第23章 LCD驱动API函数

µC/GUI 要求一个驱动器用于硬件。本章说明一个 LCD 驱动器能为µC/GUI 做什么，以及它为µC/GUI 提供什么函数（应用程序接口，或 API）。

在大多数情况下，你大概不需要读这一章，因为大多数µC/GUI 的对于 LCD 层的函数调用将会由 GUI 层完成。实际上，我们推荐你仅仅在没有 GUI 等价物（例如，如果你希望直接修改 LCD 控制器的查询表）的情况下调用 LCD 函数。原因是 LCD 驱动函数是非线程保护，不象它们的 GUI 等价物。因此它们在多任务环境不应当被直接调用。

23.1 LCD驱动API

下表按字母顺序列出了μC/GUI 与 LCD 有关的函数。函数详细的描述在下面部分给出。

LCD_L0：驱动器函数

函数	说明
初始化及显示控制组	
LCD_L0_Init ()	初始化显示屏。
LCD_L0_ReInit ()	不擦除内容而重新初始化 LCD。
LCD_L0_Off ()	关闭 LCD。
LCD_L0_On ()	开启 LCD。
绘制组	
LCD_L0_DrawBitmap ()	通用绘制位图函数。
LCD_L0_DrawHLine ()	绘制一条水平线。
LCD_L0_DrawPixel ()	以当前前景色绘制一个像素。
LCD_L0_DrawVLine ()	绘制一条垂直线。
LCD_L0_FillRect ()	直译一个矩形区域。
LCD_L0_SetPixelIndex ()	以指定颜色绘制一个像素。
LCD_L0_XorPixel ()	反转一个像素。
“Get”组	
LCD_L0_GetPixelIndex ()	返回指定像素的颜色的索引。
“Set”组	
LCD_L0_SetOrg ()	不再使用，保留给将来使用（必须在驱动器中存在）。
查询表组	
LCD_L0_SetLUTEntry ()	修改 LUT 的单个条目。
Misc.组（可选）	
LCD_L0_ControlCache ()	锁/解锁/清除 LCD 高速缓存。

LCD：LCD 层函数

函数	说明
LCD_GetXSize ()	返回 LCD 的物理 X 轴尺寸（以像素为单位）。
LCD_GetYSize ()	返回 LCD 的物理 Y 轴尺寸（以像素为单位）。
LCD_GetVXSize ()	返回 LCD 的虚拟 X 轴尺寸（以像素为单位）。
LCD_GetVYSize ()	返回 LCD 的虚拟 Y 轴尺寸（以像素为单位）。
LCD_GetBitsPerPixel ()	返回每像素的位数。
LCD_GetNumColors ()	返回有效颜色的位数。
LCD_GetFixedPalette ()	返回固定调色板模式。

23.2 初始化及显示控制组

LCD_L0_Init()

描述

使用 LCDConf.h 中的配置设置初始化 LCD。如果使用上一 GUI 层，该函数被 GUI_Init () 自动调用，因此该函数不需要手工调用。

函数原型

```
void LCD_L0_Init (void);
```

LCD_L0_ReInit()

描述

使用配置设置再次初始化 LCD，不删除显示屏的内容。

函数原型

```
void LCD_L0_ReInit (LCD_INITINFO* pInitInfo) ;
```

LCD_L0_Off ()，LCD_L0_On()

描述

分别打开或关闭显示。

函数原型

```
void LCD_L0_Off(void); void LCD_L0_On(void);
```

附加信息

函数的使用不影响图像存储器的内容或其它设置。因此你可以安全的关闭显示屏，然后重新打开而不需要更新内容。

23.3 绘制组

LCD_L0_DrawBitMap()

描述

绘制一个预先转换了的位图。

函数原型

```
LCD_L0_DrawBitMap ( int x0, int y0,
                    int Xsize, int Ysize,
                    int BitsPerPixel, int BytesPerLine,
                    const U8* pData, int Diff,
                    const LCD_PIXELINDEX* pTrans) ;
```

参　数	含　意
x0	要绘制的位图的左上角 X 轴坐标。
y0	要绘制的位图的左上角 Y 轴坐标。
Xsize	水平方向像素的数量。
Ysize	垂直方面像素的数量。
BitsPerPixel	每像素的位数。
BytesPerLine	图片每行的字节数。
pData	实际图片的指针，该数据定义了位图看起来象什么。
Diff	从左侧开始跳过的像素数。

LCD_L0_DrawHLine()

描述

在指定位置，使用当前前景颜色绘制一条一个像素宽的水平线。

函数原型

```
void LCD_L0_DrawHLine(int x0, int y, int x1);
```

参　数	含　意
x0	线段开始坐标
y	绘制线段的 Y 轴坐标。
x1	线段结束坐标。

附加信息

对于大多数 LCD 控制器，该函数执行速度非常快，因为可以立即放置多个像素，而不需要计算。很清楚，在绘制水平直线方面，该函数执行速度要比 DrawLine 函数快。

LCD_L0_DrawPixel()

描述

在指定位置，使用当前前景颜色绘制一个像素。

函数原型

```
void LCD_L0_DrawPixel(int x, int y);
```

参　数	含　意
x	绘制的像素的 X 轴坐标。
y	绘制的像素的 Y 轴坐标。

LCD_L0_DrawVLine()

描述

在指定位置，使用当前前景颜色绘制一条一个像素宽的垂直线。

函数原型

```
void LCD_L0_DrawVLine(int x, int y0, int y1);
```

参　数	含　意
x	绘制线段的 X 轴坐标。
y0	线段开始坐标。
y1	线段结束坐标。

附加信息

对于大多数 LCD 控制器，该函数执行速度非常快，因为可以立即放置多个像素，而不需要计算。很清楚，在绘制垂直直线方面，该函数执行速度要比 DrawLine 函数快。

