

LUA 例程-控件读写 API 函数说明 V1.0

类别	内容
关键词	LUA 控件 API
摘要	



修订历史

版本	日期	原因	编制	审查
V1.0	2019/02/28	创建文档	林青田	



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: www.gz-dc.taobao.com

目录

1. 适合范围.....	1
2. 开发环境版本.....	2
3. 概述.....	3
4. 参考文档.....	4
5. 教程实现.....	5
5.1 准备工程素材.....	5
5.1.1 硬件平台.....	5
5.1.2 UI素材准备.....	5
5.1.3 LUA编辑器.....	6
5.2 API函数说明.....	7
5.3 教程实现.....	9
5.3.1 读写控件值.....	9
5.3.2 显示隐藏控件.....	17
5.3.3 禁止或者使能触摸控件.....	19
5.3.4 控件前景色.....	21
5.3.5 控件背景色.....	22
5.4 下载工程.....	24
5.4.1 操作过程.....	24
6. 完整程序清单.....	26
7. 免责声明.....	34

1. 适合范围

该文档适合所有大彩物联型系列。

2. 开发环境版本

1. VisualTFT 软件版本: V3.0.0.944 及以上的版本。

版本查看:

a) 打开 VisualTFT 软件启动页面如图 2-1 软件版本, 右上角会显示的软件版本号;



图 2-1 软件版本

b) 打开 VisualTFT, 在软件右下角可以查看软件版本图 2-2 软件版本, 最新版本可登录 <http://www.gz-dc.com/> 进行下载。

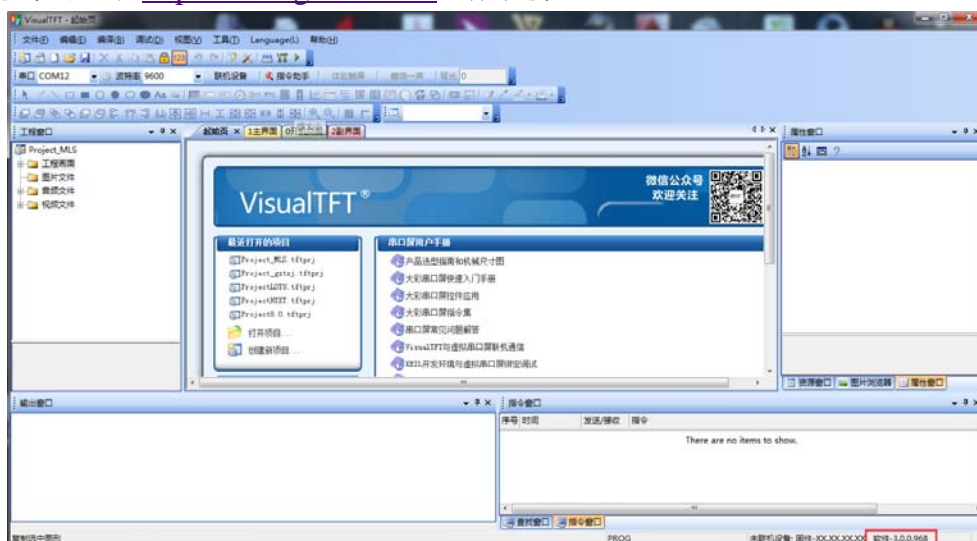


图 2-2 软件版本

2. 串口屏硬件版本: V3.0.287.0 及以上的版本。

版本查看:

- a) 查看屏幕背面版本号贴纸;
- b) VisualTFT 与屏幕联机成功后, 右下角显示的版本号。

3. LUA 语言版本 V5.5

3. 概述

物联型串口屏可以通过 LUA 脚本配合工程完成丰富多样的操作。

本文将通过教程 DEMO 介绍大彩 LUA 脚本 API 函数中的关于控件 API 函数使用方法。

4. 参考文档

1. 《大彩组态串口屏 LUA 脚本 API》 可通过以下链接下载物联型开发包获取:
<http://www.gz-dc.com/index.php?s=/List/index/cid/19.html>
2. 《LUA 基础学习》 可通过以下链接下载物联型开发包获取:
<http://www.gz-dc.com/index.php?s=/List/index/cid/19.html>
3. 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>

5. 教程实现

本章节主要通过 LUA 教程 demo 讲述如何基础运算和字符处理函数的使用以及编写程序的注意事项。本文将分为以下是 3 个阶段讲述教程 DEMO 是如何实现的：

