticeNumber 个顶点,顶点坐标用 Point 存放的多边形
Point 数组中存放的顶点必须是有序的,且顶点个数必须大于等于 3,例如多边形由线段 AB, 线段 BC, 线段 CD, 线段 DE, 线段 EA 连接组成,

则 Point[0][0]存放顶点 A 的 X 坐标,Point[0][1]存放顶点 A 的 Y 坐标

则 Point[1][0]存放顶点 B 的 X 坐标,Point[1][1]存放顶点 B 的 Y 坐标

则 Point[2][0]存放顶点 C 的 X 坐标,Point[2][1]存放顶点 C 的 Y 坐标

则 Point[3][0]存放顶点 D 的 X 坐标,Point[3][1]存放顶点 D 的 Y 坐标

则 Point[4][0]存放顶点 E 的 X 坐标,Point[4][1]存放顶点 E 的 Y 坐标

Name

i51KitG2FillArc— 绘制一个填充弧线

C Specification

iBOOL i51KitG2FillArc (iS32 Point[][2], iCOLOR Color)

Parameters

Point[][2]

Point[0][0]是需要绘制弧线的起点 X 坐标

Point[0][1]是需要绘制弧线的起点 Y 坐标

Point[1][0]是经过弧线上任意点的 X 坐标

Point[1][1]是经过弧线上任意点的 Y 坐标

Point[2][0]是需要绘制弧线的终点 X 坐标

Point[2][1]是需要绘制弧线的终点 Y 坐标

Color 绘制弧线的颜色

Returns

成功 iTRUE ,失败 iFALSE。

Notes

不在同一直线上的三点才能确定一个圆，请确定给出的三点坐标不在同一直线上

Name

i51KitG2FillCircle— 绘制一个填充圆

C Specification

iBOOL i51KitG2FillCircle (iS32 X, iS32 Y, iU32 R, iCOLOR Color)

Parameters

X 圆心 X 坐标

Y 圆心 Y 坐标

R 圆半径

Color 填充圆的颜色

Returns

成功 iTRUE, 失败 iFALSE。

Notes

Name

i51KitG2CreateImageContainer– 创建图片容器

C Specification

iBOOL

i51KitG2CreateImageContainer

```
(  
    iFILE File , iU32* ImageId , iU32 TotalImage , iU16* ContainerPathName,  
    void(*Process)( iU32 Step , iU32 Total , iHANDLE Handle) ,  
    void (*Done)(iHANDLE)  
)
```

Parameters

File 待创建的图片容器的数据来源资源文件指针，是用 i51KitStdResOpen 函数打开的一个文件句柄

ImageId 图片资源 ID 数组

TotalImage 待创建的图片容器中含有图片的总数

ContainerPathName 创建的容器文件完整文件名

Process 每解码完一张图片的回调函数，Step 是解码进度，也就当前已经解完的图片张数，Total 是总需要解码的图片张数，Handle 是解码创建的容器文件句柄

Done 解码完成或者解码失败的回调函数，解码成功 iHANDLE 传解码创建的容器文件的句柄，解码失败传 iNULL

Returns

无返回值

Notes

i51KitG2CreateImageContainer 函数中开了一个定时器，每次进来解码完一张图片后调用回调函数函数 void(*Process)(iU32 Step, iU32 Totall, iHANDLE Handle)，全部解码完成或者解码失败会调用回调函数 void(*Done)(iHANDLE)同时关闭定时器。

如果解码还没完成需要暂停解码可以调用函数 i51KitG2Pause()来挂起解码,处理完后可以调用函数 i51KitG2Resume()恢复现场继续解码。

Name

i51KitG2CloseImageContainer– 关闭打开的图片容器

C Specification

iBOOL i51KitG2CloseImageContainer (iHANDLE Handle)

Parameters

Handle 打开的图片容器文件句柄

Returns

成功 iTRUE ,失败 iFALSE。

Notes

Name

i51KitG2CheckImageContainer– 检测图片容器的完整性和合法性

C Specification

```
iBOOL i51KitG2CheckImageContainer ( iU16* ContainerPathName )
```

