Technical Note

物联型 LUA 脚本 API

工程技术笔记

类别	内容
关键词	LUA 脚本
摘要	物联型串口屏提供的 LUA 脚本 API 接口函数

修订历史

版本	日期	原因	编制
V1.0	2017/11/29	创建文档	刘仁武
V1.1	2018/09/14	增加 HTTP 下载、音视频播放通知等	刘仁武
V1.2	2019/06/19	增加记录控件 API	刘仁武
V1.3	2019/08/06	增加 MODBUS API 接口	刘仁武

销售与服务

广州大彩光电科技有限公司

电话：020-82186683
传真：020-82187676
Email：hmi@gz-dc.com（公共服务）
网站：www.gz-dc.com
地址：广州高新技术产业开发区玉树工业园富康西街 8 号 C 栋 303 房
官网零售淘宝店：https://gz-dc.taobao.com

目录

1. 适用范围

文档仅适合新物联型系列串口屏产品，W 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 get_current_screen()

获取当前画面 ID

3.1.3 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

3.1.4 get_value(screen,control)

获取控件数值，按钮、文本、进度条、滑块、仪表等

3.1.5 set_visiable(screen,control,visiable)

设置控件是否可见，visiable 为 0 隐藏，1 显示

3.1.6 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

3.1.7 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。

3.1.8 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。

3.1.9 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

3.1.10 set_text_roll(screen,control, speed)

设置文本控件滚动速度，每秒多少个像素。设置为 0 停止滚动。

3.1.11 set_text_flicker (screen,control, cycle)

设置文本控件闪烁周期，单位秒。设置为 0 停止闪烁。

3.1.12 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

3.1.13 set_options (screen,control, options)

设置选择控件的内容，例如：set_options (screen,control, ‘选项 1;选项 2;选项 3;’)

3.2 记录控件

3.2.1 record_set_event(screen,control,eventid)

告警类型-触发告警

3.2.2 record_reset_event(screen,control,eventid)

告警类型-解除告警

3.2.3 record_add(screen,control,record)

在末尾添加一条记录，record 为字符串，例如 “item1;item2;item3;”

3.2.4 record_insert (screen,control,position,record)

在指定位置插入一条记录

3.2.5 record_clear(screen,control)

清除记录数据

3.2.6 record_setoffset(screen,control,offset)

设置滚动显示位置

3.2.7 record_get_count(screen,control)

获取记录条数

3.2.8 record_read(screen,control)

读取一条记录，字符串

3.2.9 record_modify(screen,control,record)

修改一条记录

3.2.10 record_delete(screen,control,position)

删除一条记录

3.2.11 record_select(screen,control,position)

选中一条记录

3.2.12 record_export(screen,control)

导出记录到 SD 卡/U 盘

3.3 常用回调函数

3.3.1 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。

3.3.2 on_systick()

系统每隔 1 秒钟自动调用此回调函数。

3.3.3 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)。

3.3.4 on_screen_change(screen)

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

3.3.5 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

3.3.6 on_usb_inserted(driver)

U 盘插入时，执行此回调函数，dirver 为 U 盘的盘符

3.3.7 on_usb_removed()

U 盘拔出时，执行此回调函数

3.4 绘图函数

3.4.1 on_draw(screen)

当界面的显示内容需要更新时，系统自动调用此函数，用户在此函数中添加自定义的绘图操作。用户绘制的内容叠加在画面内容之上。
此函数为系统回调函数，用户不要直接调用。
下面几种情况会触发此函数：

- 界面有动画播放、视频播放、RTC 时间显示的动态刷新；
- 用户操作屏幕控件控件；
- 通过 LUA 脚本或串口指令更新控件；
- 通过执行 redraw；

总之，界面上有任何变化，都会触发此回调函数。

3.4.2 redraw()

发送重绘请求，触发 on_draw 的执行。

3.4.3 screen_shoot(filename,x,y,width,height,quality)

截取屏幕窗口范围，存储到指定文件路径
filename: 图片文件存放路径
quality: JPEG 图片质量，默认 95
例如：screen_shoot('b:/shoot.jpg',0,0,480,272, 95)

3.4.4 set_pen_color(color)

设置画笔的颜色，RGB565，用于指定线、矩形、圆等的颜色。

3.4.5 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

3.4.6 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

3.4.7 draw_rect_alpha(x0,y0,x1,y1,alpha)

绘制实心的半透明矩形，F 系列不支持
x0,y0 左上角坐标
x1,y1 右下角坐标
alpha 透明度 0 全透明~255 不透明

3.4.8 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

3.4.9 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

3.4.10 load_image (image_id,frame_id)

加载指定图片到内存，一般用在 on_init 中，牺牲了开机速度，但使运行过程更流程。
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

3.4.11 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

3.4.12 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

3.4.13 load_surface (filename)

加载图片到图层，F 系列不支持
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。

3.4.14 destroy_surface (surface)

销毁图层，F 系列不支持
surface 图层资源指针

3.4.15 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，F 系列不支持
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)

3.4.16 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

3.5 MODBUS 协议访问

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。
mb_前缀的接口，固件版本要求：W 系列>=590，F 系列>=349

