

scaler 图形缩放处理模块

源代码目录: /lib/source/graphics/scaler
接口原始头文件: image_scaler.h bitmap_scaler.h
平台合并头文件: graphics.h

该模块可以高质量的缩小/放大位图或者图片,基于扫描行的处理方式,平台无关,节约内存,速度也较快.模块分为两部分:

- 1. image 缩放模块,直接从图片文件中进行缩放操作.
- 2. bitmap 缩放模块,通过自定义数据源回调函数,实现缩放.

image 缩放模块

image 缩放初始化

IMAGE_SCALER * image_scaler_init(const char * path, int dw, int dh);

参数	const char * path	图片文件路径
	int dw	缩放宽度
	int dh	缩放高度
返回值	IMAGE_SCALER *	返回scaler对象

1. 当返回值为NULL时,初始化失败,原因可能是文件未找到,或者内存不足,错误或不支持的图片格式.

image缩放读取一行

int image_scaler_readline(IMAGE_SCALER * iscaler, void * buf, int scanline);

参数	IMAGE_SCALER * iscaler	scaler对象
	void * buf	输出行缓存
	int scanline	扫描线计数
返回值	int	错误码

- 1. 返回值为0, 执行成功.
- 2. scanline是扫描线号,即当前读取的是图片的第几行数据. 这个扫描线序号每次读取只能增加不能减少,典型的是每读取一行, scanline+1.
- 3. buf的空间必须能够容纳缩放后的至少一行数据,否则将会数据溢出.
- 4. 返回的图像数据格式为32位的RGB格式(LCD_A8R8G8B8).

卸载scaler模块

int image_scaler_deinit(IMAGE_SCALER * iscaler);

参数	IMAGE_SCALER * iscaler	scaler 对象
返回值	int	错误码(0 – 成功)

- 1. scaler 使用完毕必须使用这个函数释放资源,否则会发生内存泄露.
- 2. 不要传入无效的值,否则将发生异常.

bitmap 缩放模块

bitmap 缩放模块初始化

```
BITMAP_SCALER * bitmap_scaler_init
( int sw, int sh, int dw, int dh,
int (*readline)( int line, void * buffer, void * param ),
void * param );
```

参数	int sw	源图像宽度
	int sh	源图像高度
	int dw	目标图像宽度
	int dh	目标图像高度
	int (*readline)(int line, void * buffer, void * param)	源图像读取一行用户回调函数
	void * param	用户回调函数的自定义参数
返回值	BITMAP_SCALER *	BITMAP_SCALER对象

1. readline函数是用户自定义回调函数, 功能是获取图像指定行的数据, 典型的如下面的示例:

```
const unsigned int bitmap[320*240];    //假设这是源位图数据
int readline( int line, void * buffer, void * param )
{
    memcpy( buffer, bitmap[320* line], 320*sizeof(unsigned int) );
    return 0;    //默认返回0值
}
```

图像数据格式为LCD_A8R8G8B8.

读取一行缩放后的图像

```
int bitmap_scaler_readline( BITMAP_SCALER * scaler, void * buf, int line );
```

参数	BITMAP_SCALER * scaler	BITMAP_SCALER对象
	void * buf	输出缓存
	int line	当前读取的扫描线
返回值	int	错误码(0—成功)

1. 返回值为0, 执行成功.
2. line是扫描线号, 即当前读取的是图片的第几行数据. 这个扫描线序号每次读取只能增加不能减少, 典型的是每读取一行, line+1.
3. buf的空间必须能够容纳缩放后的至少一行数据, 否则将会数据溢出.
4. 返回的图像数据格式为32位的RGB格式(LCD_A8R8G8B8).

卸载bitmap_scaler模块

```
int bitmap_scaler_deinit( BITMAP_SCALER * scaler );
```

参数	BITMAP_SCALER * scaler	scaler 对象
返回值	int	错误码(0 – 成功)

3. scaler 使用完毕必须使用这个函数释放资源，否则会发生内存泄露.
4. 不要传入无效的值，否则将发生异常.