

UCGUI 函数表

By: Changing

更多嵌入式资料请访问 www.ichanging.org

1. 文本显示

函 数	说 明
显示文本的函数	
GUI_Dispatch()	在当前坐标显示单个字符
GUI_DispatchAt()	在指定坐标显示单个字符
GUI_DispatchChars()	按指定重复次数显示一个字符
GUI_DispatchString()	在当前坐标显示字符串
GUI_DispatchStringAt()	在指定坐标显示字符串
GUI_DispatchStringAtCEOL()	在指定坐标显示字符串，并清除到行末
GUI_DispatchStringInRect()	在指定矩形区域内显示字符串
GUI_DispatchStringLength()	在当前坐标显示指定字符数量的字符串
选择文本绘图模式	
GUI_SetTextMode()	设置文本绘图模式
选择文本对齐方式	
GUI_GetTextAlign()	返回当前文本对齐模式
GUI_SetLBorder()	设置换行后的左边界
GUI_SetTextAlign()	设置文本对齐模式
设置当前文本坐标	
GUI_GotoX()	设置当前X坐标
GUI_GotoXY()	设置当前X、Y坐标
GUI_GotoY()	设置当前Y坐标
找回当前文本坐标	
GUI_GetDispPosX()	返回当前X坐标
GUI_GetDispPosY()	返回当前Y坐标
清除视窗或其部分的函数	
GUI_Clear()	清除活动视窗（如果背景是活动视窗，则是清除整个屏幕）
GUI_DispatchCEOL()	清除从当前坐标到行末的显示内容

2. 数值显示

函 数	说 明
显示十进制数值	
GUI_DispDec()	在当前坐标显示指定数量字符的十进制数值
GUI_DispDecAt()	在指定坐标显示指定数量字符的十进制数值
GUI_DispDecMin()	在当前坐标显示最少数量字符的十进制数值
GUI_DispDecShift()	在当前坐标显示指定数量字符的长型十进制数值
GUI_DispDecSpace()	在当前坐标显示指定数量字符的十进制数值，用空格代替首位的0
GUI_DispsDec()	在当前坐标显示指定数量字符的十进制数值及显示符号
GUI_DispsDecShift()	在当前坐标显示指定数量字符的长型十进制数值及显示符号
显示浮点数值	
GUI_DispFloat()	在当前坐标显示指定数量字符的浮点数值
GUI_DispFloatFix()	显示浮点数值，指定小数点右边数字数量
GUI_DispFloatMin()	在当前坐标显示最少数量字符的浮点数值
GUI_DispsFloatFix()	显示浮点数值，指定小数点右边数字数量及显示符号
GUI_DispsFloatMin()	在当前坐标显示最少数量字符的浮点数值及显示符号
显示二进制数值	
GUI_DispBin()	在当前坐标显示二进制数值
GUI_DispBinAt()	在指定坐标显示二进制数值
显示十六进制数值	
GUI_DispHex()	在当前坐标显示十六进制数值
GUI_DispHexAt()	在指定坐标显示十六进制数值

3. 2-D 图形库

函 数	说 明
绘图模式	
GUI_SetDrawMode()	设置绘图模式。
基本绘图函数	
GUI_ClearRect()	使用背景颜色填充一个矩形区域。
GUI_DrawPixel()	绘一个单像素点。
GUI_DrawPoint()	绘一个点。
GUI_FillRect()	绘一个填充的矩形。
GUI_InvertRect()	反转一个矩形区域。
绘制位图	
GUI_DrawBitmap()	绘制一幅位图。
GUI_DrawBitmapExp()	绘制一幅位图。
GUI_DrawBitmapMag()	绘制一幅放大的位图。
GUI_DrawStreamedBitmap()	从一个位图数据流的数据绘制一幅位图。
绘线	
GUI_DrawHLine()	绘一根水平线。
GUI_DrawLine()	绘一根线。
GUI_DrawLineRel()	从当前坐标到端点绘一根线，该端点由X轴距离及Y轴距离指定。
GUI_DrawLineTo()	从当前坐标到端点(X, Y)绘一根线。
GUI_DrawPolyLine()	绘折线。
GUI_DrawVLine()	绘一根垂直线。
绘多边形	
GUI_DrawPolygon()	绘一个多边形。
GUI_EnlargePolygon()	对一个多边形进行扩边。
GUI_FillPolygon()	绘一个填充的多边形。