1. 准备工程素材，
2. 实现功能，
3. 下载工程。

5.1 准备工程素材

在实现教程 DEMO 前需要作以下三个准备：

1. 硬件平台，
2. UI 素材，
3. LUA 编辑器。

5.1.1 硬件平台

该例程使用大彩物联型 7 寸串口屏 DC80480W070 为验证开发平台。如图 5-1 所示

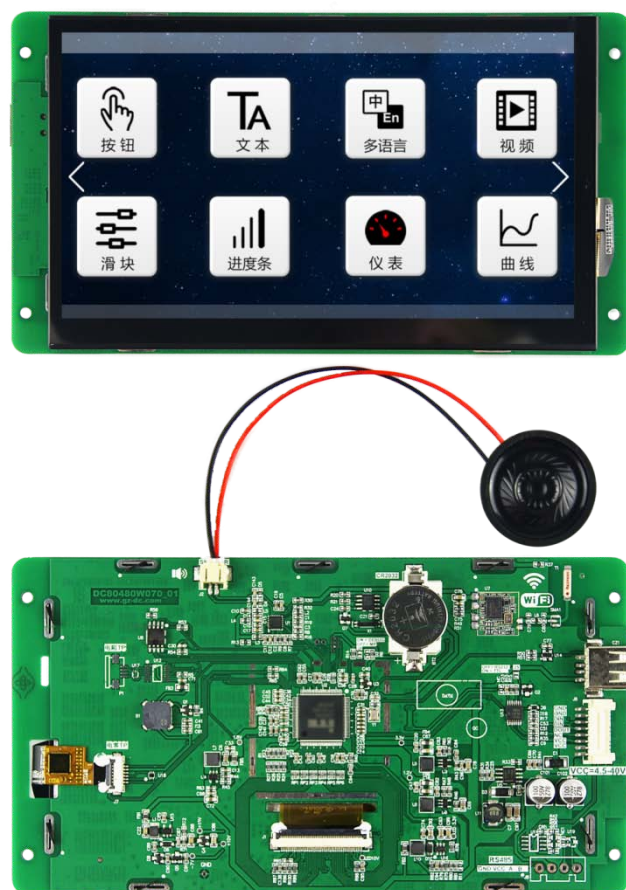


图 5-1 物联型 7 寸串口屏

其他尺寸的物联型串口屏均可借鉴此教程。

5.1.2 UI 素材准备

准备好相应的美工 UI。如图 5-2 所示

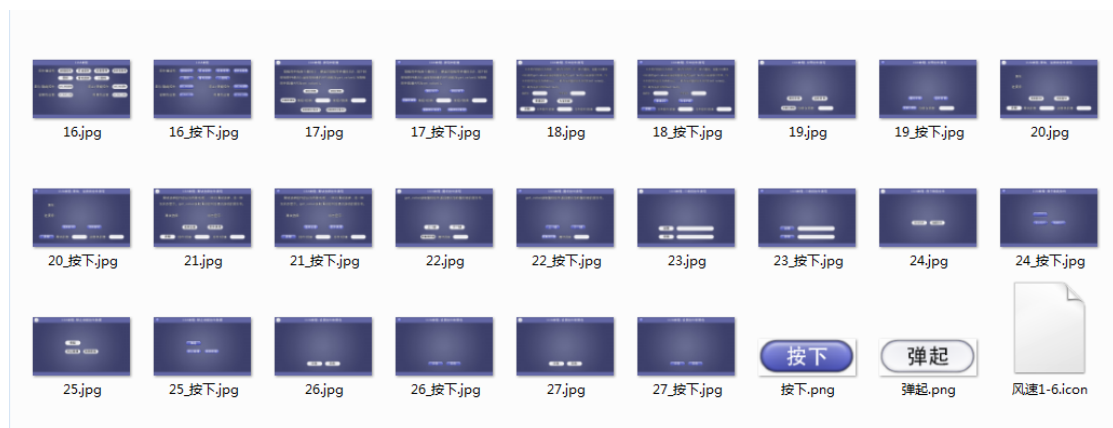


图 5-2 素材准备

5.1.3 LUA 编辑器

上位机 VisualTFT 内部已集成了 LUA 开发编译环境，点击菜单栏工具，选择 LUA 编辑器，如错误！未找到引用源。所示。

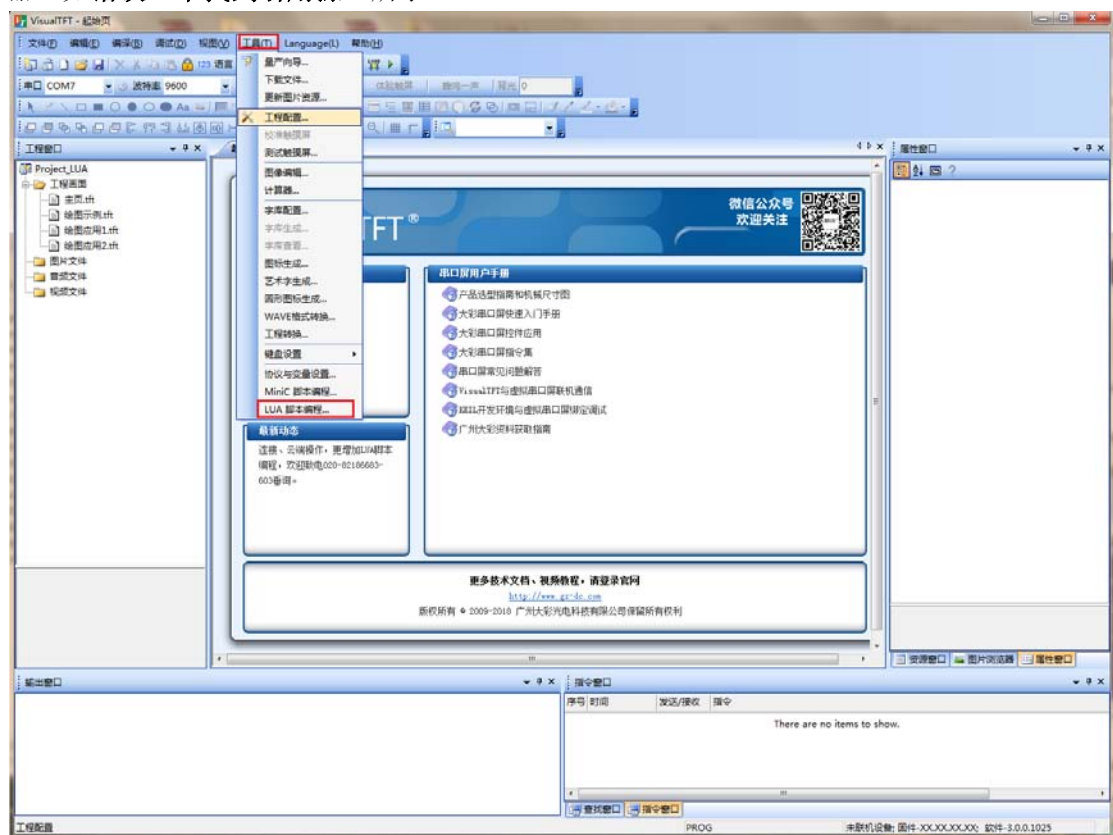


图 5-3 打开 LUA 编辑器

打开编辑器后，Visual TFT 画面如图 5-4 所示；

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

图 5-5 API 函数文档

本教程文档中所涉及到的部分的相关接口函数如下。

1. 函数 on_control_notify(screen,control,value)

注释：此函数可以在用户触摸控件后被回调。函数有三个参数：

screen 表示画面 ID

control 表示控件的编号

value 表示控件的值。

2. 函数 get_value(screen,control)

注释：获取控件值，返回的是数值

3. 函数 get_text(screen,control)

注释：获取控件的字符串内容，文本控件，二维码控件

4. 函数 set_value(screen,control,value)

注释：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

5. 函数 `set_text(screen, control, str)`
注释: 设置显示内容, 如文本控件、二维码控件
6. 函数 `set_visiable(screen, control, visiable)`
注释: 设置控件隐藏或显示
visiable 为 0 隐藏, 1 显示
7. 函数 `set_enable(screen, control, enable)`
注释: 设置触摸使能
参数: enable 为 0 禁止触摸, 1 启用触摸
8. 函数 `set_fore_color(screen, control, color)`
注释: 设置控件前景色
9. 函数 `set_back_color(screen, control, color)`
注释: 设置控件背景色

5.3 教程实现

本章节主要讲述如何设置和读取控件值的方法, 教程中每个函数的功能实现步骤如下:

1. 配置工程属性,
2. 编写程序以及编译,
3. 运行程序。

5.3.1 读写控件值

1. 读按钮值和写按钮值
 - a) 按钮控件只有 0 和 1 两个值, 按钮弹起时值为 0, 按下时值为 1。使用函数 `get_value` 可以获取到某个按钮的状态, 使用 `set_value` 则可以设置按钮的状态。教程 DEMO 中对两个按钮进行读写演示, 画面如图 5-6 所示;

