

类别	内容
关键词	LUA、背光亮度保存、文件 IO
摘要	



## 修订历史

版本	日期	原因	编制	审查
V1.0	2019/01/15	创建文档	林绍佳	刘启鑫



## 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](mailto:hmi@gz-dc.com)（公共服务）

网站： [www.gz-dc.com](http://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： [gz-dc.taobao.com](http://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. 实现教程.....	5
5.1    工程准备.....	5
5.1.1    硬件平台.....	5
5.1.2    素材准备.....	5
5.1.3    LUA 编辑器.....	6
5.2    API 函数说明 .....	7
5.3    教程实现过程.....	8
5.3.1    背光亮度保存.....	8
5.4    编译和下载.....	11
5.4.1    编译工程.....	11
5.4.2    下载.....	12
6. 完整程序清单.....	14
7. 免责声明.....	17

## 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。

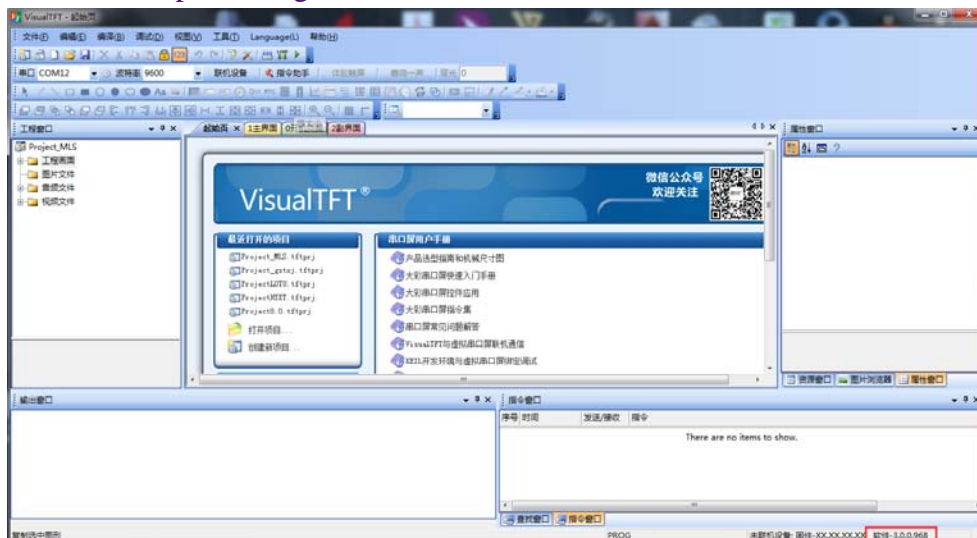


图 2-2 软件版本

2. 串口屏硬件版本: V3.0.301.0 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

物联型可以通过 LUA 脚本配合工程完成丰富多样的操作。

本文将介绍在掉电状态下还能保存背光亮度的方法。在 LUA 脚本中使用文件 IO 系统把数据保存到内存中，并能在重新上电后把数据读出来。工程中通过点击按钮实现保存，再点击读取按钮后，读取上次保存的值，并设置背光亮度值。

## 4. 参考文档

1. 《物联型 LUA 脚本 API\_v1.0》 可通过以下链接下载物联型开发包获取:  
[www.gz-dc.com/index.php?s=/List/index/cid/19.html](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:  
[www.gz-dc.com/index.php?s=/List/index/cid/19.html](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。  
[www.runoob.com/lua/lua-arrays.html](http://www.runoob.com/lua/lua-arrays.html)



## 5. 实现教程

本章节主要讲述通过 LUA 实现背光亮度掉电保存的方法。本文将分为以下 4 个阶段讲述教程 DEMO 是如何实现的：

1. 工程准备，
2. 工程配置与功能，
3. LUA 编写及调试，
4. 下载与运行。

### 5.1 工程准备

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 DC80480W070 为验证开发平台。如图 5-1 所示。

