Technical Note

Was linked LUA script API

Engineering notes

category	content
Key words LUA Script Summary Was linked to the serial screen provides LUA script API Interface	
functions	

revise history

version	date	the reason	prepared by	Examine
V1.0	2017/11/29	Create documents	Liu Renwu	Li Yong
V1.1	2018/09/14	increase HTTP Download, audio and video playback notification	Liu Renwu	Li Yong

Sales and Service

Guangzhou color Optoelectronics Technology Co., Ltd.

phone: 020-82186683

fax: 020-82187676

Email : hmi@gz-dc.com (Public Service) website: www.gz-dc.com

Address: Guangzhou High-tech Industrial Development Zone, Yushu Industrial Park, Beverly West 8 number C Building

303 Housing official website Taobao retail shop: https://gz-dc.taobao.com

table of Contents

1. Scope

Documentation only fit the new screen was linked series of serial products, W series.

2. LUA Introduction to Scripting

LUA Script beginners can learn through the link below.

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

3. API Interface functions

3.1 Control Properties class

3.1.1 change_screen (screen)

Switch to the specified screen

screen : Target screen ID

3.1.2 set_value (screen, control, value)

Setting numerical control button controls: value -0 Press, 1 Pop-up

text controls: value - Integer or decimal may be provided

progress bars, sliders, gauges, etc.

3.1.3 get_value (screen, control)

Gets the control value, buttons, text, progress bars, sliders, meters, etc.

3.1.4 set_visiable (screen, control, visiable)

Whether to set control is visible, visiable for 0 hide, 1 display

3.1.5 set_enable (screen, control, enable)

Whether to set controls can touch, enable for 0 Do not touch, 1 Touch-enabled

3.1.6 set_fore_color (screen, control, color)

Set the foreground color controls, color for RGB565

For example text control text color, progress bar display color.

3.1.7 set_back_color (screen, control, color)

Set the control background color, color for RGB565

Such as text controls the background color, the background color of the progress bar.

3.1.8 set_text (screen, control, text)

Control displays setting contents (character string), text control, and other two-dimensional code control

3.1.9 get_text (screen, control)

Gets the contents of a string (string), text control, two-dimensional code control, etc.

3.2 Common callback function

3.2.1 on_init ()

System load LUA After the script file, immediately call this callback function is typically used to perform initialization.

3.2.2 on_systick ()

Every system 1 Second automatic call this callback function.

3.2.3 on_control_notify (screen, control, value)

After modifying the user touches the controls, the implementation of this callback function.

Click the button control, modify text control, modify the slider will trigger this event.

value- Numeric type, if the need to obtain the control value of the text string, using get_text (screen, control) .

3.2.4 on_screen_change (screen)

When a need to switch the screen, execute callback function, screen For the target screen. Note that this internal function call change_screen, Not nested execution on_screen_change .

3.2.5 on_press (state, x, y)

When the user clicks on the touch screen, the execution of this callback function.

state-0 release, 1 Press, 2 Continuously pressed

x, y- Touch coordinates

3.2.6 on_usb_inserted (driver)

U When the disk is inserted, the implementation of this callback function, dirver for U Disk drive letter

3.2.7 on_usb_removed ()

U Pull out the disk, do this callback function

3.3 Drawing functions

3.3.1 on_draw (screen)

This callback function performs redrawing, drawing operations are all normally implemented in this function.

3.3.2 redraw ()

Send redraw request, triggering on_draw Execution.

3.3.3 set_pen_color (color)

Set the color brush, RGB565 For designating lines, rectangles, circles, etc. colors.

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

Draw a straight line

x0, y0 Starting point

x1, y1 End point coordinates

width Line thickness, 1 ~ 10

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

Draw a rectangle

x0, y0 The coordinates of the upper left corner

x1, y1 The lower right corner coordinates

fill for 0 Not filled, 1 filling

3.3.6 draw_circle (x, y, r, fill)

Draw a circle

x, y Coordinates of the center of the circle

r Radius of the circle

fill for 0 Not filled, 1 filling

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

Draw an Oval

x0, y0 The coordinates of the upper left corner

x1, y1 The lower right corner coordinates

fill for 0 Not filled, 1 filling

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

Draw pictures

image_id Image resources ID frame_id The corresponding icon, the frame may be provided ID Other

pictures fixed 0 dstx Image display X coordinate

dsty Image display Y coordinate

width Picture display width

height Picture display height

srcx Image cropping X coordinate

srcy Image cropping Y coordinate

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

Draw pictures, this method does not picture cache, less efficient

filename Image files, support JPEG / PNG dstx Image

display X coordinate

dsty Image display Y coordinate

width Picture display width

height Picture display height

srcx Image cropping X coordinate

srcy Image cropping Y coordinate

3.3.10 load_surface (filename)

Load picture to Layers

filename Image files, support JPEG / PNG

E.g: surface = load_surface ("c: /test.jpg")

When the layer is no longer used, you need to call destroy_surface Destruction, otherwise it will lead to memory leaks.