Parameters

ContainerPathName 图片容器的相对路径和名称

Returns

合法返回 iTRUE ,否则返回 iFALSE。

Notes

Name

i51KitG2OpenImageContainer– 打开图片容器文件

C Specification

```
iHANDLE i51KitG2OpenImageContainer ( iU16* ContainerPathName)
```

Parameters

ContainerPathName 图片容器的相对路径和名称

Returns

成功打开返回打开的容器文件句柄，否则返回 iNULL

Notes

Name

i51KitG2DestroyImageContainer– 销毁容器文件

C Specification

iBOOL i51KitG2DestroyImageContainer

(iHANDLE ImageContainer, iU16* ContainerPathName)

Parameters

ImageContainer 如果需要销毁的容器文件已经打开，则这个参数传打开的容器文件句柄，反正传 iNULL

ContainerPathName 图片容器的相对路径和名称

Returns

成功销毁返回 iTRUE ,否则返回 iFALSE。

Notes

Name

i51KitG2SetContainerContentVersion— 往容器文件中写入自定的版本号

C Specification

```
iBOOL i51KitG2SetContainerContentVersion( iHANDLE Handle, iU32 Version)
```

Parameters

Handle 打开的容器文件句柄

Version 写入的版本号

Returns

成功 iTRUE ,失败 iFALSE。

Notes

i51KitG2SetContainerContentVersion 一般和 i51KitG2GetContainerContentVersion 配合使用以确保存在的容器文件是否为我们需要的文件

Name

i51KitG2GetContainerContentVersion— 从容器文件中读取自定的版本号

C Specification

```
iBOOL i51KitG2GetContainerContentVersion( iHANDLE Handle, iU32 *Version)
```

Parameters

Handle 打开的容器文件句柄

Version 获得的版本号指针

Returns

成功 iTRUE , 失败 iFALSE。

Notes

i51KitG2GetContainerContentVersion 一般和 i51KitG2SetContainerContentVersion 配合使用以确保存在的容器文件是否为我们需要的文件

Name

i51KitG2SetImageMemoryCopy– 加载图片到内存

C Specification

iU32 i51KitG2SetImageMemoryCopy

(iHANDLE Handle, iU32 TotallImage, iU32 ImageId[])

Parameters

Handle 打开的容器文件句柄

TotallImage 需要加载进到内存的图片数量

ImageId 需要加载到内存的图片 ID 数组，如果传 iNULL 则把这个容器中的所有图片加载进内存

Returns

返回成功加载到内存的图片数量

Notes

如果内存不足以把所有待加载的图片加载进内存，则会把 ImageId 中靠前的图片加载到内存，图片加载进内存后绘制速度更快，在内存允许的前提下建议把当前需要绘制的图片加载进内存

Name

i51KitG2UnsetImageMemoryCopy– 卸载内存中的图片

C Specification

iU32 i51KitG2UnsetImageMemoryCopy

(iHANDLE Handle, iU32 TotallImage, iU32 ImageId[])

Parameters

Handle 打开的容器文件句柄

TotallImage 需要从内存卸载的图片数量

ImageId 需要从内存卸载的图片 ID 数组，如果传 iNULL 则把这个容器中的所有图片从内存中卸载

Returns

返回成功卸载的图片数量

Notes

Name

i51KitG2GetImageAttr– 获得图片的属性

C Specification

```
iBOOL i51KitG2GetImageAttr  
  
(  
  
    iHANDLE ContainerHandle,iU32 ImageID , iU32 * ImageWidth,  
  
    iU32 * ImageHeight, iU8 *MemoryCopy  
  
)
```

Parameters

ContainerHandle 打开的容器文件句柄
ImageID 图片 ID
ImageWidth 图片宽度
ImageHeight 图片高度
MemoryCopy 图片是否加载进内存