3.5.1 get_variant(name)

获取协议变量的数值，get_variant(“Variable1”)

3.5.2 set_variant(name,value)

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

3.5.3 mb_set_timeout (timeout)

设置从机应答超时时间，范围 1~255，10 毫秒单位

3.5.4 mb_read_coil_01 (slave,addr,quantity)

01 功能码，读取线圈，成功时返回字节数组，一个字节 8 个 coil，失败返回 nil

3.5.5 mb_read_input_02(slave,addr,quantity)

02 功能码，读取离散输入，成功时返回字节数组，一个字节 8 个 input，失败返回 nil

3.5.6 mb_read_reg_03(slave,addr,quantity)

03 功能码，读取保持寄存器，成功时返回 WORD 数组，失败返回 nil

3.5.7 mb_read_input_reg_04(slave,addr,quantity)

04 功能码，读取输入寄存器，成功时返回 WORD 数组，失败返回 nil

3.5.8 mb_write_coil_05 (slave,addr,status)

05 功能码，写单个线圈，成功时返回 true，失败返回 false
写入 status 1 为 ON，0 为 OFF

3.5.9 mb_write_reg_06 (slave,addr,reg)

06 功能码，写入单个保持寄存器，成功时返回 true，失败返回 false
reg 为寄存器值

3.5.10 mb_write_coil_15 (slave,addr,quantity,coils)

15 功能码，写多个线圈，成功时返回 true，失败返回 false
coils 为字节数组，一个字节 8 个 coil

3.5.11 mb_write_reg_16 (slave,addr,regs)

16 功能码，写入多个保持寄存器，成功时返回 true，失败返回 false
regs 为寄存器 WORD 数组

3.6 网络相关

F 系列不支持

3.6.1 get_wifi_cfg()

返回 4 个参数

wifi_mode, secumode, ssid, password = get_wifi_cfg()
wifi_mode 无线网络模式 0-禁用无线网络，1-无线网卡模式，2-AP 热点模式
secumode 加密模式 0-AUTO(默认值) 1-WEP 2-WPAPSK 3-WPAPSK2
ssid 无线网络名称
password 无线网络密码

3.6.2 set_wifi_cfg(wifi_mode, secumode, ssid, password)

参数说明同上

3.6.3 get_network_state()

state = get_network_state()
状态位说明
bit0-无线网络连接
bit1-有线网络连接
bit2-是否连上服务器
bit3-是否有客户端连上

3.6.4 set_network_cfg(dhcp, ipaddr, netmask, gateway, dns)

dhcp-启用 DHCP，0 禁用 1 启用，禁用时后面的参数才有效
ipaddr-静态 IP
netmask-掩码
gateway-子网掩码
dns-域名服务器

3.6.5 get_network_cfg()

返回五个参数，说明同上
dhcp, ipaddr, netmask, gateway, dns = get_network_cfg()

3.6.6 save_network_cfg()

保存网络设置，并重连网络

3.6.7 set_network_service_cfg(wificom, mode, port, server_addr)

设置网络服务参数
wificom -默认为 0，为 1 时启用透传模式（即无线串口屏）
mode -0 禁用网络服务，1 客户端模式，2 服务器模式
port -服务端口，默认 5050
server_addr -服务器地址，(屏作客户端时)

3.6.8 get_network_service_cfg()

返回 4 个参数，说明同上
wificom, mode, port, server_addr = get_network_service_cfg()

3.6.9 scan_ap()

扫描无线热点，返回热点数目
ap_count = scan_ap()

3.6.10 get_ap_info(index)

获取指定热点的信息
ssid, security, quality = get_ap_info(index)
index 热点索引
ssid 热点名称
security 加密方式
quality 信号质量

3.6.11 client_send_data(packet)

通过客户端 SOCKET 发送报文
local packet = {} –定义数组
packet[0] = 0x01
packet[1] = 0x02
…
client_send_data(packet)

3.6.12 server_send_data(packet)

通过服务端 SOCKET 发送报文

3.6.13 on_client_recv_data(packet)

当客户端 SOCKET 接收到数据时，系统自动回调此函数。

```
function on_client_recv_data(packet)
    --打印消息
    print('on_client_recv_data:')
    for i=0,#(packet) do
      print(packet[i])
    end

    --处理消息，这里简单回送数据
    client_send_data(packet)

    --返回1时，消息不通过串口发送给用户MCU
    return 1
end
```

3.6.14 on_server_recv_data(packet)

当服务端 SOCKET 接收到数据时，系统自动回调此函数。
处理方法与 on_client_recv_data 类似。

3.6.15 http_request(taskid,uri,method,content_type,postdata)

发送 HTTP 请求到服务器
taskid：请求任务编号，任意设置
uri：资源路径
method：方法，0GET，1POST
以下参数 POST 方法才需要
content_type：数据类型例如 json,xml,text 等

postdata：POST 数据

3.6.16 on_http_response(taskid,response)

HTTP 响应
taskid：响应任务编号，与 http_request 匹配
response：响应数据

3.6.17 http_download (taskid,uri,savepath)