3.3.11 destroy_surface (surface)

Destruction Layer

surface Layers resource pointer

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

Draw layers, compared to draw_image_file High efficiency of this method

Surface Layers resource pointer

dstx Image display X coordinate

dsty Image display Y coordinate

width Image display width [optional]

height Pictures show the height [optional] srcx Image cropping X Coordinate [Optional]

srcy Image cropping Y Coordinates [Optional] For example: tile draw_surface

(surface, dstx, dsty)

Zoom display draw_surface (surface, dstx, dsty, width, height)

Crop display draw_surface (surface, dstx, dsty, width, height, srcx, srcy)

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

Display text

text String

x display X coordinate

y display Y coordinate

w Display width

h Display height

font Font Number

color colour RGB565 align Alignment

bit0 ~ bit1 Horizontal alignment, 0 Left, 1 Align Center, 2 Align Right

bit2 ~ bit3 Vertical alignment, 0 The alignment, 1 Align Center, 3 Align Bottom

3.4 Register Access

LUA Access MODBUS / PLC Variables defined in the protocol, the interface need to access the following variables

3.4.1 get_variant (name)

Gets the value of the variable protocol, get_variant ("Variable1")

3.4.2 set_variant (name, value)

Set the value of the variable protocol, set_variant ("Variable1", 12345)

3.5 Network-related

3.5.1 get_wifi_cfg ()

return 4 Parameters

wifi_mode, secumode, ssid, password = get_wifi_cfg () wifi_mode Wi-Fi mode 0- Disable the wireless network, 1- Wireless LAN mode, 2-AP Hot model

secumode Encryption mode 0-AUTO (Defaults) 1-WEP 2-WPAPSK 3-WPAPSK2 ssid Wireless Network Name

password Wireless network password

3.5.2 set_wifi_cfg (wifi_mode, secumode, ssid, password)

Parameter Description above

3.5.3 get_network_state ()

state = get_network_state ()

Status bits

bit0- Wi-Fi

bit1- Wired network connection

bit2- Whether connected to the server

bit3- Is there a client connected to the

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

dhcp- Enable DHCP , 0 Disable 1 Enabled, disabled behind the argument is valid

ipaddr- Static state IP

netmask- Mask

gateway- Subnet Mask

dns- Domain name server

3.5.5 get_network_cfg ()

Return five parameters described above

dhcp, ipaddr, netmask, gateway, dns = get_network_cfg ()

3.5.6 save_network_cfg ()

Save the network settings, and then reconnect to the network

3.5.7 set_network_service_cfg (wificom, mode, port, server_addr)

Set the network service parameters

wificom - The default is 0 ,for 1 When you enable transparent mode (ie, wireless serial screen)

mode -0 Disable network services, 1 Client mode, 2 Server mode

port - Service port, default 5050 server_addr - Server address (when

the screen as a client)

3.5.8 get_network_service_cfg ()

return 4 Argument, explained above

wificom, mode, port, server_addr = get_network_service_cfg ()

3.5.9 scan_ap ()

Scanning for wireless hotspot, returns the number of hotspots

ap_count = scan_ap ()

3.5.10 get_ap_info (index)

Access to information designated hotspots

ssid, security, quality = get_ap_info (index) index Hot

Index

ssid Hotspot name

security Encryption

quality Signal quality

3.5.11 client_send_data (packet)

By client SOCKET Send message

local packet = {} - Definition array

packet [0] = 0x01

packet [1] = 0x02 ...

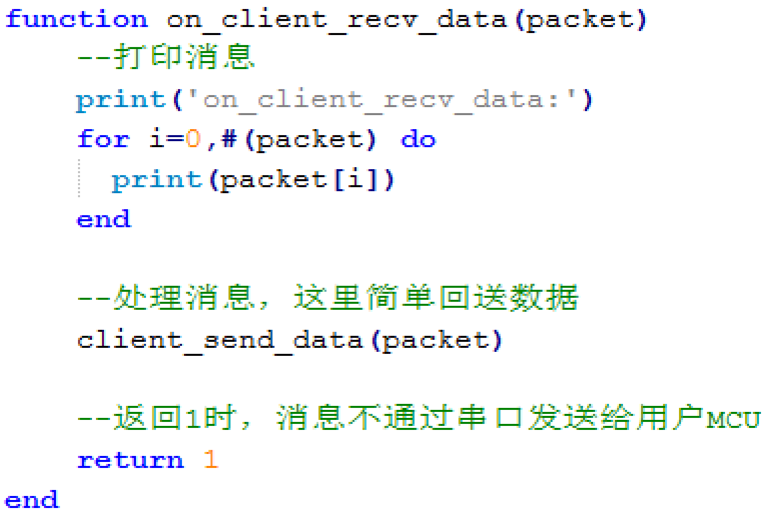
client_send_data (packet)