GUI_MagnifyPolygon()	放大一个多边形。
GUI_RotatePolygon()	按指定角度旋转一个多边形。
绘圆	
GUI_DrawCircle()	绘一个圆。
GUI_FillCircle()	绘一个填充的圆。
绘椭圆	
GUI_DrawEllipse()	绘一个椭圆。
GUI_FillEllipse()	绘一个填充的椭圆。
绘圆弧	
GUI_DrawArc()	绘一个圆弧

4. 字体

函 数	说 明
字体的选择	
GUI_GetFont()	返回当前选择字体的指针
GUI_SetFont()	设置当前字体
字体相关函数	
GUI_GetCharDistX()	返回当前字体中指定字符的宽度（X 轴，以像素为单位）
GUI_GetFontDistY()	返回当前字体 Y 轴方向间距
GUI_GetFontInfo()	返回一个包含字体信息的结构
GUI_GetFontSizeY()	返回当前字体的高度（Y 轴，以像素为单位）
GUI_GetStringDistX()	返回一个使用当前字体的文本的 X 轴尺寸
GUI_GetYDistOfFont()	返回一个特殊字体的 Y 轴间距
GUI_GetYSizeOfFont()	返回一个特殊字体的 Y 轴尺寸
GUI_IsInFont()	估计一个指定的字符是否在一种特殊字体里面

5. 颜色

定义	颜色	说明
GUI_BLACK	黑	0x000000
GUI_BLUE	蓝	0xFF0000
GUI_GREEN	绿	0x00FF00
GUI_CYAN	青	0xFFFF00
GUI_RED	红	0x0000FF
GUI_MAGENTA	洋红	0x8B008B
GUI_BROWN	褐	0x2A2AA5
GUI_DARKGRAY	深灰	0x404040
GUI_GRAY	灰	0x808080
GUI_LIGHTGRAY	浅灰	0xD3D3D3
GUI_LIGHTBLUE	淡蓝	0xFF8080
GUI_LIGHTGREEN	淡绿	0x80FF80
GUI_LIGHTCYAN	淡青	0x80FFFF
GUI_LIGHTRED	淡红	0x8080FF
GUI_LIGHTMAGENTA	淡洋红	0xFF80FF
GUI_YELLOW	黄	0x00FFFF
GUI_WHITE	白	0xFFFFFFFF

6. 存储设备

函 数	说 明
基本函数	
GUI_MEMDEV_Create ()	建立存储设备（第一步）
GUI_MEMDEV_CopyToLCD ()	将存储设备的内容拷贝到 LCD
GUI_MEMDEV_Delete ()	存储设备释放使用的存储空间
GUI_MEMDEV_Select ()	选择一个存储设备作为目标用于绘图操作
高级特性	
GUI_MEMDEV_Clear ()	将存储设备的内容标志为未改变 M
GUI_MEMDEV_CopyFromLCD ()	将 LCD 的内容拷贝到存储设备
GUI_MEMDEV_CopyToLCDAA ()	以反锯齿方式拷贝存储设备的内容
GUI_MEMDEV_GetYSize ()	返回存储设备的 Y 轴尺寸
GUI_MEMDEV_ReduceYSize ()	减少存储设备的 Y 轴尺寸
GUI_MEMDEV_SetOrg ()	在 LCD 上改变存储设备的原点
分片存储设备	
GUI_MEMDEV_Draw ()	使用一个存储设备进行绘图
自动设备对象函数	
GUI_MEMDEV_CreateAuto ()	建立一个自动设备对象。
GUI_MEMDEV_DeleteAuto ()	删除一个自动设备对象。
GUI_MEMDEV_DrawAuto ()	使用一个 GUI_AUTODEV 对象进行绘图

