Technical Note

LUA 例程-简单的运算和字符处理 V1.0

工程技术笔记

类别	内容
关键词	运算、字符处理
摘要	教程

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. 参考资料.....4
5. 实现教程.....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.....10
5.3.1 简单运算.....10
5.3.2 math库函数.....13
5.3.3 字符串处理.....16
5.3.4 简单的逻辑运算.....18
5.4 下载工程.....21
5.4.1 操作过程.....21
6. 完整程序清单.....23
7. 免责声明.....30

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。

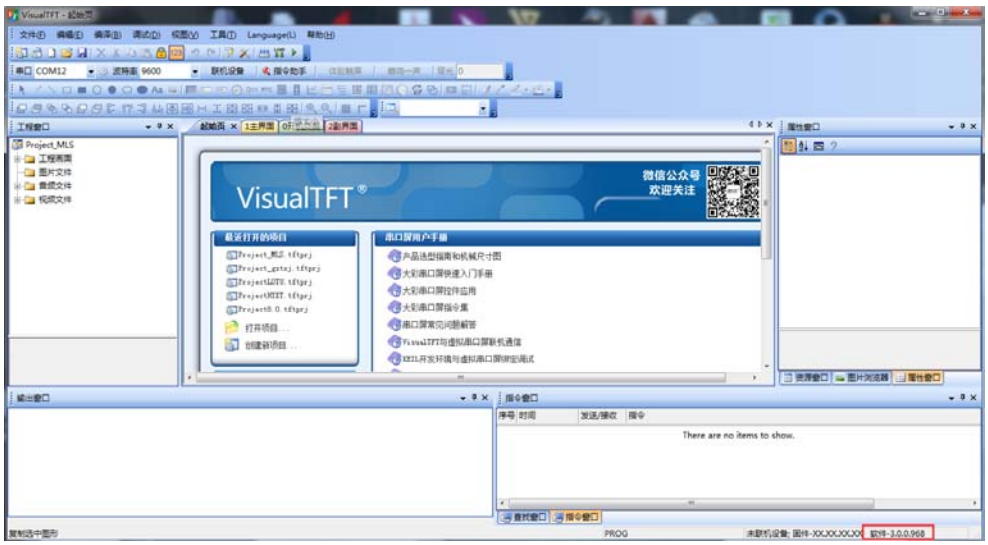

图 2-2 软件版本

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

物联型串口屏通过 LUA 脚本配合工程可以完成大部分的内部逻辑处理，可以做到让 MCU 只参与数据处理，不参与屏的逻辑处理。

本文将介绍如何使用 LUA 脚本在串口屏中实现简单的逻辑运算以及字符处理，本文中部分使用的运算函数和字符处理函数来自于 LUA 的 math 库和 string 库，使用这些函数时可以借助网上的资料了解使用方法。