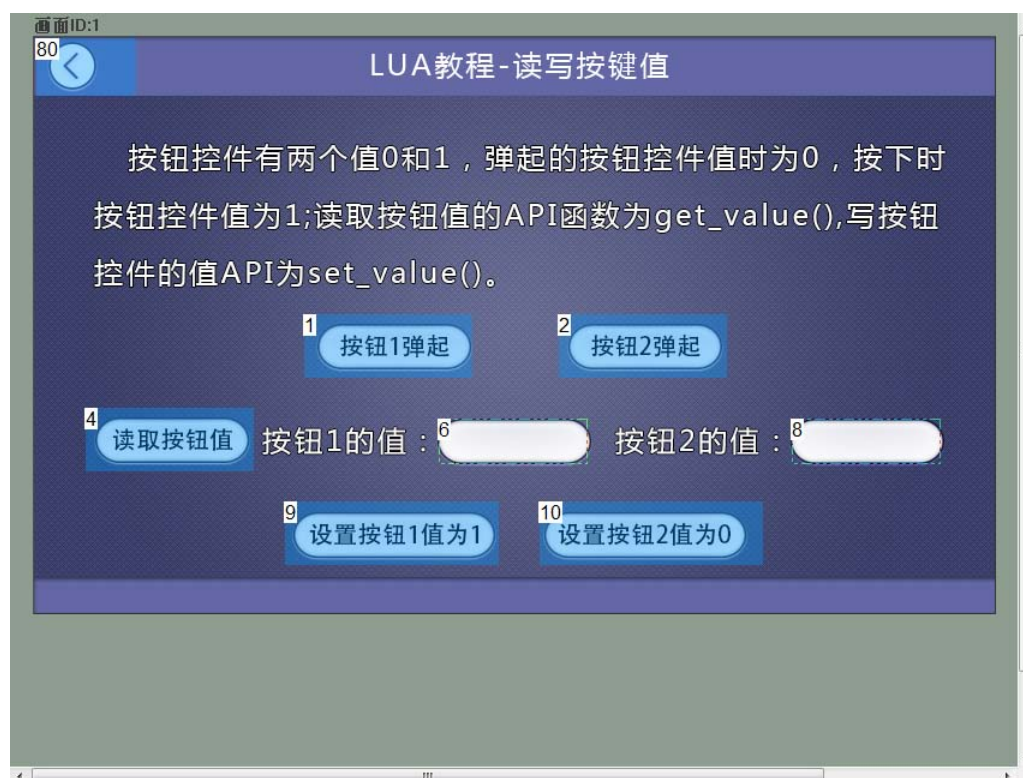


图 5-6 按钮控件读写

- b) 教程演示说明：按下控件 4，读取按钮控件 1 和按钮控件 2 的值，并在对应的文本框中显示出来；按下按钮 9 和按钮 10 分别设置按钮 1 的值为 1，按钮 2 的值为 0。按钮控件读写显示的程序如程序清单 1 中所示。

程序清单 1 按钮控件值读写

```
--*****
--功能：按钮控件画面读写按钮控件值
--调用函数：set_value(screen,control,value) 设置控件数值（也可以设置进度条、滑块、仪表等）
--参数：          screen 画面 ID
--          control 控件 ID
--          value 控件值(0 or 1)

--调用函数：get_value(screen,control) 获取控件的数值
--*****

if screen == 1 then
    if control == 4 and value == 1 then          --按下按钮 4 获取按钮 1，2 的值
        local contorl1_value = get_value(1,1)
        local contorl2_value = get_value(1,2)
        set_value(1,6,contorl1_value)
        set_value(1,8,contorl2_value)
    end
    if control == 9 and value == 1 then          --按下按钮 9 设置按钮 1 的值为 1
        set_value(1,1,1)
    end
    if control == 10 and value == 1 then         --按下按钮 10 设置按钮 2 的值为 0
        set_value(1,2,0)
    end
end
end
```

2. 文本控件值的读写

- a) 文本控件的值分为两种，一种为数值，一种为字符串。根据开发诉求可以直接是用 `get_value` 直接获取数值或者是使用 `get_text` 获取字符串，使用 `set_value` 可以在文本控件中显示数值或者使用 `set_txet` 在文本控件内容。教程演示文本控件读写的画面如图 5-7 所示

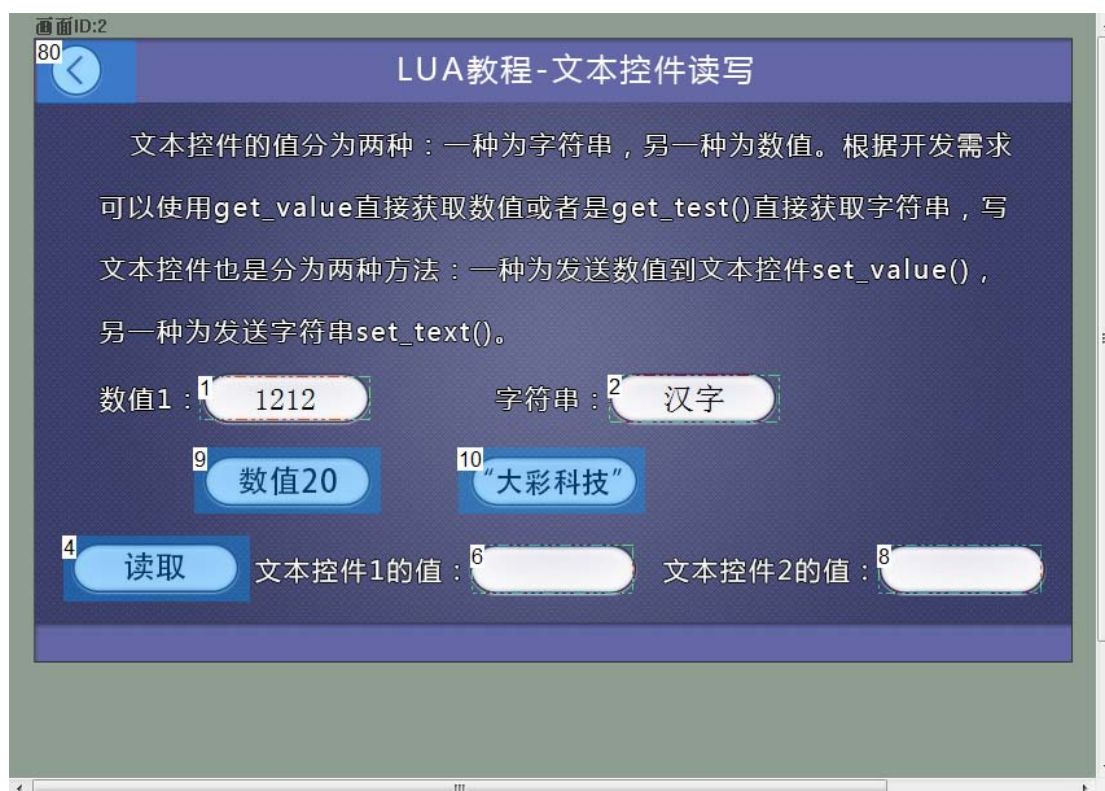


图 5-7 文本控件

- b) 教程说明：点击读取按钮可以将文本控件 1 和文件 2 的值读取出来，点击按钮控件 9 使用函数 `set_value` 将文本控件 1 显示数值 20，点击按钮 10 使用函数 `set_text` 将字符串在文本控件 2 中显示出来。读写文本控件的程序如程序清单 2 所示；

程序清单 2 文本控件

```
if screen == 2 then
    if control == 4 and value == 1 then                --按下按钮 4 读取文本控件 1，2 的内容
        local text1_value = get_value(2,1)
        local text2_value = get_text(2,2)
        set_value(2,6,text1_value)
        set_text(2,8,text2_value)
    end
    if control == 9 and value == 1 then                --按下按钮 9 设置文本控件 1 的内容为数值 20
        set_value(2,1,20)
    end
    if control == 10 and value == 1 then               --按下按钮 10 设置文本控件 2 的内容为大彩科技
        set_text(2,2,"大彩科技")
    end
end
end
.....
```

3. 仪表控件读写

- a) 仪表控件的值为数值，所以读取仪表控件值是使用函数 `get_value`，设置仪表的值则使用函数 `set_value`。教程演示画面如图 5-8 所示；