7. 视窗管理器

函数	说明
基本函数	
<code>WM_CreateWindow()</code>	创建一个窗口。
<code>WM_CreateWindowAsChild()</code>	创建一个子窗口。
<code>WM_DeleteWindow()</code>	删除一个窗口。
<code>WM_Exec()</code>	通过执行回调函数（所有工作）重绘有效窗口。
<code>WM_Exec1()</code>	通过执行一个回调函数（仅一个工作）重绘有效窗口。
<code>WM_GetClientRect()</code>	返回活动窗口的。
<code>WM_GetDialogItem()</code>	返回一个对话框项目（控件）的窗口句柄。
<code>WM_GetOrgX()</code>	返回活动窗口的原点 X 坐标。
<code>WM_GetOrgY()</code>	返回活动窗口的原点 Y 坐标。
<code>WM_GetWindowOrgX()</code>	返回一个窗口的原点 X 坐标。
<code>WM_GetWindowOrgY()</code>	返回一个窗口的原点 Y 坐标。
<code>WM_GetWindowRect()</code>	返回活动窗口的屏幕坐标。
<code>WM_GetWindowSizeX()</code>	返回一个窗口的水平尺寸（宽度）。
<code>WM_GetWindowSizeY()</code>	返回一个窗口的垂直尺寸（高度）。
<code>WM_HideWindow()</code>	使一个窗口不可见。
<code>WM_InvalidateArea()</code>	使显示屏的某些部分无效。
<code>WM_InvalidateRect()</code>	使一个窗口部分无效。
<code>WM_InvalidateWindow()</code>	使一个窗口无效。
<code>WM_MoveTo()</code>	设置一个窗口的坐标。
<code>WM_MoveWindow()</code>	移动一个窗口到另一个位置。
<code>WM_Paint()</code>	立即绘制或重绘一个窗口。
<code>WM_ResizeWindow()</code>	改变一个窗口尺寸。
<code>WM_SelectWindow()</code>	设置用于绘图操作的活动窗口。
<code>WM_ShowWindow()</code>	使一个窗口可见。
高级特性	
<code>WM_Activate()</code>	激活视窗管理器。
<code>WM_BringToBottom()</code>	在其同胞窗口之后放置一个窗口。
<code>WM_BringToTop()</code>	在其同胞窗口之前放置一个窗口。
<code>WM_ClrHasTrans()</code>	清除 has 透明标识。
<code>WM_Deactivate()</code>	解除视窗管理器。
<code>WM_DefaultProc()</code>	处理信息的默认函数。

WM_GetActiveWindow()	返回活动窗口的句柄。
WM_GetDesktopWindow()	返回桌面窗口的句柄。
WM_GetFirstChild()	返回窗口的第一个子窗口的句柄。
WM_GetNextSibling()	返回窗口的下一相同胞窗口的句柄。
WM_GetHasTrans()	返回 has 透明标志的当前值。
WM_GetParent()	返回窗口的父窗口的句柄。
WM_Init()	初始化视窗管理器。不再需要由 GUI_Init() 来完成。
WM_IsWindow()	判断一个指定的句柄是否一个有效句柄。
WM_SendMessage()	向一个窗口发送信息。
WM_SetDesktopColor()	设置桌面窗口颜色。
WM_SetCallback()	为一个窗口设置回调函数。
WM_SetCreateFlags()	当创建一个新窗口时设置一个默认标识符。
WM_SetHasTrans()	设置 has 透明标志。
WM_SetUserClipRect()	临时减小剪切区域。
WM_ValidateRect()	使一个窗口的部分有效。
WM_ValidateWindow()	使一个窗口有效。
存储设备支持（可选）	
WM_DisableMemdev()	禁止用于重绘一个窗口的存储设备的使用。
WM_EnableMemdev()	启用用于重绘一个窗口的存储设备的使用。