4. 参考资料

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. 实现教程

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

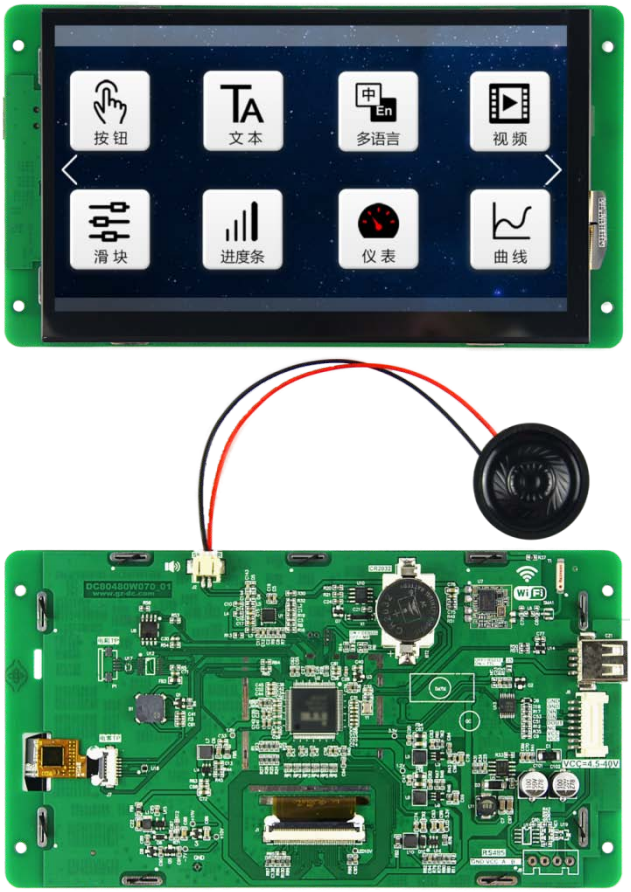

图 5-1 物联型 7 寸串口屏

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，如图 5-3 所示。

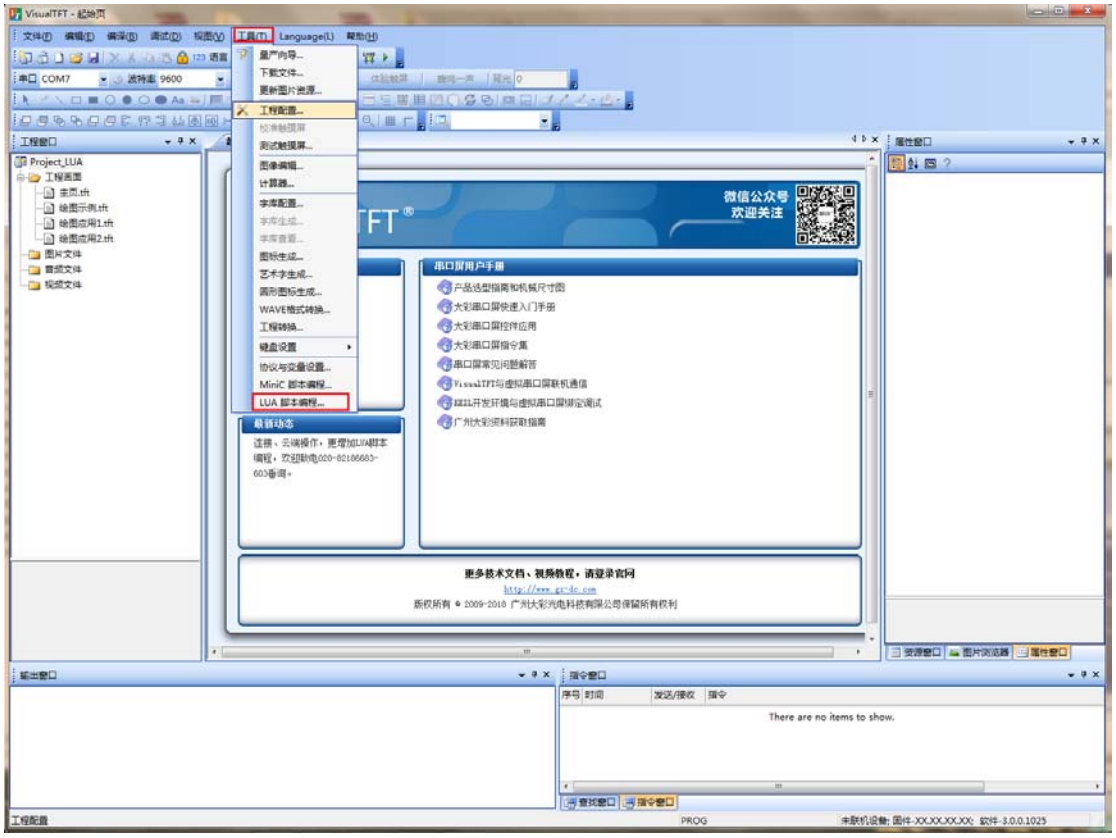

图 5-3 打开 LUA 编辑器

打开编辑器后，Visual TFT 画面如图 5-4 所示；

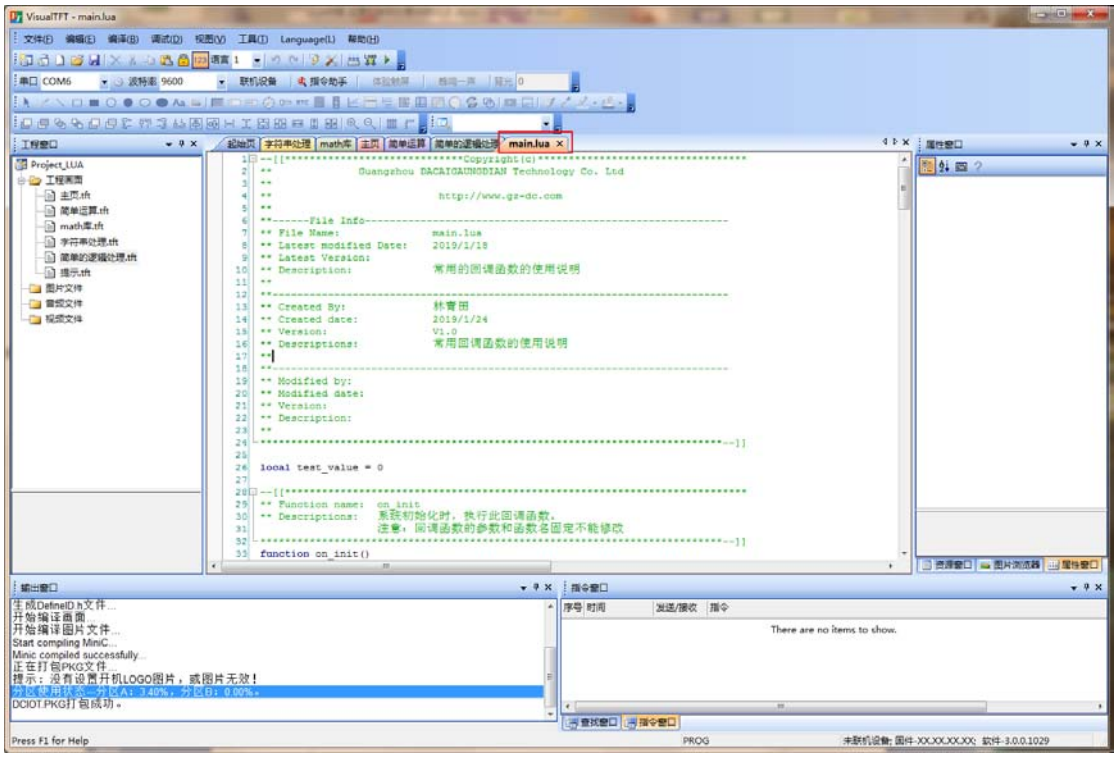

图 5-4 LUA 编辑器

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《物联型 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

1. 本教程文档中所涉及到的部分的相关接口函数如下

函数 set_value(screen,control,value)

注释：设置控件的值

参数 ：screen 画面 ID

control 控件 ID

value 控件值

函数 get_value(screen,control)

注释：获取控件的值。

参数 ：screen 画面 ID

control 控件 ID

函数 set_text((screen,control,str)

注释：获取控件的值。

参数 ：screen 画面 ID

control 控件 ID

str 字符串

函数 get_text((screen,control)

注释：获取控件的值。

参数：screen 画面 ID

control 控件 ID

函数 on_screen_change(screen)

注释：当画面切换至目标画面 ID 时，执行此回调函数

参数：screen 表示目标画面 ID

注：更多 API 函数请参考文档《物联型 LUA 脚本 API_V1.0》

2. math 库函数

教程中用到的部分 math 库函数说明：

函数：math.sqrt(num)

功能：开方

参数：num 数值

函数：math.fmod(x，y)

功能：取模运算（类求 x/y 余）

参数： x 数值

参数： y 数值

函数：math.abs(x)

功能：取绝对值

参数： x 数值

函数：math.modf(x)

功能：获取数值的整数部分及小数部分

参数： x 数值

函数：math.cos(x)

功能：余弦函数

参数： x 弧度

调用函数：math.sin(x) 正弦函数

math.rad(30) 角度转换弧度

参数： 30 角度

函数：math.tan(x) 正切函数

参数： x 弧度

函数：math.max(arg...)