Returns

成功 iTRUE ,失败 iFALSE。

Notes

Name

i51KitG2DrawImage— 按正常方式绘制图片

C Specification

iBOOL i51KitG2DrawImage

(iHANDLE ContainerHandle,iU32 ImageID , iRECT* DesRect , iRECT* SrcRect)

Parameters

ContainerHandle 待绘制图片所在容器的句柄

ImageID 图片 ID

DesRect 图片绘制的目的区域

ImageHeight 图片绘制的裁剪区

Returns

成功 iTRUE , 失败 iFALSE。

Notes

Name

i51KitG2MirrorImage– 按镜像方式绘制图片

C Specification

iBOOL i51KitG2MirrorImage

(iHANDLE ContainerHandle,iU32 ImageID , iRECT* DesRect , iRECT* SrcRect)

Parameters

ContainerHandle 待绘制图片所在容器的句柄

ImageID 图片 ID

DesRect 图片绘制的目的区域

ImageHeight 图片绘制的裁剪区

Returns

成功 iTRUE , 失败 iFALSE。

Notes

Name

i51KitG2FlipImage– 按倒立方式绘制图片

C Specification

iBOOL i51KitG2FlipImage

(iHANDLE ContainerHandle,iU32 ImageID , iRECT* DesRect , iRECT* SrcRect)

Parameters

ContainerHandle 待绘制图片所在容器的句柄

ImageID 图片 ID

DesRect 图片绘制的目的区域

ImageHeight 图片绘制的裁剪区

Returns

成功 iTRUE , 失败 iFALSE。

Notes

Name

i51KitG2RotatImage— 任意角度旋转图片后绘制图片

C Specification

iBOOL

i51KitG2RotatImage

```
(  
  
    iHANDLE ContainerHandle, iU32 ImageID , iRECT* DesRect , iRECT* SrcRect,  
  
    iS32 SrcMoveX , iS32 SrcMoveY, iS32 RotateCenterX , iS32 RotateCenterY ,  
  
    iU32 RotateType, iS32 Angle  
  
)
```

Parameters

ContainerHandle 待绘制图片所在容器的句柄

ImageID 图片 ID

DesRect 图片绘制的目的区域

ImageHeight 图片绘制的裁剪区

SrcMoveX 图片旋转前平移至点 (SrcMoveX,SrcMoveY)

SrcMoveY 图片旋转前平移至点 (SrcMoveX,SrcMoveY)

RotateCenterX 图片旋转中心 X 坐标

RotateCenterY 图片旋转中心 Y 坐标

RotateType 旋转类型，一般填 0

Angle 旋转角度

Returns

成功 iTRUE , 失败 iFALSE。

Notes

Angle 为正是顺时针方向旋转，为负按逆时针方向旋转

Name

i51KitG2ImageFillRect– 用一张图片填充一个区域

C Specification

iBOOL i51KitG2ImageFillRect

(iHANDLE ContainerHandle,iU32 ImageID , iRECT* DesRect , iRECT* SrcRect)

Parameters

ContainerHandle 待绘制图片所在容器的句柄

ImageID 图片 ID

DesRect 图片填充的目的区域

ImageHeight 图片绘制的裁剪区

Returns

成功 iTRUE , 失败 iFALSE。

Notes

Name

i51KitG2FontCachEnable— 绘制文字加速

C Specification

iBOOL i51KitG2FontCachEnable (iFILE ReFile, iU32 TypeheadID)

Parameters

ReFile 绘制文字字库所在的资源文件

TypeheadID 字库文件在资源文件中的 ID

Returns

成功 iTRUE , 失败 iFALSE。

Notes

使用这个函数实际是将常用汉字的字模加载进内存，常用汉字在字库生成的时候可以自己指定

Name

i51KitG2FontCachDisable— 取消绘制文字加速

C Specification

iBOOL i51KitG2FontCachDisable (iFILE ReFile, iU32 TypeheadID)