8. 有效控件

名 称	说 明
BUTTON	可以按下的按钮。在按钮上可以显示文本或位图。
CHECKBOX	复选框，提供多个选项。
EDIT	单行文本编辑框，提示用户输入数字或文本。
FRAMEWIN	框架窗口，建立典型的 GUI 外形。
LISTBOX	列表框，当被选中的项目会高亮显示。
PROGBAR	进度条，用于观察。
RADIOBUTTON	单选按钮，可以选择。同一时间只有一个按钮被选中。
SCROLLBAR	滚动条，可以是水平或垂直的。
SLIDER	滑动条，用于改变数值。
TEXT	文本控件，典型应用在对话框中。

函数	说明
WM_DeleteWindow()	删除一个窗口。
WM_DisableMemdev()	禁止存储设备的重绘功能。
WM_EnableMemdev()	启用存储设备的重绘功能。
WM_InvalidateWindow()	使一个窗口无效。
WM_Paint()	立即绘制或重绘一个窗口。

BUTTON API 函数

函数	说明
BUTTON_Create()	建立按钮。
BUTTON_CreateAsChild()	将按钮作为一个子窗口建立。
BUTTON_CreateIndirect()	从资源表项目中建立按钮。
BUTTON_SetBitmap()	设置按钮显示时使用的位图。
BUTTON_SetBitmapEx()	设置按钮显示时使用的位图。
BUTTON_SetBkColor()	设置按钮的背景颜色。
BUTTON_SetFont()	选择文本字体。
BUTTON_SetState()	设置按钮状态（由触摸屏模块自动处理）。
BUTTON_SetStreamedBitmap()	设置按钮显示时使用的位图。
BUTTON_SetText()	设置文本。
BUTTON_SetTextColor()	设置文本的颜色。

复选框 API 函数

函 数	说 明
CHECKBOX_Check()	将复选框设置为选中状态。
CHECKBOX_Create()	创建复选框。
CHECKBOX_CreateIndirect()	从资源表项目中创建复选框。
CHECKBOX_IsChecked()	返回复选框当前的状态（选中或未选中）。
CHECKBOX_Uncheck()	将复选框设置为未选中状态（默认）。

编辑框 API 函数

函 数	说 明
EDIT_AddKey()	键盘输入函数。
EDIT_Create()	创建编辑区。
EDIT_CreateIndirect()	从资源表项目中创建编辑区。
EDIT_GetDefaultFont()	返回默认字体。
EDIT_GetText()	获得用户输入。
EDIT_GetValue()	返回当前数值。
EDIT_SetBinMode()	启用二进制编辑模式。
EDIT_SetBkColor()	设置编辑区的背景颜色。
EDIT_SetDecMode()	启用十进制编辑模式。
EDIT_SetDefaultFont()	设置用于编辑区的默认字体。
EDIT_SetDefaultTextAlign()	设置编辑区默认的文本对齐方式。
EDIT_SetFont()	给文本选择字体。
EDIT_SetHexMode()	启用十六进制模式。
EDIT_SetMaxLen()	设置编辑区的最大字符数量。
EDIT_SetText()	设置文本。
EDIT_SetTextAlign()	设置编辑区文本的对齐方式。
EDIT_SetTextColor()	设置文本的颜色。
EDIT_SetValue()	设置当前数值。
GUI_EditBin()	在当前光标位置编辑一个二进制数值。
GUI_EditDec()	在当前光标位置编辑一个十进制数值。
GUI_EditHex()	在当前光标位置编辑一个十六进制数值。
GUI_EditString()	在当前光标位置编辑一个字符串。

框架窗 API 函数

函数	说明
FRAMEWIN_Create()	创建框架窗口。
FRAMEWIN_CreateAsChild()	创建一个作为一个子窗口的框架窗口控件。
FRAMEWIN_CreateIndirect()	从资源表条目中创建一个框架窗口。
FRAMEWIN_GetDefaultBorderSize()	返回默认边框尺寸。
FRAMEWIN_GetDefaultCaptionSize()	返回标题栏的默认尺寸。
FRAMEWIN_GetDefaultFont()	返回用于框架窗口标题的默认字体。
FRAMEWIN_SetActive()	设置框架窗口的状态。
FRAMEWIN_SetBarColor()	设置标题栏的背景颜色。