功能：筛选出输入的参数中最大的参数

参数： arg 数值（可多个参数，参数数量不定）

函数：math.min(arg...)

功能：筛选出输入的参数中最小的参数

参数： arg 数值（可多个参数，参数数量不定）

注：关于更多 math 库函数请自行百度相关资料

3. String 库函数

函数：tonumber (e [, base])

功能：尝试将参数 e 转换为数字，当不能转换时返回 nil

参数： e 字符串

[, base]（可选参数，不填时默认转十进制数值，填 16 转十六进制进制数）

函数：tonumber (e)

功能：尝试将参数 e 转换为字符串，当不能转换时返回 nil

参数： e 数值

函数：string.len(str)

功能：在文本框中显示获取字符串长度

参数： str 字符串

函数：string.sub(str,i,j)

功能：在文本框中显示截取字的符串 str_value 的第一个字符

参数： str 字符串

i 起始索引

j 截止索引

注：关于更多 string 库函数请自行百度相关资料

5.3 教程实现

除了我们提供的 API 函数之外还可以使用部分 LUA 相关的库函数，如 math 库，string 库等。本章节主要讲述调用部分库函数在串口屏上的实现简单的逻辑运算和字符处理，教程中每个功能的实现步骤如下：

1. 配置工程属性，
2. 编写程序以及编译，
3. 运行程序。

5.3.1 简单运算

1. 教程 DEMO 在此画面中简单的演示了 LUA 基础运算中的加减乘除和关系运算，教程 DEMO 画面如图 5-6 所示，关于画面中控件具体属性配置请参考教程 DEMO。