LCD_L0_FillRect()

描述

在指定位置，使用当前前景颜色绘制一个填充的矩形。

函数原型

```
void LCD_L0_FillRect(int x0, int y0, int x1, int y1);
```

参　数	含　意
x0	左上角 X 轴坐标。
y0	左上角 Y 轴坐标。
x1	左下角 X 轴坐标。
y1	右下角 Y 轴坐标。

LCD_L0_SetPixelIndex()

描述

以指定的颜色绘制一个像素。

函数原型

```
void LCD_L0_SetPixelIndex(int x, int y, int ColorIndex);
```

参　数	含　意
x	绘制的像素的 X 轴坐标。
y	绘制的像素的 Y 轴坐标。
ColorIndex	使用的颜色。

LCD_L0_XorPixel()

描述

反转一个像素。

函数原型

```
void LCD L0XorPixel(int x, int y) ;
```

参 数	含 意
x	反转的像素的 X 轴坐标。
y	反转的像素的 Y 轴坐标。

23.4 “Get”组

LCD_L0_GetPixelIndex()

描述

返回指定像素的 RGB 颜色索引。

函数原型

```
int LCD_L0_GetPixelIndex(int x, int y);
```

参 数	含 意
x	像素的 X 轴坐标。
y	像素的 Y 轴坐标。

返回值

像素的索引。

附加信息

更多的信息请参阅第 9 章：颜色。

23.5 查询表（LUT）组

LCD_L0_SetLUTEntry()

描述

修改 LCD 控制器的 LUT 的单个条目。

函数原型

```
void LCD_L0_SetLUTEntry(U8 Pos, LCD_COLOR Color;
```

参　数	含　意
Pos	查询中的位置。应当小于颜色的数量，例如，对于 2bpp，为 0～3，4bpp 为 0～15，8bpp 为 0～255。
Color	24 位 RGB 值。最接近的可能的值将用于 LUT。如果一个颜色 LUT 被初始化，所有 3 个组成部分会被使用。在单色模式，使用绿色部分，但是依然推荐（为了更好的理解程序代码）将 3 种基色设置为同一个数值（例如 0x555555 或 0xa0a0a0）。

23.6 杂项组

LCD_L0_ControlCache()

描述

锁/解锁/清除 LCD 高速缓存。该函数可以用于将高速缓存设置为一个锁定状态，这样导致所有的在驱动器上的绘制操作转而在图像存储器缓存（在 CPU RAM 中）中实现，不过不会导致任何可见的输出。解锁或清除导致这样改变写入到显示屏当中。这可以帮助避免显示的闪烁，也可以加快绘制速度。它与有多少不同的绘制操作被执行无关；改变会立即写入显示屏。为了能够做到这样， LCD_SUPPORT_CACHECONTROL 必须在配置文件中启用。

函数原型

```
U8 LCD_ControlCache(U8 command);
```

参　数	含　意
Command	指定赋于高速缓存的命令。使用下表所示的符号值。

参数 Command 的允许值

LCD_CC_UNLOCK	设置默认模式：高速缓是透明的。
LCD_CC_LOCK	锁定高速缓存，直到高速缓存被解锁或清除，才能执行写操作。
LCD_CC_FLUSH	清除高速缓存，将所有修改了的数据写入视频 RAM 中。

返回值

缓存状态的信息。忽略。

附加信息

当高速缓存被锁定，驱动器维持一个“hitlist”——一列已经被修改并需要写入显示屏

的字节。该“hitlist”在图像存储器中每字节使用一位。

这是一个可选择的特性，并不支持所有的 LCD 驱动器。

范例

下面范例的代码在显示屏上执行交迭的绘制操作。为了加速显示屏的更新，避免闪烁，锁定高速缓存，直到这些操作执行完毕后才解锁。

```
LCD_ControlCache(LCD_CC_LOCK);
GUI_FillCircle(30,30,20);
GUI_SetDrawMode(GUI_DRAWMODE_XOR);
GUI_FillCircle(50,30,10);
GUI_SetTextMode(GUI_TEXTMODE_XOR);
GUI_DispStringAt ("Hello world\n", 0, 0);
GUI_DrawHLine(16,5,25);
GUI_DrawHLine(18,5,25);
GUI_DispStringAt("XOR Text",0,20);
GUI_DispStringAt("XOR Text",0,60);
LCD_ControlCache(LCD_CC_UNLOCK);
```

LCD_GetXSize()，LCD_GetYSize()

描述

分别返回 LCD 的物理 X 轴或 Y 轴尺寸，以像素为单位。

函数原型 s

```
int LCD_GetXSize (void) int LCD_GetYSize (void)
```

返回值

显示屏的物理 X/Y 轴尺寸。

LCD_GetVXSize()，LCD_GetVYSize()

描述

分别返回 LCD 的虚拟 X 轴或 Y 轴尺寸，以像素为单位。在大多数情况下，虚拟尺寸等于

物理尺寸。

函数原型

```
int LCD_GetVXSize (void), int LCD_GetVYSize (void)
```

返回值

显示屏的虚拟 X/Y 尺寸。

LCD_GetBitsPerPixel()

描述

返回每像素位的数量。

函数原型

```
int LCD_GetBitsPerPixel(void);
```

返回值

每像素位的数量。

LCD_GetNumColors()

描述

返回 LCD 当前有效颜色的数量。

函数原型

```
int LCD_GetNumColors(void);
```

返回值

有效颜色的数量。

LCD_GetFixedPalette()

描述

返回固定调色板的模式。

函数原型

```
int LCD_GetFixedPalette(void);
```

返回值

固定调色板模式。关于固定调色板的更多信息，请参阅第 9 章：“颜色”。