FRAMEWIN_SetClientColor()	设置客户区的背景。
FRAMEWIN_SetDefaultBarColor()	设置标题栏默认颜色。
FRAMEWIN_SetDefaultBorderSize()	设置默认边框尺寸。
FRAMEWIN_SetDefaultCaptionSize()	设置标题栏的默认高度。
FRAMEWIN_SetDefaultFont()	设置标题栏的默认字体。
FRAMEWIN_SetFont()	为标题文本选择字体。
FRAMEWIN_SetText()	设置标题文本。
FRAMEWIN_SetTextAlign()	设置标题文本的对齐方式。
FRAMEWIN_SetTextColor()	设置标题文本的颜色。
FRAMEWIN_SetTextPos()	设置标题文本的位置。

列表框 API 函数

函数	说明
LISTBOX_Create()	创建列表框。
LISTBOX_CreateAsChild()	以子窗口的形式追寻列表框。
LISTBOX_CreateIndirect()	从资源表条目追寻列表框。
LISTBOX_DecSel()	减小选择范围。
LISTBOX_GetDefaultFont()	返回列表框的默认字体。
LISTBOX_GetSel()	返回所选择行的数目。
LISTBOX_IncSel()	增加选择范围。
LISTBOX_SetBackColor()	设置背景颜色。
LISTBOX_SetDefaultFont()	修改列表框控件的默认字体。
LISTBOX_SetFont()	选择字体。
LISTBOX_SetSel()	设置选择的行。
LISTBOX_SetTextColor()	设置前景颜色。

进度条 API 函数

函数	说明
PROGBAR_Create()	创建进度条。
PROGBAR_CreateIndirect()	从资源表条目创建进度条。
PROGBAR_SetBarColor()	设置进度条的颜色。
PROGBAR_SetFont()	选择文字使用的字体。
PROGBAR_SetMinMax()	设置进度条的最小和最大数值。
PROGBAR_SetText()	设置（可选择）条棒图形的文字。bargraph.
PROGBAR_SetTextAlign()	设置对齐方式（默认为居中）。
PROGBAR_SetTextColor()	设置文字的颜色。
PROGBAR_SetTextPos()	设置坐标（默认为 0, 0）。
PROGBAR_SetValue()	设置条棒图形的数值（如果没有文字则显示百分比）。

单选按钮 API 函数

函数	说明
RADIO_Create()	建立一组单选按钮。
RADIO_CreateIndirect()	从资源表条目创建一组单选按钮。
RADIO_Dec()	将按钮选择序号减 1。
RADIO_GetValue()	返回当前选择的按钮。
RADIO_Inc()	将按钮选择序号加 1。
RADIO_SetValue()	设置选择的按钮。

滚动条 API 函数

函数	说明
SCROLLBAR_AddValue()	滚动条的数值增加或减少一个指定的值。
SCROLLBAR_Create()	创建滚动条。
SCROLLBAR_CreateAttached()	创建一个吸附到一个窗口的滚动条。
SCROLLBAR_CreateIndirect()	从资源表条目中创建滚动条。
SCROLLBAR_Dec()	滚动条数值减 1。
SCROLLBAR_GetValue()	返回当前条目的数值。
SCROLLBAR_Inc()	滚动条数值加 1。
SCROLLBAR_SetNumItems()	设置滚动条目的值。
SCROLLBAR_SetPageSize()	设置页尺寸（按条目的数量）。
SCROLLBAR_SetValue()	设置滚动条当前数值。
SCROLLBAR_SetWidth()	设置滚动条的宽度。

滑动条 API 函数

函 数	说 明
SLIDER_Create()	创建滑动条。
SLIDER_CreateIndirect()	从资源表条目创建滑动条。
SLIDER_Dec()	滑动条数值减 1。
SLIDER_GetValue()	返回滑动条当前数值。
SLIDER_Inc()	滑动条数值增 1。
SLIDER_SetRange()	设置滑动条数值范围。
SLIDER_SetValue()	设置滑动条当前数值。
SLIDER_SetWidth()	设置滑动条的宽度。