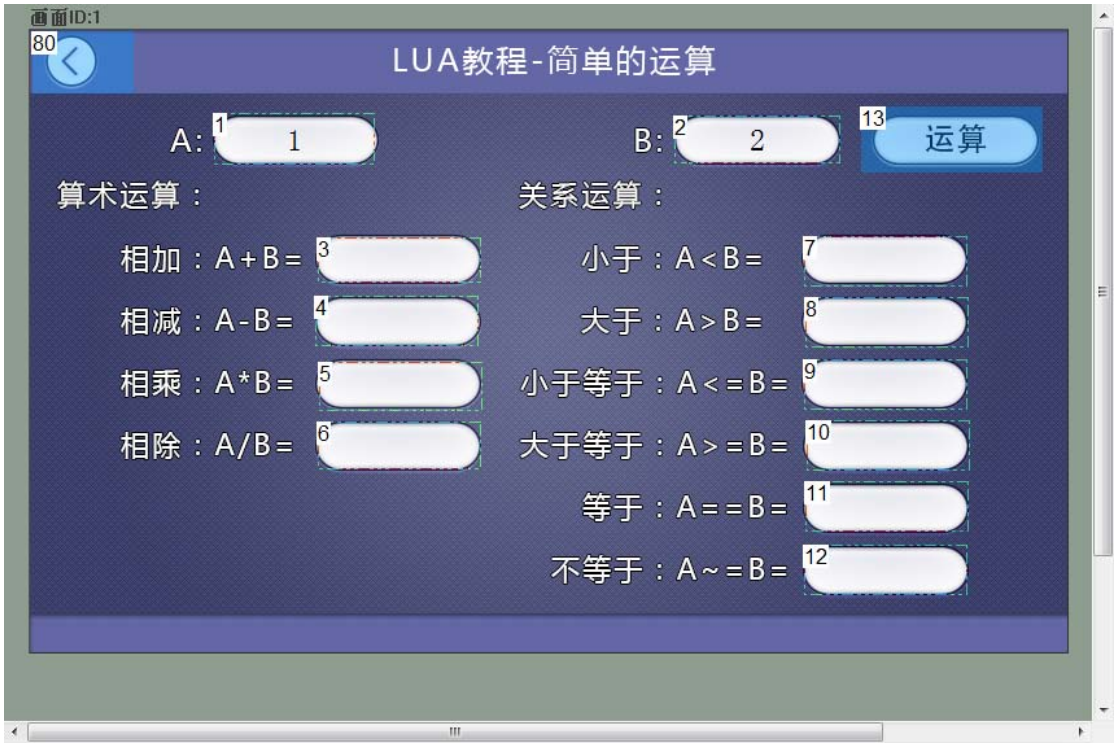

图 5-6 简单的运算

2. 完成配置控件属性后，打开 LUA 编辑器，编写程序实现显示加减乘除等运算及在画面中显示运算结果。程序逻辑说明：点击运算按钮后系统会调用 on_control_notify(screen,control,value)，并将当前画面 ID、按钮控件 ID 以及按钮的

值传入函数中，通过 IF 判断当按钮按下的时候将运算结果显示用 set_value( )。程序如程序清单 1 所示；

程序清单 1 显示运算结果

```
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)
--调用函数：get_value(screen,control)
--Input value :     screen    画面 ID
--                            control  控件 ID
--                            value     控件值
--*****************************************************
if    screen == 1 then
      if control == 13 and value == 1 then              --按下按钮显示运算结果
            local A = get_value(1,1)                    --获取画面 1 的控件 1 的内容
            local B = get_value(1,2)
            set_value(1,3,A+B)                          --显示 A+B 的结果
            set_value(1,4,A-B)                          --显示 A-B 的结果
             set_value(1,5,A*B)                         --显示 A*B 的结果
             set_value(1,6,A/B)                         --显示 A/B 的结果
             if A < B    then                           --显示 A 与 B 的比较
                   set_value(1,7,1)                     --将 A 和 B 比较结果在画面 1 中的文本控件 7 中显示
             else
                  set_value(1,7,0)
               end
             if A > B    then
                   set_value(1,8,1)
             else
                  set_value(1,8,0)
             end
             if A <= B    then
                   set_value(1,9,1)
             else
                  set_value(1,9,0)
             end
             if A >= B    then
                   set_value(1,10,1)
             else
                  set_value(1,10,0)
             end
             if A == B    then
                   set_value(1,11,1)
             else
```

```
                set_value(1,11,0)
            end
            if A ~= B    then
                set_value(1,12,1)
            else
                set_value(1,12,0)
            end
        end
    end
```