使用 HTTP 协议下载文件
taskid：请求任务编号，任意设置
uri：资源路径
savepath：存放位置

3.6.18 on_http_download (taskid, status)

下载响应
taskid：响应任务编号，与 http_download 匹配
status：下载状态：0 下载失败，1 下载成功但存储失败，2 下载并存储成功

3.6.19 udp_create(port)

创建 UDP 套接字，并绑定服务端口
例如：sockfd = udp_create(12345)

3.6.20 udp_close(sockfd)

关闭 UDP 套接字

3.6.21 udp_recvfrom(sockfd)

接收 UDP 数据报文
ret,ip,port,packet = udp_recvfrom(sockfd)
ret=-1 表示发生错误，ret=0 表示无数据，其他值表示数据长度
ip,port 发送端的 IP 和端口
packet 为数据报文，table 类型

3.6.22 upd_sendto(sockfd,ip,port,packet)

发送 UDP 数据报文

3.7 定时器

3.7.1 start_timer(timer_id, timeout, countdown, repeat)

启动定时器，超时后系统自动调用 on_timer
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

3.7.2 stop_timer(timer_id)

停止定时器

3.7.3 on_timer(timer_id)

定时器超时回调函数

3.7.4 get_timer_value(timer_id)

获取定时器当前计时时间，单位毫秒

3.8 串口

3.8.1 uart_send_data(packet)

通过串口发送数据

3.8.2 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

3.8.3 uart_set_baudrate(baudrate)

设置波特率

3.8.4 uart_get_baudrate()

获取波特率

3.8.5 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

```
function on_uart_recv_data(packet)
    --打印消息
    print('on_uart_recv_data:')
    for i=0,#(packet) do
      print(packet[i])
    end
end
```

3.9 音视频

3.9.1 play_sound(filename)

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

3.9.2 stop_sound()

停止声音播放

3.9.3 on_audio_callback (state)

声音播放结束回调通知，state 保留未使用。

3.9.4 set_volume(level)

设置音量 0~100

3.9.5 get_volume()

获取音量

3.9.6 play_video(pathname,repeat)

播放视频，pathname 为视频路径，repeat 为重复次数，F 系列不支持

3.9.7 pause_video()

暂停视频播放，F 系列不支持

3.9.8 resume_video()

恢复视频播放，F 系列不支持

3.9.9 stop_video()

停止视频播放，F 系列不支持

3.10 FLASH 存储器读写

屏幕提供 128K 用户 FLASH，可用于存储配置参数。
固件版本要求：W 系列>=590，F 系列>=349

3.10.1 write_flash(addr,data)

写用户 FLASH 数据，addr 写入地址，data 字节数组。

3.10.2 read_flash(addr,length)

读用户 FLASH 数据，addr 写入地址，length 读取字节数
data = read_flash(addr,length)

3.10.3 write_flash_string(addr,str)

写字符串到指定 FLASH 地址

3.10.4 read_flash_string(addr)

从指定 FLASH 地址读取字符串
str = read_flash_string(addr)

3.10.5 flush_flash()

系统会对 FLASH 写入操作进行缓存优化，以提高写入效率。
flush_flash 操作会立即把数据写入 FLASH。

3.11 其他

3.11.1 set_backlight(level)

设置背光亮度 0~100

3.11.2 get_backlight()

level = get_backlight

3.11.3 set_language(lang)

设置当前语言选项

3.11.4 get_language ()

获取当前语言选项

3.11.5 set_wakeup_mode (mode)

唤醒模式设置，可组合设置
0x1 单击唤醒，0x2 双击唤醒，0x4 串口唤醒

3.11.6 sleepmode (on)

进入睡眠低功耗模式，睡眠后屏幕功能不再运行，只能使用触摸唤醒。

3.11.7 standbymode (on)

进入待机低功耗模式，屏幕的串口可正常运行。F 系列不支持。

3.11.8 beep(time)

蜂鸣器叫，单位毫秒

3.11.9 get_tick_count (time)

获取上电以后运行时间，单位毫秒。
32 位计数器，大约 49 天后溢出归零重新计时。

3.11.10 get_date_time ()

获取当前日期时间
year,mon,day,hour,min,sec,week = get_date_time()

3.11.11 set_date_time (time)

设置当前日期时间
set_date_time(year,mon,day,hour,min,sec)

3.11.12 upgrade_logo (url)

通过 U 盘更新开机 LOGO，F 系列不支持

```
function on_usb_inserted(driver)
    upgrade_logo(driver..'logo.jpeg')
end
```

3.11.13 gpio_set_in (pin)

PIN 引脚设置为输入模式，F 系列不支持。

3.11.14 gpio_set_out (pin)

PIN 引脚设置为输出模式，F 系列不支持。

3.11.15 gpio_set_value (pin,value)

设置输出 PIN 引脚为（高电平 1/低电平 0），F 系列不支持。

3.11.16 gpio_get_value (pin)

获取输入 PIN 引脚电平（高电平 1/低电平 0），F 系列不支持。

4. 声明与服务

感谢您选用大彩系列产品，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：020-82186683-601， Emial: hmi@gz-dc.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。