3.5.12 server_send_data (packet)

By the server SOCKET Send message

3.5.13 on_client_recv_data (packet)

When a client SOCKET Upon receiving the data, the system automatically call back function.



```
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)

When the server SOCKET Upon receiving the data, the system automatically call back function.

Treatment and on_client_recv_data similar.

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

send HTTP Requests to the server

taskid : Request task number, any set

uri : Resource Path

method :method, 0GET , 1POST

The following parameters POST The method requires only

content_type : Data type e.g. json, xml, text Wait

postdata : POST data

3.5.16 on_http_response (taskid, response)

HTTP response

taskid : Response task number, and http_request match

response : Response Data

3.5.17 http_download (taskid, uri, savepath)

use HTTP Download files

taskid : Request task number, any set

uri : Resource Path

savepath :Storage location

3.5.18 on_http_download (taskid, status)

Download the response

taskid : Response task number, and http_download match

status : Download Status: 0 download failed, 1 Download success but failure to store, 2 Download and store success

3.6 Timer

3.6.1 start_timer (timer_id, timeout, countdown, repeat)

Start timer

timer_id- Timer ID , 0 ~ 31 timeout- Timeout

milliseconds

countdown-0 Of counting, 1 Countdown

repeat- repeat times, 0 It indicates an infinite repetition

3.6.2 stop_timer (timer_id)

Stop the timer

3.6.3 on_timer (timer_id)

Timer timeout callback function

3.6.4 get_timer_value (timer_id)

Current time acquisition timer counting

3.7 Serial ports

3.7.1 uart_send_data (packet)

Serial data transmission

3.7.2 uart_set_timeout (timeout, timeout_inter)

Set serial receive timeout

timeout- Received Total time out

timeout_inter- Byte Interval Timeout

3.7.3 uart_set_baudrate (baudrate)

Set the baud rate

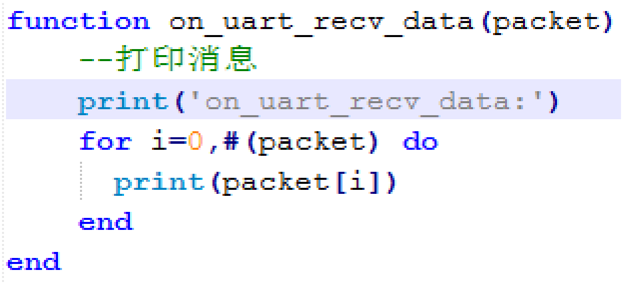
3.7.4 uart_get_baudrate ()

Gets baud rate

3.7.5 on_uart_recv_data (packet)

The callback function receives serial data, there are two ways to trigger the execution of this function:

- Use custom serial command: format EE B5 [Custom Data] FF FC FF FF
- Use the free serial protocol: in LUA Global variables defined in the script 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
```

3.8 Audio and video

3.8.1 play_sound (filename)

Plays the specified sound file, for example, play_sound ('a: /sounds/welcome.wav')

3.8.2 on_audio_callback (state)

Sound playback end notification callback, state Retention is not used.

3.8.3 set_volume (level)

Set the volume 0-100

3.8.4 get_volume ()

Get volume

3.9 other

3.9.1 set_backlight (level)

Setting the backlight brightness 0-100

3.9.2 get_backlight ()

level = get_backlight

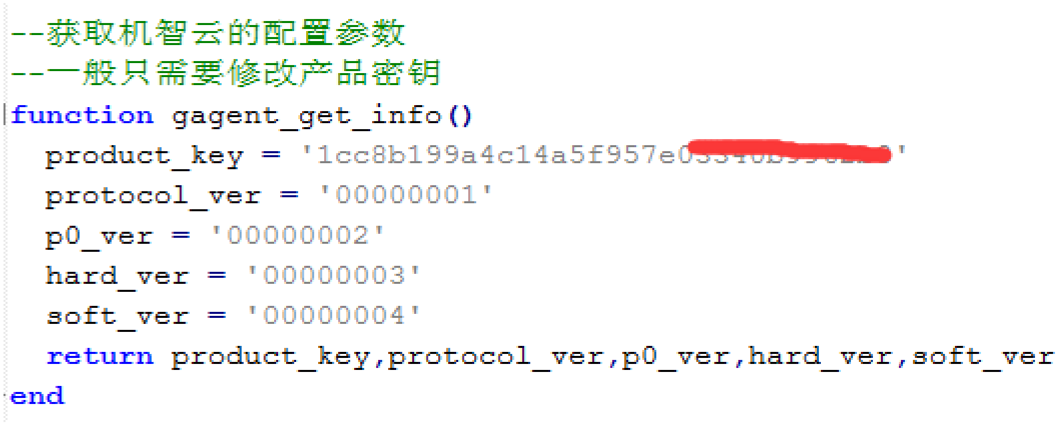
3.9.3 beep (time)