Parameters

ReFile 绘制文字字库所在的资源文件

TypeheadID 字库文件在资源文件中的 ID

Returns

成功 iTRUE , 失败 iFALSE。

Notes

卸载加载进内存的常用字库字模

Name

i51KitG2FontGetDemension– 获得需要绘制文字的区域宽高

C Specification

iBOOL

i51KitG2FontGetDemension

```
(  
  
    iFILE ReFile, iU32 TypeheadID, iU32 WordsNum, iU32 RowWords,  
  
    iU8 WordsDirection , iS16 RowGap, iS16 ColumnGap , iU32 * DrawWidth,  
  
    iU32 * DrawHeight  
  
)
```

Parameters

ReFile 绘制文字字库所在的资源文件
TypeheadID 字库文件在资源文件中的 ID
WordsNum 需要绘制文字的个数
RowWords 每行绘制文字的个数
WordsDirection 绘制文字方向，取值为 i51KITG2_FONT_WORDS_ROTATE_0 正常显示，取值为 i51KITG2_FONT_WORDS_ROTATE_90 旋转 90°
RowGap 绘制文字行间距
ColumnGap 绘制文字字间距
DrawWidth 绘制文字需要的宽度
DrawHeight 绘制文字需要的高度

Returns

成功 iTRUE, 失败 iFALSE。

Notes

Name

i51KitG2FontDraw— 绘制文字

C Specification

iBOOL

i51KitG2FontDraw

(

iFILE Handle, iU32 TypeheadID, void* text , iU16 WordsNum,

iRECT* DrawRect , iRECT * SrcRect, iU16 DrawDirect, iS16 WordsRowGap,

iS16 WordsColumnGap, iCOLOR fgcolor , iCOLOR bgcolor, iU16 Effect

)

Parameters

Handle 绘制文字字库所在的资源文件

TypeheadID 字库文件在资源文件中的 ID

Text 字符串指针

WordsNum 需要绘制文字的个数

DrawRect 绘制文字的目的区域

SrcRect 绘制文字的原始区域

WordsDirection 绘制文字方向，取值为 i51KITG2_FONT_WORDS_ROTATE_0 正常显示，取值为 i51KITG2_FONT_WORDS_ROTATE_90 旋转 90°

WordsRowGap 绘制文字行间距

WordsColumnGap 绘制文字字间距

fgcolor 绘制文字颜色

bgcolor 绘制文字背景颜色

Effect 绘制特效，包括是否需要下划线和是否需要背景色，如果都不需要则传 0，只需下划线传 i51KITG2_FONT_UNDERLINE，只需背景色传

i51KITG2_FONT_BKCOLOR，两者都需要传

i51KITG2_FONT_UNDERLINE|i51KITG2_FONT_BKCOLOR

Returns

成功 iTRUE , 失败 iFALSE。

Notes

Name

i51KitG2VPEnable— 设置绘制在 VP 上进行

C Specification

```
iBOOL i51KitG2VPEnable( void )
```

Parameters

Returns

成功 iTRUE ,失败 iFALSE。

Notes

调用此函数后所有的绘制(绘制图形、图像和文字)都是在 VP 上进行，这时的绘制是不会绘制到屏幕上需要调用 i51KitG2VPRender 函数才是真正的往屏幕上绘制

Name

i51KitG2VPDisable— 设置绘制在直接绘制到屏幕上

C Specification

```
iBOOL i51KitG2VPDisable( void )
```

Parameters

Returns

成功 iTRUE ,失败 iFALSE。

Notes

调用此函数后所有的绘制在屏幕上进行

Name

i51KitG2VPCreate— 新建一个 VP

C Specification

```
iHANDLE i51KitG2VPCreate ( iU16 TotalElement , iRECT * Rect , void * Reserve)
```

Parameters

TotalElement 新建 VP 上最多能绘制的元素个数，元素包括图形图像和文字

Rect 新建的 VP 大小

Reserve 保留字

Returns

成功返回新建 VP 的句柄，失败 iFALSE。

Notes

VP 是一个虚拟的层，创建 VP 的时候只是记录了 VP 的一些属性，没有真正分配内存，在上面绘制元素时实际上也只是记录相关信息，调用 i51KitG2VPRender 才会真正把元素绘制到屏幕上。