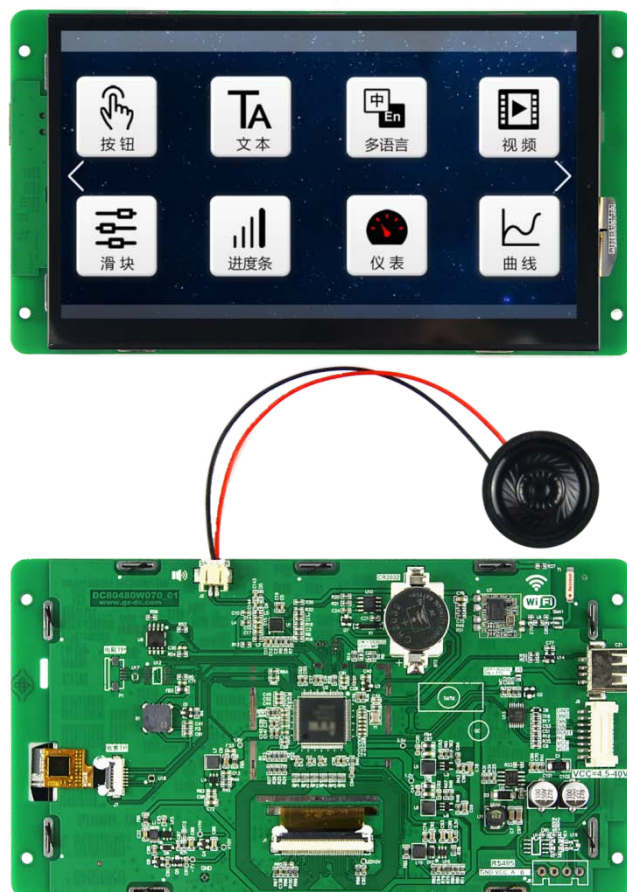


图 5-1 物联型 7 寸

#### 5.1.2 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，如图 5-3 所示。

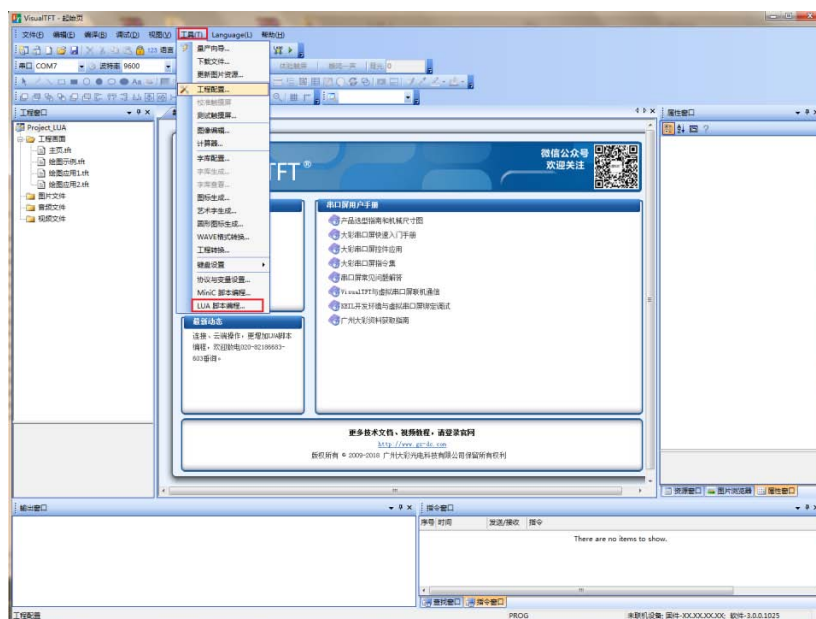


图 5-3 打开 LUA 编辑器

LUA 编辑器的开发环境，如图 5-4 所示。

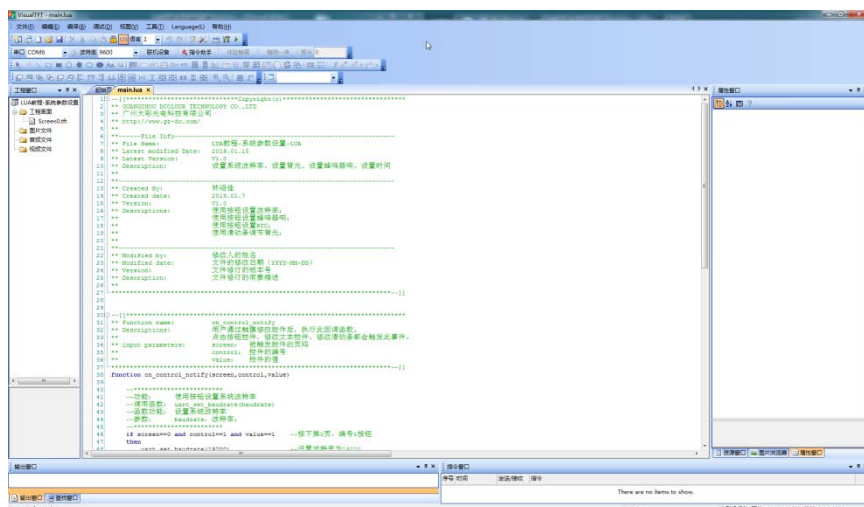


图 5-4 LUA 编辑器界面

## 5.2 API 函数说明

大彩科技针对 LUA 脚本提供了丰富的 API 接口函数，具体函数可以查阅文档《大彩组态串口屏 LUA 脚本 API》，如图 5-5 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

### 1. 函数 on\_control\_notify(screen,control,value)

注释：此函数可以在用户触摸修改控件后被回调。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\_backlight()

注释：此函数获取背光亮度值。

返回背光亮度值

### 3. 函数 set\_backlight(level)，level 表示背光值

注释：此函数背光亮度。

### 4. 函数 set\_value(screen,control,value)

注释：此函数可以设置控件的值。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。

5. 函数 io.open (filename [, mode])

注释：此函数可以打开文件。函数有两个参数：