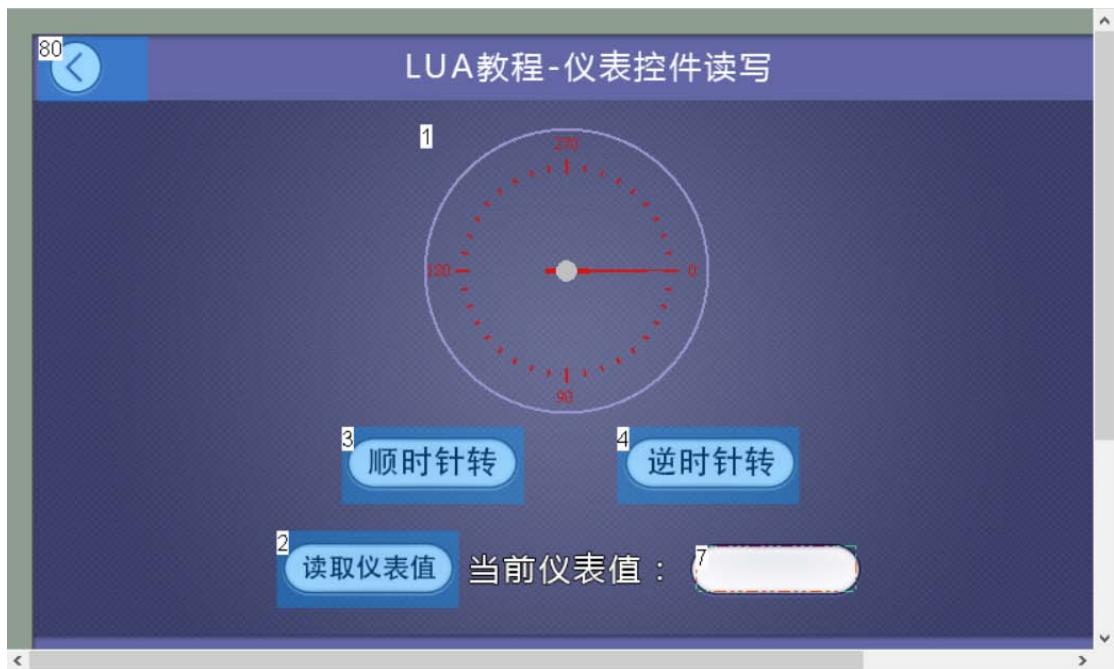


图 5-8 仪表控件

b) 教程说明：当按下按钮 3，仪表顺时针转动；按下按钮 4 仪表逆时针转动，按下按钮 2 可以读取当前仪表控件的值。读写仪表控件的值如程序清单 3 所示

程序清单 3 仪表控件

```
--*****
--功能：仪表控件画面读写控件值
--调用函数：set_value(screen,control,value)
--调用函数：get_value(screen,control)
--参数：    screen 画面 ID
--          control 控件 ID
--          value  控件值
--*****

if screen == 3 then
    if control == 3 and value == 1 then
        meter_value = meter_value + 5
        if meter_value >= 360 then
            meter_value = 0
        end
        set_value(3,1,meter_value)
    end
    if control == 4 and value == 1 then
        meter_value = meter_value - 5
        if meter_value < 0 then
            meter_value = 355
        end
        set_value(3,1,meter_value)
    end
end
```



```

end
if control == 2 and value == 1 then
    local value = get_value(3,1)
    set_value(3,7,value)
end
end
end

```

4. 进度条、滑块控件读写

a) 进度条、滑块控件的值为数值，读取控件值时使用函数 `get_value`，设置控件的值则使用函数 `set_value`。教程演示画面如图 5-9 所示；

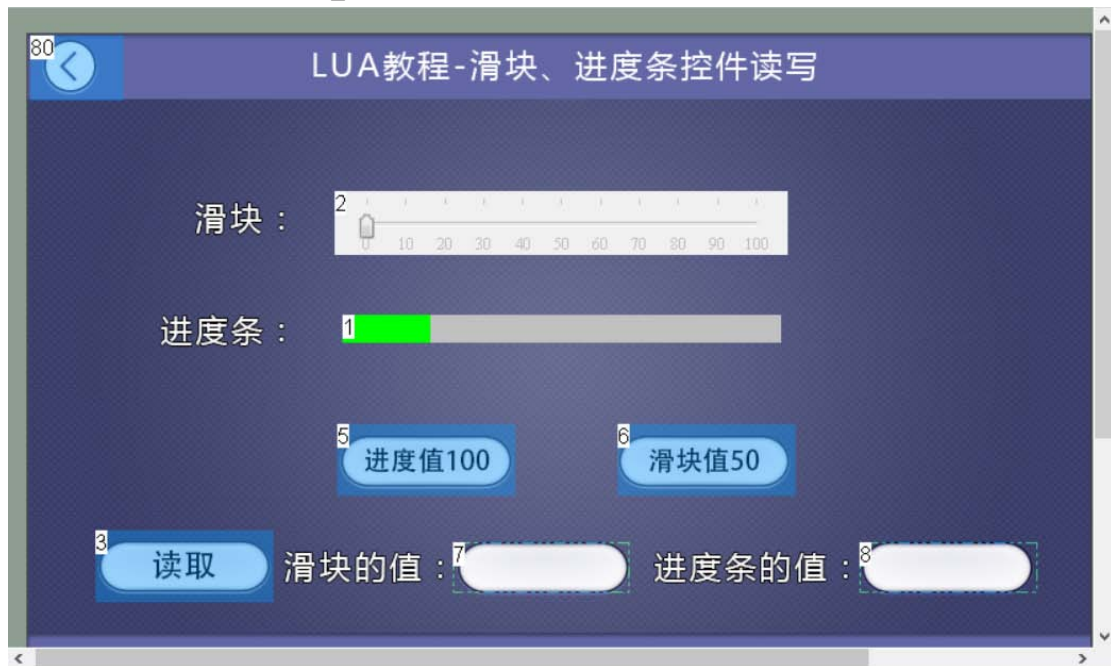


图 5-9 滑块、进度条

b) 教程说明：按下按钮 5、6 分别对进度条和滑块控件的值进行设置，按下按钮 3 可以读取到当前的进度条和滑块的值，读写控件的程序如程序清单 4 所示

程序清单 4 滑块、进度条

```

if screen == 4 then
    if control == 5 and value == 1 then
        set_value(4,1,100)
    end
    if control == 6 and value == 1 then
        set_value(4,2,50)
    end
    if control == 3 and value == 1 then
        local progress_value = get_value(4,1)
        local slider_value = get_value(4,2)
        set_value(4,7,slider_value)
        set_value(4,8,progress_value)
    end
end

```

end

5. 图标控件读写

- a) 图标的值为图标每一帧索引号，索引号是从 0 开始的。使用函数 `get_value` 可以获取到当前图标控件的索引号，使用 `set_value` 可以设置图标显示某一帧，教程演示画面如图 5-10 所示



图 5-10 图标控件

- b) 教程说明：按钮 2、按钮 3 可以控制图标上一帧、下一帧切换，按下按钮 4 可以读取到当前的索引号。读写图标控件的程序如程序清单 5 所示；

程序清单 5 图标控件

```

If screen == 5 then
    if control == 2 and value == 1 then
        Icon_value = Icon_value - 1
        If Icon_value < 0 then
            Icon_value = 6
        end
        set_value(5,1,Icon_value)
    end
    if control == 3 and value == 1 then
        Icon_value = Icon_value + 1
        if Icon_value > 6 then
            Icon_value = 0
        end
        set_value(5,1,Icon_value)
    end
    if control == 4 and value == 1 then
        local value = get_value(5,1)
    end
end

```

```

        set_value(5,9,value)
    end
end

```

6. 滑动选择控件读写

a) 滑动选择控件有两种功能，一种滑动选择，另一种状态显示。调用函数 `get_value` 可以获取到当前滑动选择控件的索引号，使用 `set_value` 可以设置滑动选择选中某一项，教程画面如图 5-11 所示；