文本 API 函数

函数	说明
TEXT_Create()	创建文本。
TEXT_CreateIndirect()	从资源表条目中创建文本。
TEXT_GetDefaultFont()	返回用于文本的默认字体。
TEXT_SetDefaultFont()	设置用于文本的默认字体。
TEXT_SetFont()	设置用于一个指定的文本控件的字体。
TEXT_SetText()	设置一个指定的文本控件的文本。
TEXT_SetTextAlign()	设置一个指定的文本控件的文本对齐方式。
TEXT_SetTextPos()	设置在一个指定的文本控件当中文本的位置。

9. 抗锯齿 API 函数

控制函数	说 明
GUI_AA_DisableHiRes()	禁止高分辨率座标。
GUI_AA_EnableHiRes()	启用高分辨率座标。
GUI_AA_GetFactor()	返回当前的抗锯齿系数。
GUI_AA_SetFactor()	设置当前的抗锯齿系数。

绘图函数	说 明
GUI_AA_DrawArc()	绘一段抗锯齿圆弧。
GUI_AA_DrawLine()	绘一根抗锯齿直线。
GUI_AA_DrawPolyOutline()	绘一个抗锯齿多边形的轮廓。
GUI_AA_FillCircle()	绘一个抗锯齿的圆。
GUI_AA_FillPolygon()	绘一个填充和抗锯齿的多边形。

10. 输入设备

指针光标输入设备 API 函数

函 数	说 明
GUI_PID_GetState()	返回 PID 的当前状态。
GUI_PID_StoreState()	存储 PID 的当前状态。

通用鼠标 API 函数

函 数	说 明
GUI_MOUSE_GetState()	返回鼠标的当前状态。
GUI_MOUSE_StoreState()	存储鼠标的当前状态。

触摸屏 API 函数

函 数	说 明
GUI_TOUCH_GetState()	返回触摸屏的当前状态。
GUI_TOUCH_StoreState()	存储触摸屏的当前状态。

函 数	说 明
GUI_TOUCH_Calibrate()	更改刻度。
GUI_TOUCH_Exec()	激活 X 轴和 Y 轴的测量;需要大约每秒 100 次的调用。
GUI_TOUCH_SetDefaultCalibration()	恢复默认刻度。

函 数	说 明
TOUCH_X_ActivateX()	准备 Y 轴的测量。
TOUCH_X_ActivateY()	准备 X 轴的测量。
TOUCH_X_MeasureX()	返回 A/D 转换器 X 轴的结果。
TOUCH_X_MeasureY()	返回 A/D 转换器 Y 轴的结果。

11. 时间相关函数

函 数	说 明
GUI_Delay()	延时一个指定时段。
GUI_Exec()	运行回调函数（所有的工作）。
GUI_Exec1()	运行一个回调函数（仅一项工作）。
GUI_GetTime()	返回当前的系统时间。

12. GUI_X 函数参考

函 数	说 明
初始化函数	
GUI_X_Init()	从 GUI_Init() 调用，能用于初始化硬件。
时间函数	
GUI_X_Delay()	在一个指定的时间段后返回。
GUI_X_ExecIdle()	只从视窗管理器的非阻塞函数调用。
GUI_X_GetTime()	返回当前系统时间，以毫秒为单位。
内核接口函数	
GUI_X_InitOS()	初始化内核接口模块（建立一个资源旗语/互斥）。
GUI_X_GetTaskId()	返回一个当前任务/线程唯一的 32 位标识符。
GUI_X_Lock()	锁上 GUI（阻塞资源旗语/互斥）。
GUI_X_Unlock()	解锁 GUI（解除资源旗语/互斥阻塞）。
GUI_X_Log()	返回调试信息。如果启用记录的话，这是必需的。

13. LCD 驱动 API 函数

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 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。