filename 表示文件名  
[,mode]表示模式。可写可不写。  
返回文件句柄

6. 函数 file:write(text)

注释：此函数可以对文件写入数据。函数有一个参数：

text 表示写入文件的字符串。

7. 函数 file:read([, mode])

注释：此函数可以读出文件的数据。函数有一个参数：

[,mode]表示读的方式。可写可不写。  
返回读出的字符串

8. 函数 file: seek(optional whence, optional offset)

注释：此函数设置和获取当前文件位置。函数有两个参数：

whence 表示设置的方式。可写可不写。  
offset 表示设置的位置。可写可不写。  
都不写时，返回当前位置。

9. 自定义函数 writefile(filename,info)，此函数使用 file:write(text)保存数据。

10. 自定义函数 readfile(filename)，此函数使用 file:read([, mode])读取数据。

注：更多 API 函数请参考文档《物联型 LUA 脚本 API\_V1.0》和网站：

[www.runoob.com/lua](http://www.runoob.com/lua)

## 5.3 教程实现过程

串口屏的背光亮度通过 LUA 脚本保存到内存中。通过点击按钮实现保存动作，再点击读取按钮后，读取上次保存的值，并设置背光亮度值，教程中功能的实现步骤如下：

1. 配置工程属性，
2. 编写 LUA 程序，
3. 运行程序。

### 5.3.1 背光亮度保存

1. 首先设置一个进度条，滑动通知选择关联滑动条。因为背光值太低时会黑屏，所以初始值选择 20。如要使用自定义图片，则在背景类型选择透明，前景类型选择前景图片，再在前景图片中选择自定义的图片，如图 5-6 所示。