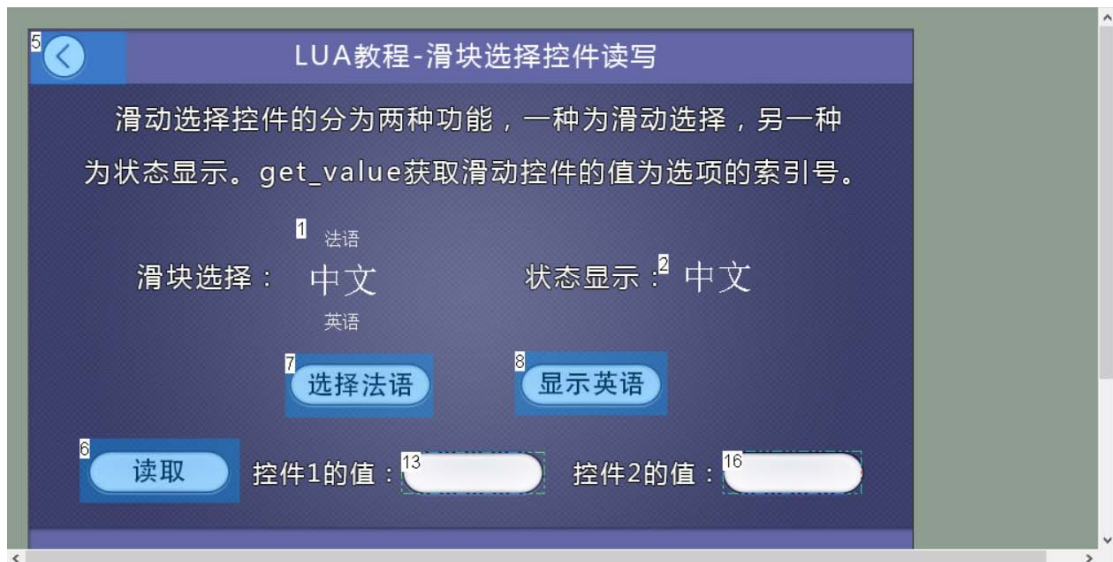


图 5-11 滑动选择

b) 教程说明：按下按钮 7、8 可以控制控件显示指定的某一项，按下按钮 6 可以读取到控件当前的值。读写控件值的程序如程序清单 6 所示

程序清单 6 滑动选择

```

if screen == 6 then
    if control == 7 and value == 1 then
        set_value(6,1,2)
    end
    if control == 8 and value == 1 then
        set_value(6,2,1)
    end
    if control == 6 and value == 1 then
        local selector1_value = get_value(6,1)
        local selector2_value = get_value(6,2)
        set_value(6,13,selector1_value)
        set_value(6,16,selector2_value)
    end
end
end

```

7. 二维码控件读写

a) 二维码控件可以使用函数 `get_text` 获取控件的内容，使用函数 `set_text` 可以修改二维码控件的内容，教程中默认为大彩官网链接，使用浏览器扫描可以直接进入

我司的官网，按下按钮后可以修改二维码控件显示为我司淘宝店铺链接，使用淘宝 app 扫描可以直接跳转到官方店铺。教程画面如图 5-12 所示；



图 5-12 二维码控件

b) 读写二维码控件程序如程序清单 7 所示；

程序清单 7 二维码控件

```
if screen == 7 then
    if control == 7 and value == 1 then
        localqccode_value = get_text(7,2)
        set_text(7,1,qccode_value)
    end
    if control == 6 and value == 1 then
        localqccode_value = get_text(7,1)
        set_text(7,3,qccode_value)
    end
end
end
```

8. 写完控件读写功能模块后将工程和脚本一起进行编译，编译成功后可以使用软件中的虚拟屏查看程序是否实现功能，点击工具栏中编译工程按钮，可将工程和脚本的一起编译，操作如图 5-13 所示

