

物联型 LUA 脚本 API

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	审查
V1.0	2017/11/29	创建文档	刘仁武	李勇
V1.1	2018/09/14	增加 HTTP 下载、音视频播放通知等	刘仁武	李勇



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.....	6
2. LUA 脚本介绍	7
3. API 接口函数	8
3.1 控件属性类.....	8
3.1.1 change_screen(screen)	8
3.1.2 set_value(screen,control,value)	8
3.1.3 get_value(screen,control)	8
3.1.4 set_visiable(screen,control,visiable)	8
3.1.5 set_enable(screen,control,enable)	8
3.1.6 set_fore_color(screen,control,color).....	8
3.1.7 set_back_color(screen,control,color)	8
3.1.8 set_text(screen,control,text)	8
3.1.9 get_text(screen,control).....	8
3.2 常用回调函数.....	8
3.2.1 on_init().....	8
3.2.2 on_systick()	8
3.2.3 on_control_notify(screen,control,value)	8
3.2.4 on_screen_change(screen).....	9
3.2.5 on_press(state,x,y).....	9
3.2.6 on_usb_inserted(driver)	9
3.2.7 on_usb_removed().....	9
3.3 绘图函数.....	9
3.3.1 on_draw(screen)	9
3.3.2 redraw()	9
3.3.3 set_pen_color(color).....	9
3.3.4 draw_line(x0,y0,x1,y1,width)	9
3.3.5 draw_rect(x0,y0,x1,y1,fill).....	9
3.3.6 draw_circle(x,y,r,fill)	9
3.3.7 draw_ellipse(x0,y0,x1,y1,fill)	10
3.3.8 draw_image(image_id,frame_id,dstx,dsty,width,height,srcx,srcy)	10
3.3.9 draw_image_file(filename,dstx,dsty,width,height,srcx,srcy)	10
3.3.10 load_surface (filename).....	10
3.3.11 destroy_surface (surface)	10
3.3.12 draw_surface (surface,dstx,dsty,width,height,srcx,srcy)	10
3.3.13 draw_text(text,x,y,w,h,font,color,align).....	11
3.4 寄存器访问.....	11
3.4.1 get_variant(name).....	11
3.4.2 set_variant(name,value)	11
3.5 网络相关.....	11
3.5.1 get_wifi_cfg().....	11
3.5.2 set_wifi_cfg(wifi_mode, secumode, ssid, password).....	11
3.5.3 get_network_state()	11

3.5.4	set_network_cfg(dhcp, ipaddr, netmask, gateway, dns)	12
3.5.5	get_network_cfg()	12
3.5.6	save_network_cfg()	12
3.5.7	set_network_service_cfg(wificom, mode, port, server_addr)	12
3.5.8	get_network_service_cfg()	12
3.5.9	scan_ap()	12
3.5.10	get_ap_info(index)	12
3.5.11	client_send_data(packet)	12
3.5.12	server_send_data(packet)	13
3.5.13	on_client_recv_data(packet)	13
3.5.14	on_server_recv_data(packet)	13
3.5.15	http_request(taskid,uri,method,content_type,postdata)	13
3.5.16	on_http_response(taskid,response)	13
3.5.17	http_download (taskid,uri,savepath)	13
3.5.18	on_http_download (taskid, status)	14
3.6	定时器	14
3.6.1	start_timer(timer_id, timeout, countdown, repeat)	14
3.6.2	stop_timer(timer_id)	14
3.6.3	on_timer(timer_id)	14
3.6.4	get_timer_value(timer_id)	14
3.7	串口	14
3.7.1	uart_send_data(packet)	14
3.7.2	uart_set_timeout(timeout, timeout_inter)	14
3.7.3	uart_set_baudrate(baudrate)	14
3.7.4	uart_get_baudrate()	14
3.7.5	on_uart_recv_data(packet)	14
3.8	音视频	15
3.8.1	play_sound(filename)	15
3.8.2	on_audio_callback (state)	15
3.8.3	set_volume(level)	15
3.8.4	get_volume()	15
3.9	其他	15
3.9.1	set_backlight(level)	15
3.9.2	get_backlight()	15
3.9.3	beep(time)	15
3.10	机智云接口	15
3.10.1	gagent_get_info()	15
3.10.2	gagent_send_data(packet)	16
3.10.3	gagent_reset()	16
3.10.4	gagent_get_bind_url()	16
3.10.5	gagent_get_status()	16
3.10.6	on_gagent_recv_data(packet)	16
4.	声明与服务	18



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 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 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`。

3.2.5 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，driver 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 网络相关

3.5.1 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

3.5.2 set_wifi_cfg(wifi_mode, secumode, ssid, password)

参数说明同上

3.5.3 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

3.5.4 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

3.5.5 get_network_cfg()

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

3.5.6 save_network_cfg()

保存网络设置, 并重连网络

3.5.7 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)

3.5.8 get_network_service_cfg()

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

3.5.9 scan_ap()

扫描无线热点, 返回热点数目
ap_count = scan_ap()

3.5.10 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

3.5.11 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)
```

3.5.12 server_send_data(packet)

通过服务端 SOCKET 发送报文

3.5.13 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
```

3.5.14 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。

3.5.15 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

3.5.16 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

3.5.17 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

3.5.18 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 定时器

3.6.1 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

3.6.2 stop_timer(timer_id)

停止定时器

3.6.3 on_timer(timer_id)

定时器超时回调函数

3.6.4 get_timer_value(timer_id)

获取定时器当前计时时间

3.7 串口

3.7.1 uart_send_data(packet)

通过串口发送数据

3.7.2 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

3.7.3 uart_set_baudrate(baudrate)

设置波特率

3.7.4 uart_get_baudrate()

获取波特率

3.7.5 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
end
```

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()

level = get_backlight

3.9.3 beep(time)

蜂鸣器叫，单位毫秒

3.10 机智云接口

3.10.1 gagent_get_info()

系统自动调用此函数，获取机智云配置信息

```
--获取机智云的配置参数
--一般只需要修改产品密钥
function gagent_get_info()
    product_key = '1cc8b199a4c14a5f957e03346b9962a'
    protocol_ver = '00000001'
    p0_ver = '00000002'
    hard_ver = '00000003'
    soft_ver = '00000004'
    return product_key,protocol_ver,p0_ver,hard_ver,soft_ver
end
```

3.10.2 gagent_send_data(packet)

发送数据到机智云

3.10.3 gagent_reset()

复位机智云设备绑定信息

3.10.4 gagent_get_bind_url()

获取机智云绑定链接，可用于二维码控件扫描绑定

bind_url = gagent_get_bind_url()

3.10.5 gagent_get_status()

获取机智云连接状态

status = gagent_get_status()

3.10.6 on_gagent_recv_data(packet)

机智云数据接收回调函数


```
--当接收到机智云发送的消息时，
--系统自动调用此函数，packet为消息字节数组
--返回1时，机智云的消息不发给用户MCU
function on_gagent_rcv_data(packet)

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

    --回复请求
    action = packet[0]

    --WIFI模块控制设备
    if action==0x01
    then
        gagent_wifi_ctrl_mcu(packet) --处理控制命令
        update_cloud_ui()
    end

    --WIFI模块读取设备状态
    if action==0x02
    then
        gagent_send_status(3)
    end

    return 1
end
```

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， Email: 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。