从 VP 创建到往 VP 上绘制元素到把元素显示到屏幕上，整个过程都没有增加额外的内存开销，所以相对真正的层来说节约了大量的内存。

引用 VP 最大好处是节约内存，同时与 i51KitG2VPSetCoord 配合使用就能轻松实现滚屏效果。

Name

i51KitG2VPDestroy– 销毁一个 VP

C Specification

iBOOL i51KitG2VPDestroy (iHANDLE vpanel)

Parameters

vpanel 需要销毁的 VP 句柄

Returns

成功 iTRUE , 失败 iFALSE。

Notes

Name

i51KitG2VPSetCoord– 移动 VP

C Specification

iBOOL i51KitG2VPSetCoord (iHANDLE vpanel , iS32 X, iS32 Y)

Parameters

vpanel 需要移动的 VP 句柄

X VP 移动后左上角相对屏幕的 X 坐标

Y VP 移动后左上角相对屏幕的 Y 坐标

Returns

成功 iTRUE , 失败 iFALSE。

Notes

Name

i51KitG2VPSetActivity– 设置当前使用的 VP

C Specification

```
iBOOL i51KitG2VPSetActivity ( iHANDLE vpanel )
```

Parameters

vpanel 设置使用的 VP 句柄

Returns

成功 iTRUE , 失败 iFALSE。

Notes

调用此函数后 vpanel 就是当前活动的 VP，所有的绘制都是在其上进行，直到再次调用 i51KitG2VPSetActivity 设置其他 VP 为当前活动 VP 或者调用 i51KitG2VPDisable 停止使用 VP

Name

i51KitG2VPGetActivity– 获得当前使用的 VP

C Specification

```
iHANDLE i51KitG2VPGetActivity ( void )
```

Parameters

Returns

当前使用的 VP 句柄

Notes

Name

i51KitG2VPClean— 清空 VP 上的元素

C Specification

```
iBOOL i51KitG2VPClean ( iHANDLE vpanel )
```

Parameters

vpanel 需要清空的 VP 句柄

Returns

成功 iTRUE , 失败 iFALSE

Notes

Name

i51KitG2VPGetAttribute—获得 V P 属性

C Specification

iBOOL i51KitG2VPGetAttribute

(iHANDLE vpanel , iRECT * Rect , iU16 * ResidueSpace , iU32 * Reserve)

Parameters

vpanel 待获得属性 VP 句柄

Rect VP 的区域

ResidueSpace VP 上剩余空间，也就是 VP 还能绘制元素个数

Reserve 保留字

Returns

成功 iTRUE , 失败 iFALSE

Notes

Name

i51KitG2VPiSFull—判断 VP 是否已经画满

C Specification

```
iBOOL i51KitG2VPiSFull( iHANDLE vpanel )
```

Parameters

vpanel 待判断 VP 句柄

Returns

如果已经画满返回 iTRUE，否则返回 iFALSE

Notes

Name

i51KitG2VPRender—将 VP 上的元素绘制到屏幕上

C Specification

```
iBOOL i51KitG2VPRender ( iHANDLE vpanel , iRECT * DrawRect)
```

Parameters

vpanel 待绘制 VP 句柄

DrawRect 需要绘制的区域

Returns

成功绘制所有 VP 上所有元素返回 iTRUE，否则返回 iFALSE

Notes

Name

i51KitG2Update—刷新屏幕

C Specification

```
iBOOL i51KitG2Update( void )
```

Parameters

Returns

成功 iTRUE, 否则 iFALSE

Notes

Name

i51KitG2CleanScreen—用指定颜色清除屏幕

C Specification

```
iBOOL i51KitG2CleanScreen(iCOLOR Color, iRECT *Rect)
```

Parameters

Color 清除屏幕的颜色

Rect 清除屏幕的区域

Returns

成功 iTRUE，否则 iFALSE

Notes

Name

i51KitG2SetAlpha—设置系统 Alpha

C Specification

```
iBOOL i51KitG2SetAlpha( iALPHA Alpha )
```

Parameters

Alpha 系统待设置的 Alpha 值

Returns

成功 iTRUE, 否则 iFALSE

Notes

调用此函数后, 所有的绘制函数都将以此 Alpha 来绘制, 此函数的有效期是从设置时开始直到下次设置结束。

Alpha 的等级从 0 到 255, 数字越大则绘制的元素越透明, 当为 255 时是全透明, 为 0 则完全不透明, 设置系统 Alpha>0 后绘制效率会有所降低