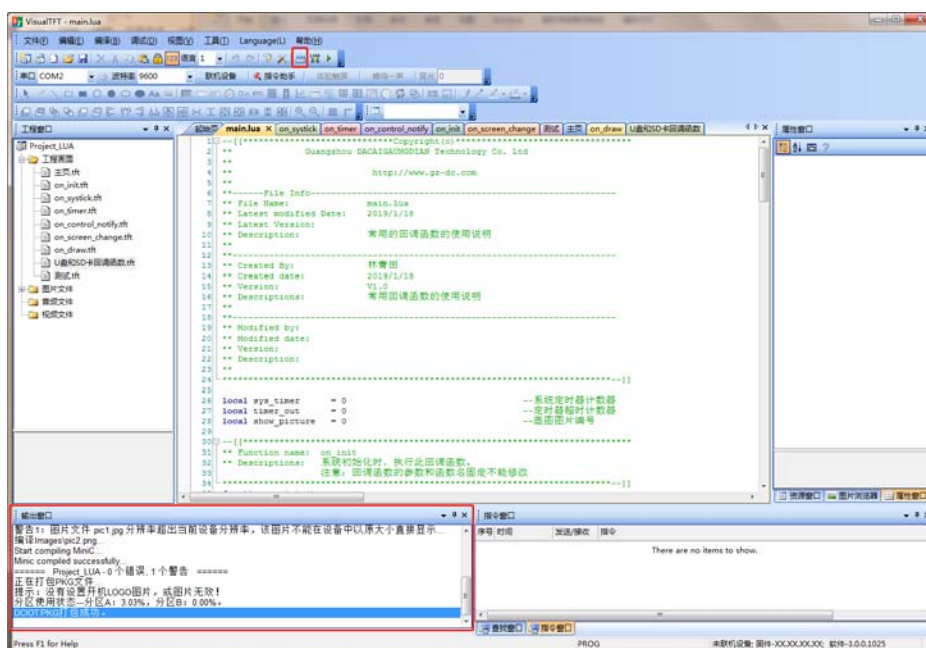


图 5-13 编译工程

注意：目前在软件 Visual TFT 集成的 LUA 脚本编译器无法进行语法检测，所以编写 LUA 程序是尽量分模块编写，已达到节省调试的时间。

9. 点击工具栏中的运行虚拟串口屏，软件先会将当前工程进行编译，编译成功后工程会在虚拟屏上运行起来，如图 5-14 所示

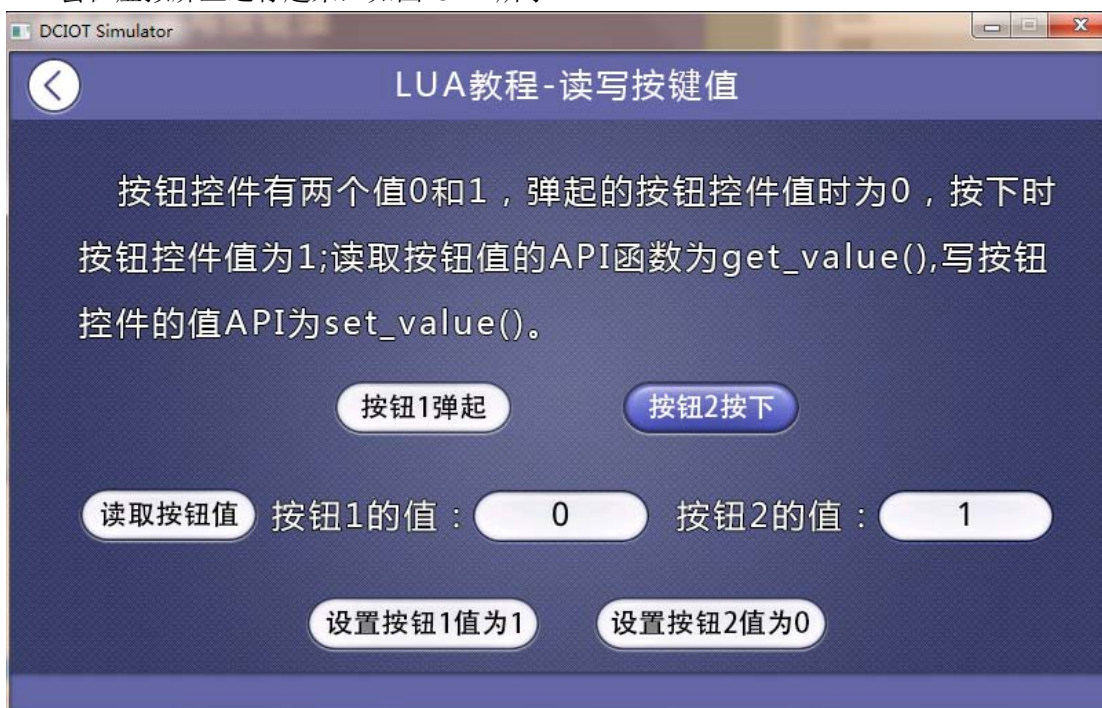


图 5-14 运行程序

5.3.2 显示隐藏控件

1. 使用函数 `set_visible(screen,control,value)` 可以控制组态控件隐藏或者显示，教程画面如图 5-15 所示；



图 5-15 显示或隐藏控件

2. 教程程序如程序清单 8 所示;

程序清单 8 显示或隐藏控件

```

__*****
--功能: 画面 8 控制控件显示或隐藏
--调用函数: set_visiable(screen,control,value) 设置控件是否可见, visiable 为 0 隐藏,
--Input value :    screen 画面 ID
--                control 控件 ID
--                value  0 or 1
--                0 为隐藏 1 为显示
__*****

if screen == 8 then
    if control == 12 and value == 1 then
        for i = 1,11 do
            set_visiable(8,i,0)
        end
    end
    if control == 13 and value == 1 then
        for i = 1,11 do
            set_visiable(8,i,1)
        end
    end
end
end

```

编译工程的方法请参考 5.3.1 中第 8 步；

3. 打开虚拟屏切换到当前画面中，点击按钮控制控件显示/隐藏 如图 5-17 和图 5-16 所示；

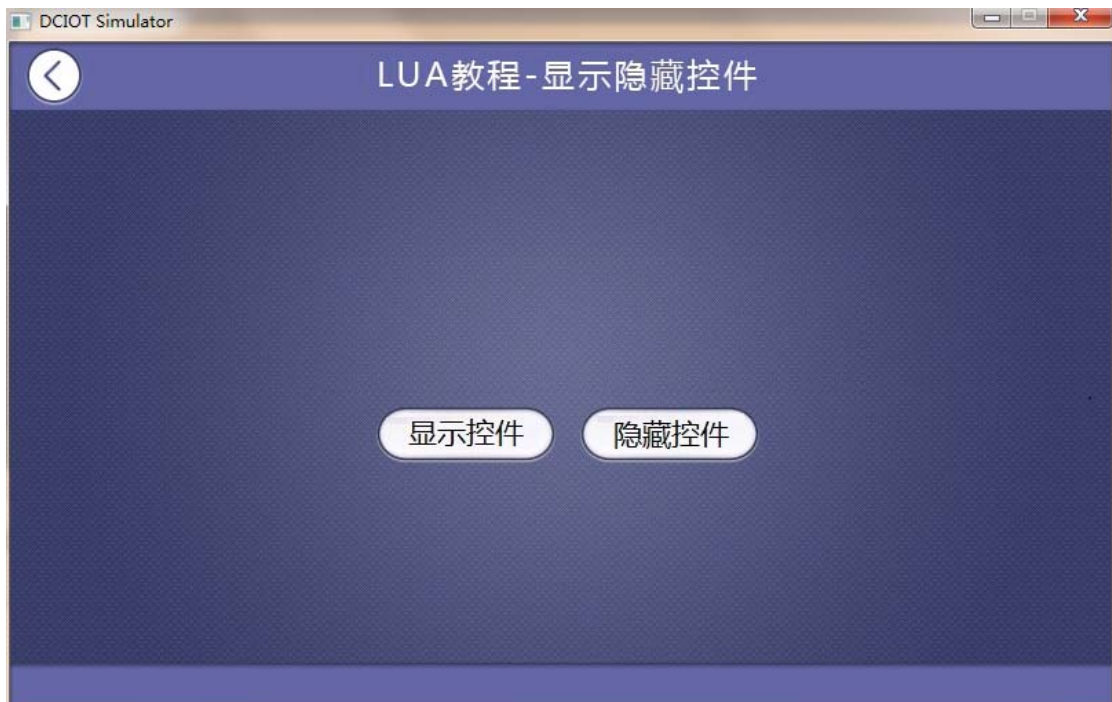


图 5-16 隐藏控件



图 5-17 显示控件

5.3.3 禁止或者使能触摸控件

1. 调用函数 `set_enable (screen ,control, value)` 可以设置控件是否可触摸，教程画面如图 5-18 所示；



图 5-18 禁止使能触摸

2. 教程程序如程序清单 9 所示;

程序清单 9 触摸使能或禁止

```

__*****
--功能：画面 9 控制控件能否触摸
--调用函数：set_enable(screen,control,value) 设置控件是否可触摸
--Input value :    screen 画面 ID
--                control 控件 ID
--                value  0 or 1
--                0 为禁止触摸 1 为触摸使能
__*****

if screen == 9 then
    if control == 12 and value == 1 then
        for i = 1,11 do
            set_enable(9,i,0)
        end
    end
    if control == 13 and value == 1 then
        for i = 1,11 do
            set_enable(9,i,1)
        end
    end
end
end

```


编译工程的方法请参考 5.3.1 中第 8 步；

3. 打开虚拟屏切换到当前画面中，点击按钮控制控件触摸使能/禁止，如图 5-18 所示；



图 5-19 隐藏控件

5.3.4 控件前景色

1. 调用函数 `set_fore_color(screen ,control ,color)`可以设置控件前景色，教程画面如图 5-20 所示



图 5-20 设置前景色

2. 教程程序如程序清单 10 所示；

程序清单 10 设置前景色

```

--*****
--功能: 画面 10 设置控件前景色
--调用函数: set_fore_color(screen,control,color) 设置控件前景色
--Input value : screen 画面 ID
--               control 控件 ID
--               color   RGB565
--*****

if screen == 10 then
    if control == 12 and value == 1 then
        for i = 1,2 do
            set_fore_color(10,i,0xF800)
        end
    end
    if control == 13 and value == 1 then
        for i = 1,2 do
            set_fore_color(10,i,0x1f)
        end
    end
end
end

```

编译工程的方法请参考 5.3.1 中第 8 步;

3. 打开虚拟屏切换到当前画面中, 点击按钮控制控件前景色, 如图 5-21 所示;



图 5-21 前景色

5.3.5 控件背景色

1. 调用函数: set_back_color(screen,control,color)可以设置控件背景色, 教程画面如图

5-22 所示;



图 5-22 设置控件背景色

2. 教程程序如程序清单 11 所示;

