



欣瑞达 LUA 脚本 API

深圳市欣瑞达液晶显示技术有限公司



欣瑞达液晶



400-069-8808 0755-26018666



广东省深圳市南山区高新园北四道16号庆邦电子大厦8栋4层



www.xrd-lcd.com



使用淘宝扫一扫
欣瑞达官方旗舰店



使用微信扫一扫
欣瑞达官方微信

1. 适用范围

文档仅适合新系列串口屏产品，S 系列

2. LUA 脚本介绍

LUA 脚本初学者可以通过下面链接进行学习。

<http://www.runoob.com/lua/lua-arrays.html>

3. API 接口函数

3.1 控件属性类

3.1.1 change_screen(screen)

切换到指定画面

screen: 目标画面 ID

3.1.2 set_value(screen,control,value)

设置控件数值

按钮控件: value -0 按下, 1 弹起

文本控件: value -整数或小数

也可以设置进度条、滑块、仪表等

3.1.3 get_value(screen,control)

获取控件数值, 按钮、文本、进度条、滑块、仪表等

3.1.4 set_visiable(screen,control,visiable)

设置控件是否可见, visiable 为 0 隐藏, 1 显示

3.1.5 set_enable(screen,control,enable)

设置控件是否可触摸，enable 为 0 禁止触摸，1 启用触摸

3.1.6 set_fore_color(screen,control,color)

设置控件前景色，color 为 RGB565
例如文本控件文字颜色，进度条显示颜色。

3.1.7 set_back_color(screen,control,color)

设置控件背景色，color 为 RGB565
例如文本控件背景颜色，进度条背景颜色。

3.1.8 set_text(screen,control,text)

设置控件显示内容(字符串)，文本控件，二维码控件等

3.1.9 get_text(screen,control)

获取控件字符串内容(字符串)，文本控件，二维码控件等

3.2 常用回调函数

3.2.1 on_init()

系统加载 LUA 脚本文件之后，立即调用此回调函数，通常用于执行初始化操作。

3.2.2 on_systick()

系统每隔 1 秒钟自动调用此回调函数。

3.2.3 on_control_notify(screen,control,value)

用户触摸修改控件后，执行此回调函数。
点击按钮控件，修改文本控件、修改滑动条都会触发此事件。
value-为数值类型，如果需要获取文本控件的字符串值，使用
get_text(screen, control)。

3.2.4 on_screen_change(screen)

当画面需要切换时，执行此回调函数，screen 切换到的画面。
注意，此函数内部调用 change_screen, 不会嵌套执行 on_screen_change。

3.2.5 on_press(state,x,y)

用户点击触摸屏时，执行此回调函数。
state-0 松开，1 按下，2 持续按压
x, y-为触摸坐标

3.2.6 on_usb_inserted(driver)

U 盘插入时，执行此回调函数，dirver 为 U 盘的盘符

3.2.7 on_usb_removed()

U 盘拔出时，执行此回调函数

3.3 绘图函数

3.3.1 on_draw(screen)

重绘时执行此回调函数，通常所有绘图操作都在此函数中实现。

3.3.2 redraw()

发送重绘请求，触发 on_draw 的执行。

3.3.3 set_pen_color(color)

设置画笔的颜色，RGB565，用于指定线、矩形、圆等的颜色。

3.3.4 draw_line(x0,y0,x1,y1,width)

绘制直线

x0, y0 起始点坐标

x1, y1 结束点坐标

width 为线条的厚度，1~10

3.3.5 draw_rect(x0,y0,x1,y1,fill)

绘制矩形

x0, y0 左上角坐标

x1, y1 右下角坐标

fill 为 0 不填充，1 填充

3.3.6 draw_circle(x,y,r,fill)

绘制圆形

x, y 圆的中心坐标

r 圆的半径

fill 为 0 不填充，1 填充

3.3.7 draw_ellipse(x0,y0,x1,y1,fill)

绘制椭圆

x0, y0 左上角坐标

x1, y1 右下角坐标

fill 为 0 不填充，1 填充

3.3.8 draw_image(image_id,frame_id,dstx,dsty, width,height,srcx,srcy)

绘制图片

image_id 图片资源的 ID

frame_id 对应图标，可以设置帧 ID，其他图片固定为 0

dstx 图片显示 X 坐标

dsty 图片显示 Y 坐标

width 图片显示宽度

height 图片显示高度

srcx 图片裁剪 X 坐标

srcy 图片裁剪 Y 坐标

3.3.9 draw_image_file(filename,dstx,dsty,width, height,srcx,srcy)

绘制图片，此方法不对图片进行缓存，效率较低

filename 图片文件，支持 JPEG/PNG

dstx 图片显示 X 坐标

dsty 图片显示 Y 坐标

width 图片显示宽度

height 图片显示高度

srcx 图片裁剪 X 坐标

srcy 图片裁剪 Y 坐标

3.3.10 load_surface (filename)