Name

i51KitG2ScreenZoom—缩放物理屏幕

C Specification

iBOOL i51KitG2ScreenZoom

(iRECT * DesRect, iRECT * SrcRect, iU32 ZoomRateX, iU32 ZoomRateY)

Parameters

DesRect: 影像缩放到的目的区域

SrcRect: 影像缩放的原始区域

ZoomRateX: 影像 X 方向缩放的比例

ZoomRateY: 影像 Y 方向缩放的比例

Returns

成功 iTRUE, 否则 iFASLE

Notes

将物理 Buffer 的指定区域进行缩放后显示到新的区域

Name

i51KitG2UpdataContainerFromImage—更新容器文件中的图片

C Specification

```
iBOOL i51KitG2UpdataContainerFromImage ( iHANDLE ContainerHandle , void *
```

```
ImageMess, iU8 MessType, iU32 DesImageId)
```

Parameters

ContainerHandle 容器句柄

ImageMess 源文件句柄

MessType 源文件句柄类型

DesImageId 要更新目的图片 id 号

Returns

成功 iTRUE, 否则 iFALSE

Notes

MessType 类型参考枚举: i51KitG2_MESSAGE_TYPE_FILE 文件句柄,

i51KitG2_MESSAGE_TYPE_MEMORY 内存

Name

i51KitG2UpdataContainerFromRe—更新容器文件中的图片

C Specification

iBOOL i51KitG2UpdataContainerFromRe (iHANDLE ContainerHandle , void *

ReMess, iU32 SrcImageld , iU8 MessType, iU32 DesImageld)

Parameters

ContainerHandle 容器句柄

ReMess 源文件句柄

SrcImageld 源图片 id 号

MessType 源文件句柄类型

DesImageld 要更新目的图片 id 号

Returns

成功 iTRUE, 否则 iFASLE

Notes

MessType 类型参考枚举: i51KitG2_MESSAGE_TYPE_FILE 文件句柄,

i51KitG2_MESSAGE_TYPE_MEMORY 内存

Name

i51KitG2ContainerImageldExist—判断容器中是否存在指定 ID 号图片

C Specification

```
iBOOL i51KitG2ContainerImageldExist( iHANDLE ContainerHandle, iU32 Imageld)
```

Parameters

ContainerHandle 容器句柄
Imageld 图片 id 号

Returns

成功 iTRUE，否则 iFALSE

Notes

Name

i51KitG2GetContainerAttr–获取容器属性

C Specification

```
iBOOL i51KitG2GetContainerAttr(iHANDLE ContainerHandle, iU32 * TotalCapacity,  
iU32 * UsedCapacity, iU32 * ResidueCapacity, iU32 * UsableMinID )
```

Parameters

ContainerHandle 容器句柄
TotalCapacity 总容量
UsedCapacity 已使用的容量
ResidueCapacity 剩余容量
UsableMinID 图片最小 id 号

Returns

成功 iTRUE，否则 iFALSE

Notes

Name

i51KitG2ContainerDeletelImage—删除容器文件中指定 ID 号的图片文件

C Specification

iBOOL i51KitG2ContainerDeletelImage(iHANDLE ContainerHandle, iU32 ImageId)

Parameters

ContainerHandle 容器句柄
ImageId 图片 id 号

Returns

成功 iTRUE，否则 iFALSE

Notes

Name

i51KitG2StopDecode—停止解码

C Specification

iBOOL i51KitG2StopDecode(iHANDLE Handle)

Parameters

Handle 容器句柄

Returns

成功 iTRUE，否则 iFALSE

Notes