程序清单 11 设置背景色

```

--*****
--功能: 画面 11 设置控件背景色
--调用函数: set_back_color(screen,control,color) 设置控件背景色
--Input value :   screen   画面 ID
--               control   控件 ID
--               color     RGB565
--*****

if screen == 11 then
    if control == 12 and value == 1 then
        for i = 1,2 do
            set_back_color(11,i,0xF800)
        end
    end
    if control == 13 and value == 1 then
        for i = 1,2 do
            set_back_color(11,i,0x1f)
        end
    end
end
end

```

编译工程的方法请参考 5.3.1 中第 8 步；

3. 打开虚拟屏切换到当前画面中，点击按钮控制控件背景色，如所示；



图 5-23 背景色

5.4 下载工程

在我司的上层软件 Visual TFT 中集成了 LUA 程序的编译器，可以实现在编译工程的同时将 LUA 脚本程序一起编译，并且将编译后的图片和程序集合在一个名为 DCIOT.PKG 的文件中。编译后只需要把 DCIOT.PKG 文件拷贝到 U 盘中，接上串口屏并重新上电即可将图片和程序下载到屏中。

5.4.1 操作过程

工程编译成功后在输出窗口会提示编译成功，如所示。编译成功后打开工程目录，找到 output 文件夹，将文件夹中的 DCIOT.PKG 文件拷贝到 U 盘中，如图 5-24 和图 5-25 所示；接上串口屏重新上电，等到提示烧录工程成功后，拔掉 U 盘重新上电即可。

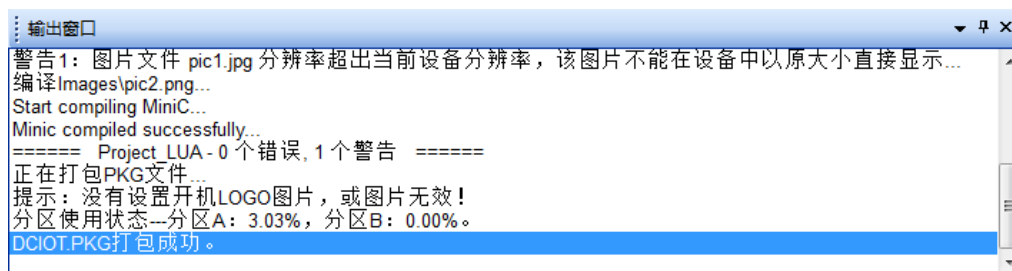


图 5-24 编译成功

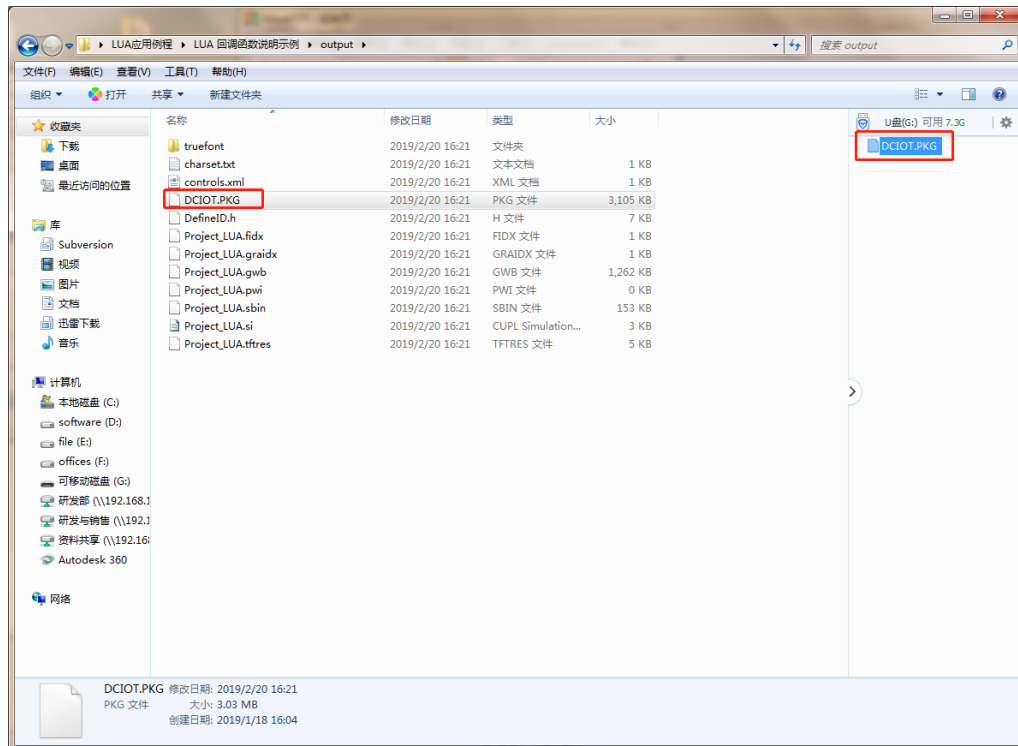


图 5-25 拷贝

6. 完整程序清单

LUA 教程-控件读写 API 函数说明 demo 的完整程序，如程序清单 12 所示，查看程序时请结合教程 demo 理解：

程序清单 12 完整程序

```
--[[*****Copyright(c)*****]]
**
**          Guangzhou DACAIGAUNGDIAN Technology Co. Ltd
**
**
**          http://www.gz-dc.com
**lua 教程网站;          http://www.runoob.com/lua/lua-arrays.html
**-----File Info-----
** File Name:          main.lua
** Latest modified Date: 2019/1/18
** Latest Version:
** Description:        控件 API 函数使用说明
**
**-----
** Created By:          林青田
** Created date:        2019/1/24
** Version:             V1.0
** Descriptions:        控件 API 函数使用说明
**
**-----
** Modified by:
** Modified date:
** Version:
** Description:
**
*****--]]

local  meter_value = 0          --仪表
local  Icon_value = 0          --图标索引号

--[[*****]]
** Function name:  on_init
** Descriptions:   系统初始化时，执行此回调函数。
**                注意：回调函数的参数和函数名固定不能修改
*****--]]

function  on_init()
end

--[[*****]]
** Function name:  on_systick
** Descriptions:   定时回调函数，系统每隔 1 秒钟自动调用。
**                注意：回调函数的参数和函数名固定不能修改
*****--]]
```

```

function on_systick()
end

--[[*****]]
** Function name: on_timer
** Descriptions: 定时器超时，执行此回调函数
                  注意：回调函数的参数和函数名固定不能修改
** Input value : timer_id 定时超时的定时器 ID 号，定时器编号 0~31
                *****--]]

function on_timer(timer_id)
end

--[[*****]]
** Function name: on_control_notify
** Descriptions: 用户通过触摸修改控件后，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。
                  注意：回调函数的参数和函数名固定不能修改
** Input value : screen 画面 ID
                  control 控件 ID
                  value 控件值(包括文本控件输入的值)
                *****--]]

function on_control_notify(screen,control,value)
    _*****
    --功能：按钮控件画面读写按钮控件值
    --调用函数：set_value(screen,control,value) 设置控件数值（也可以设置进度条、滑块、仪表等）
    --调用函数：get_value(screen,control) 获取控件的数值
    --Input value : screen 画面 ID
    --                  control 控件 ID
    --                  value 控件值(0 or 1)
    _*****
    if screen == 1 then
        if control == 4 and value == 1 then --按下按钮 4 获取按钮 1, 2 的值
            local contorl1_value = get_value(1,1)
            local contorl2_value = get_value(1,2)
            set_value(1,6,contorl1_value)
            set_value(1,8,contorl2_value)
        end
        if control == 9 and value == 1 then --按下按钮 9 设置按钮 1 的值为 1
            set_value(1,1,1)
        end
        if control == 10 and value == 1 then --按下按钮 10 设置按钮 2 的值为 0
            set_value(1,2,0)
        end
    end
end

_*****
--功能：文本控件画面读写控件值