Buzzer called, in milliseconds

3.10 Wit cloud Interface

3.10.1 gagent_get_info ()

The system automatically calls this function, get witty cloud configuration information



```
--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 = '1cc8b199a4c14a5f957e0[redacted]'
  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)

Send data to the cloud wit

3.10.3 gagent_reset ()

Reset wit cloud device binding information

3.10.4 gagent_get_bind_url ()

Get wit cloud binding link can be used to control two-dimensional code scanning bind

bind_url = gagent_get_bind_url ()

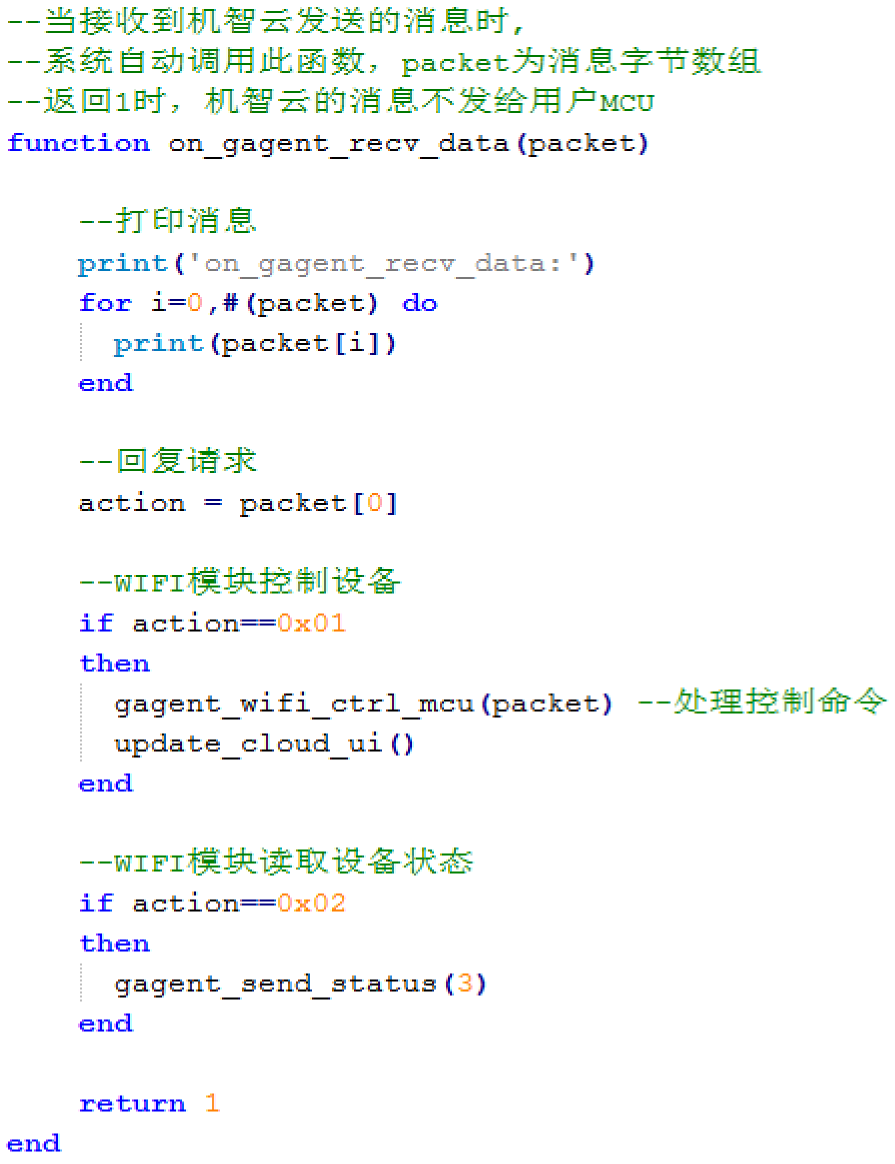
3.10.5 gagent_get_status ()

A connection state acquisition cloud wit

status = gagent_get_status ()

3.10.6 on_gagent_recv_data (packet)

Cloud data received callback function wit



```
--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_recv_data(packet)

    --打印消息
    print('on_gagent_recv_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. Notices and Services

Thank you for choosing large color products, if you have any objections or questions about documentation, please feel free to get in touch with us. phone: 020-82186683-601 , Emial: hmi@gz-dc.com . Of course, if the document has any error or misunderstanding, welcome comments and suggestions to us, we will promptly correct and improve.