图 5-6 按钮的属性

然后设置滑动条，滑块的编号等于进度条的编号加一，滑动条初始值与进度条一致，设置自定义图片的操作也基本一样，如图 5-7。

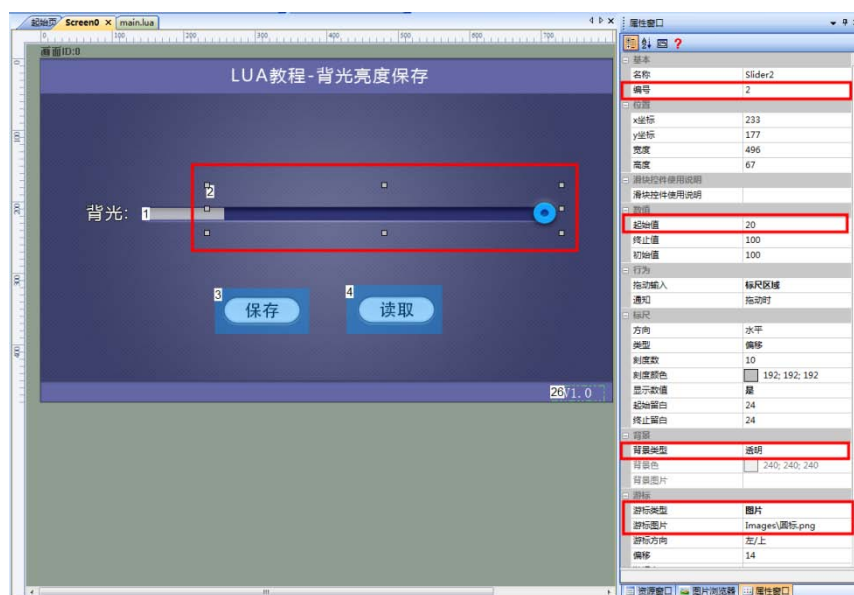


图 5-7 设置滑动条

设置一个按钮用来触发 LUA 中的保存函数，事件通知选择是，再设置按下图片并选择裁剪，如图 5-8。



图 5-8 设置保存按钮

设置一个按钮，事件通知选择是，如图 5-9。



图 5-9 设置读取按钮

2. 滑动滑块时，设置背光，如程序清单 5-1。

程序清单 5-1 设置背光亮度值

```
if screen==0 and control==2
then
    set_backlight(value)        --设置背光
end
```

在按下按钮后，读取文件的数据，然后设置背光并设置进度条，如程序清单 5-2。



程序清单 5-2 读出数据

```

if screen==0 and control==3 and value==1
then
    backlight_value=readfile("backlight.txt")      --读数据
    set_backlight(backlight_value)                --设置背光
    set_value(0,1,backlight_value)                --设置滑块
end