```

```

--调用函数: set_value(screen,control,value)
--          set_text(screen,control,value)    设置控件显示内容(字符串), 文本控件, 二维码控件等
--          get_value(screen,control)
--          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

--Input value :   screen   画面 ID
--                  control 控件 ID
--                  value   控件值(包括文本控件输入的值)
__*****

if screen == 2 then
    if control == 4 and value == 1 then        --按下按钮 4 读取文本控件 1, 2 的内容
        local text1_value = get_value(2,1)
        local text2_value = get_text(2,2)
        set_value(2,6,text1_value)
        set_text(2,8,text2_value)
    end
    if control == 9 and value == 1 then        --按下按钮 9 设置文本控件 1 的内容为数值 20
        set_value(2,1,20)
    end
    if control == 10 and value == 1 then       --按下按钮 10 设置文本控件 2 的内容为大彩科技
        set_text(2,2,"大彩科技")
    end
end
__*****

--功能: 仪表控件画面读写控件件值
--调用函数: set_value(screen,control,value)
--调用函数: get_value(screen,control)
--参数 :      screen   画面 ID
--          control 控件 ID
--          value   控件值
__*****

if screen == 3 then
    if control == 3 and value == 1 then
        meter_value = meter_value + 5
        if meter_value >= 360 then
            meter_value = 0
        end
        set_value(3,1,meter_value)
    end
    if control == 4 and value == 1 then
        meter_value = meter_value - 5
        if meter_value < 0 then
            meter_value = 355
        end
    end
end

```



```

        set_value(3,1,meter_value)
    end
    if control == 2 and value == 1 then
        local value = get_value(3,1)
        set_value(3,7,value)
    end
end
--*****
--功能：图标控件画面读写控件值
--调用函数：set_value(screen,control,value)
--调用函数：get_value(screen,control)
--参数：    screen  画面 ID
--          control 控件 ID
--          value   控件值
--*****
if screen == 5 then
    if control == 2 and value == 1 then
        Icon_value = Icon_value -1
        if Icon_value < 0 then
            Icon_value = 6
        end
        set_value(5,1,Icon_value)
    end
    if control == 3 and value == 1 then
        Icon_value = Icon_value +1
        if Icon_value > 6 then
            Icon_value = 0
        end
        set_value(5,1,Icon_value)
    end
    if control == 4 and value == 1 then
        local value = get_value(5,1)
        set_value(5,9,value)
    end
end
--*****
--功能：滑块，进度条控件画面读写控件值
--调用函数：set_value(screen,control,value)
--调用函数：get_value(screen,control)
--Input value：    screen 画面 ID
--                  control 控件 ID
--                  value  控件值(包括文本控件输入的值)
--*****
if screen == 4 then

```

```

    if control == 5 and value == 1 then
        set_value(4,1,100)
    end
    if control == 6 and value == 1 then
        set_value(4,2,50)
    end
    if control == 3 and value == 1 then
        local progress_value = get_value(4,1)
        local slider_value = get_value(4,2)
        set_value(4,7,slider_value)
        set_value(4,8,progress_value)
    end
end
--*****
--功能：滑动选择控件画面读写控件值
--调用函数：set_value(screen,control,value)
--调用函数：get_value(screen,control)
--Input value :   screen  画面 ID
--                control  控件 ID
--                value   控件值(包括文本控件输入的值)
--*****
if screen == 6 then
    if control == 7 and value == 1 then
        set_value(6,1,2)
    end
    if control == 8 and value == 1 then
        set_value(6,2,1)
    end
    if control == 6 and value == 1 then
        local selector1_value = get_value(6,1)
        local selector2_value = get_value(6,2)
        set_value(6,13,selector1_value)
        set_value(6,16,selector2_value)
    end
end
--*****
--功能：二维码控件读写
--调用函数：set_text(screen,control,value)
--调用函数：get_text(screen,control)
--Input value :   screen  画面 ID
--                control  控件 ID
--                value   控件值(包括文本控件输入的值)
--*****
If screen == 7 then

```

```

    if control == 7 and value == 1 then
        local qccode_value = get_text(7,2)
        set_text(7,1,qccode_value)
    end
    if control == 6 and value == 1 then
        local qccode_value = get_text(7,1)
        set_text(7,3,qccode_value)
    end
end

--*****
--功能：画面 8 控制控件显示或隐藏
--调用函数：set_visiable(screen,control,value) 设置控件是否可见，visiable 为 0 隐藏，
--Input value : screen 画面 ID
--                control 控件 ID
--                value 0 or 1
--                0 为隐藏 1 为显示
--*****

if screen == 8 then
    if control == 12 and value == 1 then
        for i = 1,11 do
            set_visiable(8,i,0)
        end
    end
    if control == 13 and value == 1 then
        for i = 1,11 do
            set_visiable(8,i,1)
        end
    end
end

--*****
--功能：画面 9 控制控件能否触摸
--调用函数：set_enable(screen,control,value) 设置控件是否可触摸
--Input value : screen 画面 ID
--                control 控件 ID
--                value 0 or 1
--                0 为禁止触摸 1 为触摸使能
--*****

if screen == 9 then
    if control == 12 and value == 1 then
        for i = 1,11 do
            set_enable(9,i,0)
        end
    end
    if control == 13 and value == 1 then

```

```

        for i = 1 ,11 do
            set_enable(9,i,1)
        end
    end
end

--*****

--功能: 画面 10 设置控件前景色
--调用函数: set_fore_color(screen,control,color) 设置控件前景色
--Input value :   screen   画面 ID
--               control   控件 ID
--               color     RGB565
--*****

if screen == 10 then
    if control == 12 and value == 1 then
        for i = 1 ,2 do
            set_fore_color(10,i,0xF800)
        end
    end
    if control == 13 and value == 1 then
        for i = 1 ,2 do
            set_fore_color(10,i,0x1f)
        end
    end
end

--*****

--功能: 画面 11 设置控件背景色
--调用函数: set_back_color(screen,control,color) 设置控件背景色
--Input value :   screen   画面 ID
--               control   控件 ID
--               color     RGB565
--*****

if screen == 11 then
    if control == 12 and value == 1 then
        for i = 1 ,2 do
            set_back_color(11,i,0xF800)
        end
    end
    if control == 13 and value == 1 then
        for i = 1 ,2 do
            set_back_color(11,i,0x1f)
        end
    end
end

end
end

```

```
--[[*****]]
** Function name:  on_screen_change
** Descriptions:  当画面切换至目标画面 ID 时，执行此回调函数
                   注意：回调函数的参数和函数名固定不能修改
** Input value :   screen  目标画面 ID
*****--]]

function on_screen_change(screen)

end

--[[*****]]
END FILE
*****--]]
```

7. 免责声明

广州大彩光电科技有限公司所提供的服务内容旨在协助客户加速产品的研发进度，在服务过程中或者其他渠道所提供的任何例程程序、技术文档、CAD 图等资料和信息，都仅供参考，客户有权不使用或自行参考修改，本公司不提供任何的完整性、可靠性等保证，若是客户使用过程中因任何原因造成的特别的、偶然的或间接的损失，本公司不承担任何责任。