加载图片到图层

filename 图片文件，支持 JPEG/PNG

例如：surface = load_surface (“c:/test.jpg”)

图层不再使用时，需要调用 destroy_surface 进行销毁，否则会导致内存泄漏。

3.3.11 destroy_surface (surface)

销毁图层

surface 图层资源指针

3.3.12 draw_surface (surface,dstx,dsty,width,height,srcx,srcy)

绘制图层，相比于 draw_image_file，此方法效率较高

Surface 图层资源指针

dstx 图片显示 X 坐标

dsty 图片显示 Y 坐标

width 图片显示宽度[可选]

height 图片显示高度[可选]

srcx 图片裁剪 X 坐标[可选]

srcy 图片裁剪 Y 坐标[可选]

例如：

平铺显示 draw_surface(surface, dstx, dsty)

缩放显示 draw_surface(surface, dstx, dsty, width, height)

裁剪显示 draw_surface(surface, dstx, dsty, width, height ,srcx, srcy)

3.3.13 draw_text(text,x,y,w,h,font,color,align)

显示文字

text 字符串

x 显示 X 坐标

y 显示 Y 坐标

w 显示宽度

h 显示高度

font 字体编号

color 颜色 RGB565

align 对齐方式

bit0~bit1 水平对齐方式，0 左对齐，1 居中对齐，2 右对齐

bit2~bit3 垂直对齐方式，0 上对齐，1 居中对齐，3 下对齐

3.4 寄存器访问

LUA 中访问 MODBUS/PLC 等协议中定义的变量，需要通过下面的变量访问接口

3.4.1 get_variant(name)

获取协议变量的数值，get_variant(“Variable1”)

3.4.2 set_variant(name,value)

设置协议变量的数值，set_variant(“Variable1”,12345)

3.5 定时器

3.5.1 start_timer(timer_id, timeout, countdown, repeat)

启动定时器

timer_id-定时器 ID, 0~31

timeout-超时时间, 单位毫秒

countdown-0 顺计时, 1 倒计时

repeat-重复次数, 0 表示无限重复

3.5.2 stop_timer(timer_id)

停止定时器

3.5.3 on_timer(timer_id)

定时器超时回调函数, timer_id 超时的定时器号

3.5.4 get_timer_value(timer_id)

获取定时器当前计时时间

3.6 单串口

3.6.1 uart_send_data(packet)

通过串口发送数据

3.6.2 uart_set_timeout(timeout, timeout_inter)

设置串口接收超时时间

timeout-接收总超时

timeout_inter-字节间隔超时

3.6.3 uart_set_baudrate(baudrate)

设置波特率

3.6.4 uart_get_baudrate()

获取波特率

3.6.5 on_uart_recv_data(packet)

串口接收数据的回调函数，有两种方式可以触发此函数执行：

使用自定义串口指令：格式为 EE B5 【自定义数据】 FF FC FF FF

使用自由串口协议：在 LUA 脚本中定义全局变量 `uart_free_protocol = 1`

3.7 多路通信

3.7.1 dev_send_data(port, packet)

通过通信设备接口发送数据（port：设备逻辑接口号，设备接口类型可为串口、CAN 总线、网络端口等）

3.7.2 dev_set_timeout(port, timeout, timeout_inter)

设置端口接收超时时间

timeout-接收总超时

timeout_inter-字节间隔超时

3.7.3 dev_set_bitrate(port, bitrate)

设置设备通信速率

3.7.4 dev_get_bitrate(port)

获取设备通信速率

3.7.5 on_dev_recv_data(port, packet)

端口接收数据的回调函数，有两种方式可以触发此函数执行：

使用自定义串口指令：格式为 EE B5 【自定义数据】 FF FC FF FF

使用自由串口协议：在 LUA 脚本中定义全局变量 `uart_free_protocol = 1`

3.8 音视频

3.8.1 play_sound(filename)

播放指定的声音文件，例如 `play_sound('a:/sounds/welcome.wav')`

3.8.2 on_audio_callback (state)

声音播放结束回调通知，state 保留未使用。

3.8.3 set_volume(level)

设置音量 0~100

3.8.4 get_volume()

获取音量

3.9 其他

3.9.1 set_backlight(level)

设置背光亮度 0~100

3.9.2 get_backlight()

获取背光亮度

```
level = get_backlight
```

3.9.3 beep(time)

蜂鸣器叫，单位毫秒

4. 声明与服务

感谢您选用欣瑞达系列产品，若您对文档有什么异议或疑问，欢迎随时与我们取得联系。

电话：0755-26018666， 网址：www.xrd-lcd.com。当然若文档有什么错误或误解之处，欢迎给我们提出批评和建议，我们将及时纠正和改进。