```

把程序编写入软件的 LUA 编辑器中，如图 5-10 所示。

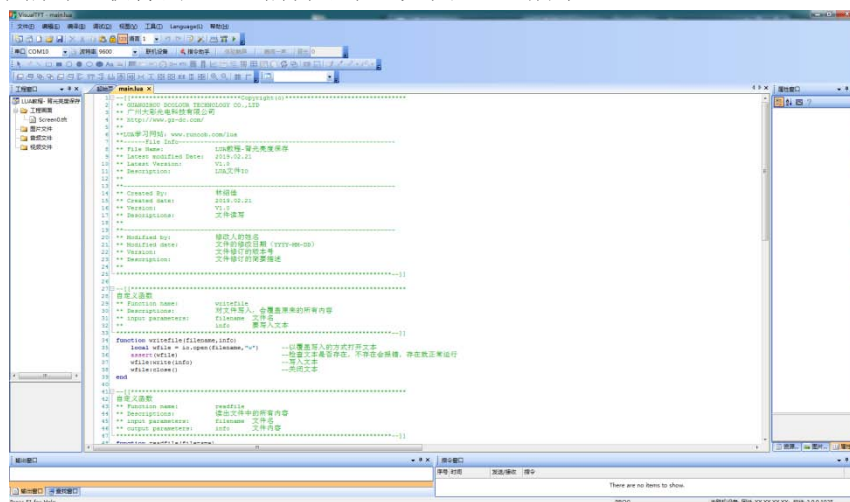


图 5-10 把程序写入 LUA 编辑器

3. 把工程下载到串口屏中，滑动滑块设置背光，点击保存。重新上电，点击读取，背光被设置为上一次的保存值。

## 5.4 编译和下载

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 编译工程

点击菜单中的编译工程后在输出窗口会显示编译过程和结果，如工程控件或图片设置有错误和不恰当，就会提示编译失败和显示错误或警告的位置。如图 5-11 所示

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 程序时尽量分模块编写，以达到节省调试的时间。

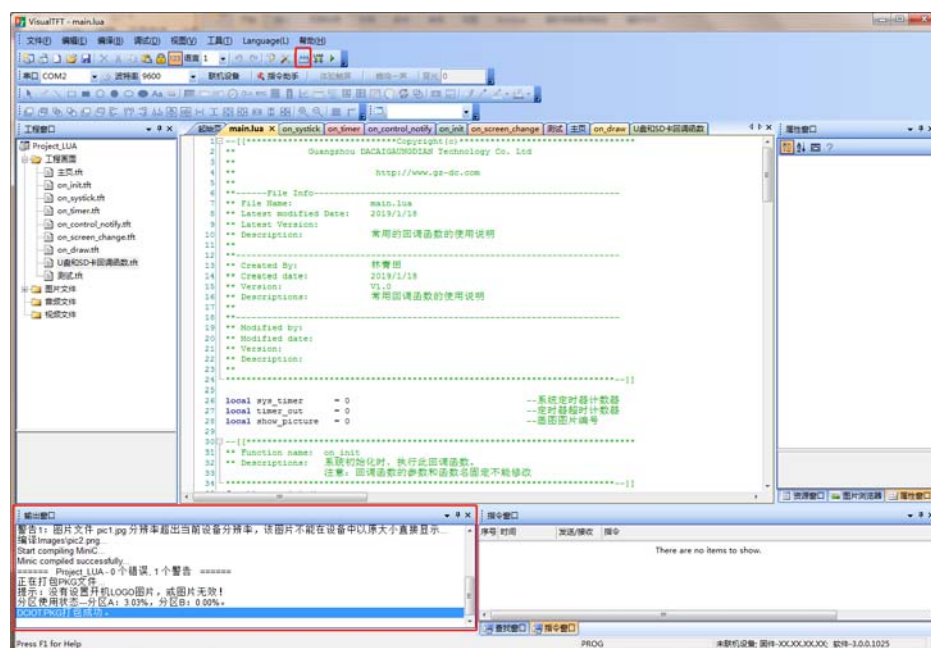


图 5-11 编译

### 5.4.2 下载

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 5-12 和图 5-13 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。

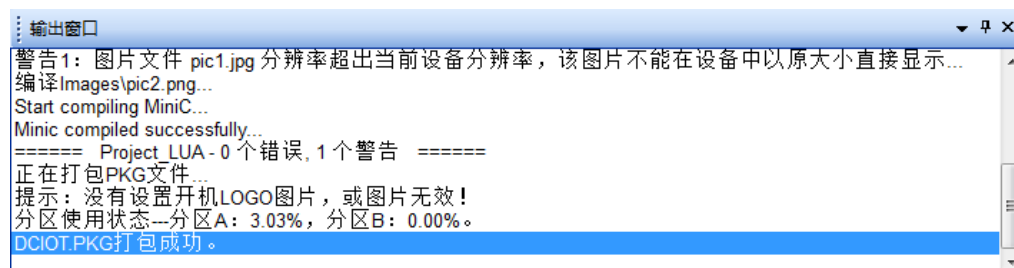


图 5-12 编译成功



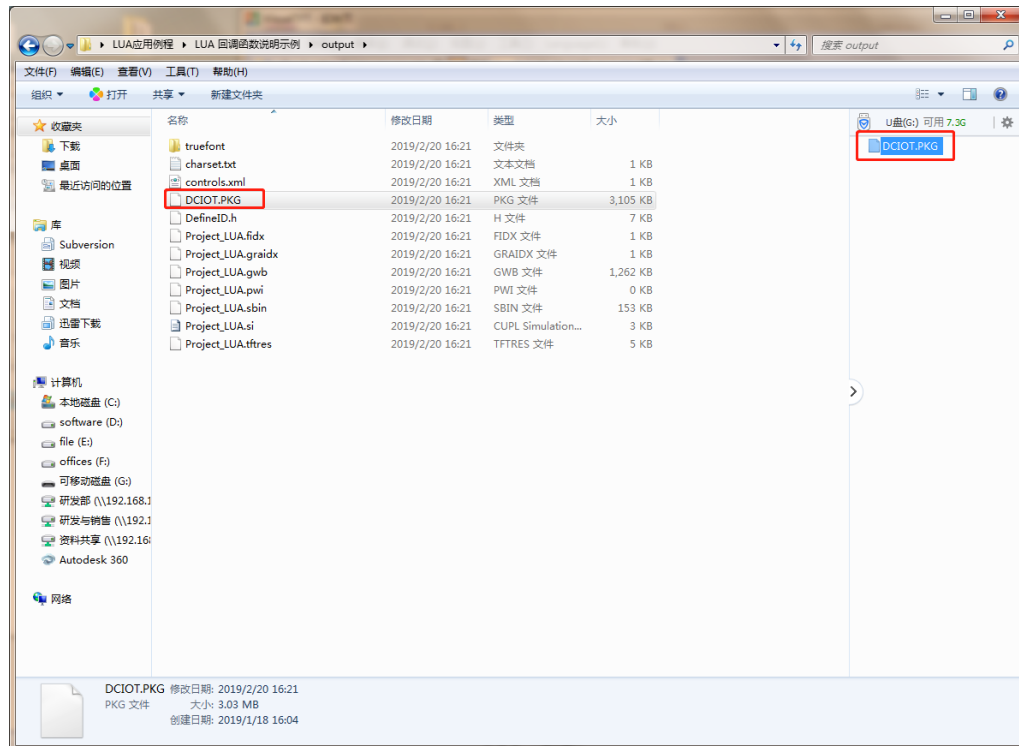


图 5-13 拷贝

## 6. 完整程序清单

背光亮度保存 demo 的完整程序，如程序清单 3 所示，查看程序时请结合教程 demo 理解：

程序清单 3 完整程序

```
--[[*****Copyright(c)*****]]
** GUANGZHOU DCOLOUR TECHNOLOGY CO.,LTD
** 广州大彩光电科技有限公司
** http://www.gz-dc.com/
**
**LUA 学习网站: www.runoob.com/lua
**-----File Info-----
** File Name:          LUA 教程-背光亮度保存-LUA
** Latest modified Date: 2019.02.21
** Latest Version:      V1.0
** Description:         背光亮度保存
**
**-----
** Created By:          林绍佳
** Created date:         2019.02.21
** Version:              V1.0
** Descriptions:         文件读写
**
**-----
** Modified by:          修改人的姓名
** Modified date:         文件的修改日期 (YYYY-MM-DD)
** Version:              文件修订的版本号
** Description:          文件修订的简要描述
**
**-----
*****--]]

--[[*****
自定义函数
** Function name:        writefile
** Descriptions:         对文件写入，会覆盖原来的所有内容
** input parameters:     filename 文件名
**                        info      要写入文本
**-----
*****--]]

function writefile(filename,info)
    local wfile = io.open(filename,"w")    --以覆盖写入的方式打开文本
    assert(wfile)                          --检查文本是否存在，不存在会报错，存在就正常运行
    wfile:write(info)                      --写入文本
    wfile:close()                          --关闭文本
```

```

end

--[[*****]]
自定义函数
** Function name:      readfile
** Descriptions:      读出文件中的所有内容
** input parameters:  filename  文件名
** output parameters: info      文件内容
*****--]]

function readfile(filename)
    local wfile = io.open(filename,"r")    --以只读的方式打开文本
    assert(wfile)                          --检查文本是否存在，不存在会报错，存在就正常运行
    wfile:seek("set")                      --把文件位置定位到开头
    info=wfile:read("a")                   --从当前位置读取整个文件，并赋值到字符串中
    wfile:close()                         --关闭文本
    return info                            --返回字符串
end

--[[*****]]
系统函数
** 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。
** input parameters:  screen:  被触发控件的页码
**                    control:  控件的编号
**                    value:    控件的值
*****--]]

function on_control_notify(screen,control,value)

    --[[*****]]
    --功能:      设置背光
    --调用函数:  set_backlight(level)
    --函数功能:  设置背光值
    --参数:      level:  背光值
    --*****--]]

    if screen==0 and control==2
    then
        set_backlight(value)    --设置背光
    end

    --[[*****]]
    --功能:      把背光值覆盖 backlight.txt 中的内容，原来的数据会被清 0
    --调用函数 1: writefile(filename,info)
    --调用函数 2: get_backlight()

```

```
--函数功能:  获取背光值
--返回:      背光值
--[[*****--]]

if screen==0 and control==3 and value==1
then
    backlight_value=get_backlight()      --获取背光值
    writefile("backlight.txt",backlight_value)  --保存背光值
end

--[[*****
--功能:      读取 backlight.txt 中的背光值, 并设置
--调用函数 1:  readfile(filename)
--调用函数 2:  set_backlight(level)
--函数功能:   设置背光亮度值
--参数:      level:   背光值
--调用函数 3:  set_value(screen,control,text)
--函数功能:   设置控件的值
--参数:      screen:   被触发控件的页码
--            control:  控件的编号
--            text:     要设置的内容
--*****--]]

if screen==0 and control==4 and value==1
then
    backlight_value=readfile("backlight.txt")    --读数据
    set_backlight(backlight_value)              --设置背光
    set_value(0,2,backlight_value)              --设置滑块
end

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。