编写完基础运算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 5-7 所示

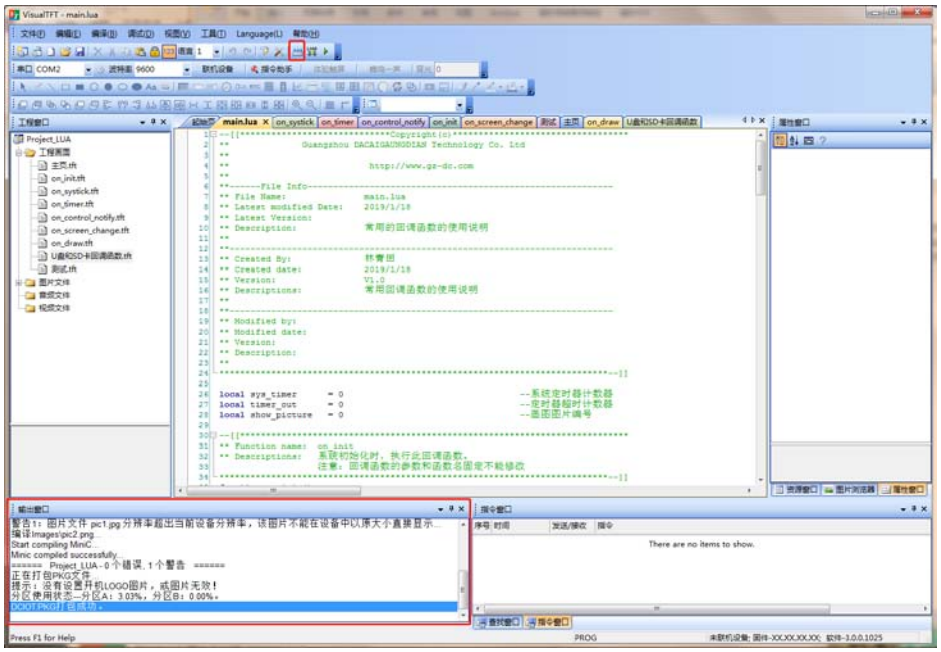

图 5-7 编译工程

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。

3. 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，如和所示

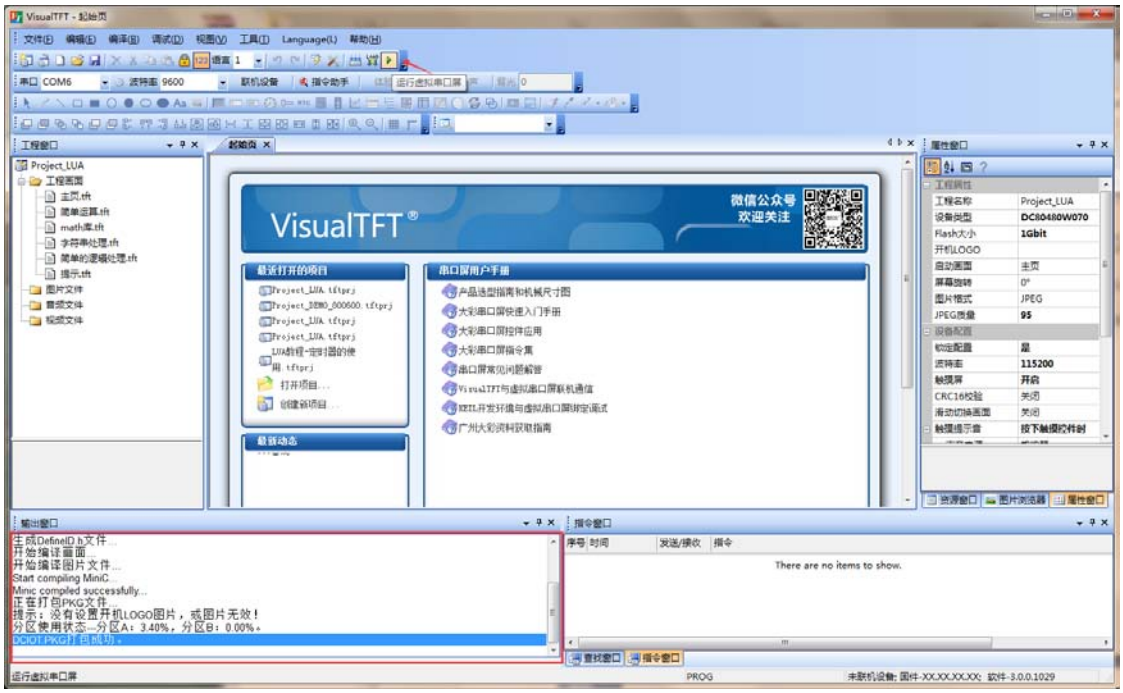

图 5-8 打开虚拟屏

在虚拟屏中切换到在当前画面并在 A，B 文本框输入值后点击运算按钮，会在相对应的文本框中显示运算结果，如图 5-9 虚拟屏所示；

图 5-9 虚拟屏

5.3.2 math 库函数

1. LUA 的 math 库函数中提供了各种数学运算函数，对于大部分数据处理可以直接在屏内使用脚本进行处理，无需外部 MCU 的参与。教程 DEMO 演示了部分常用的数学运算函数的使用方法，画面如图 5-10 所示；

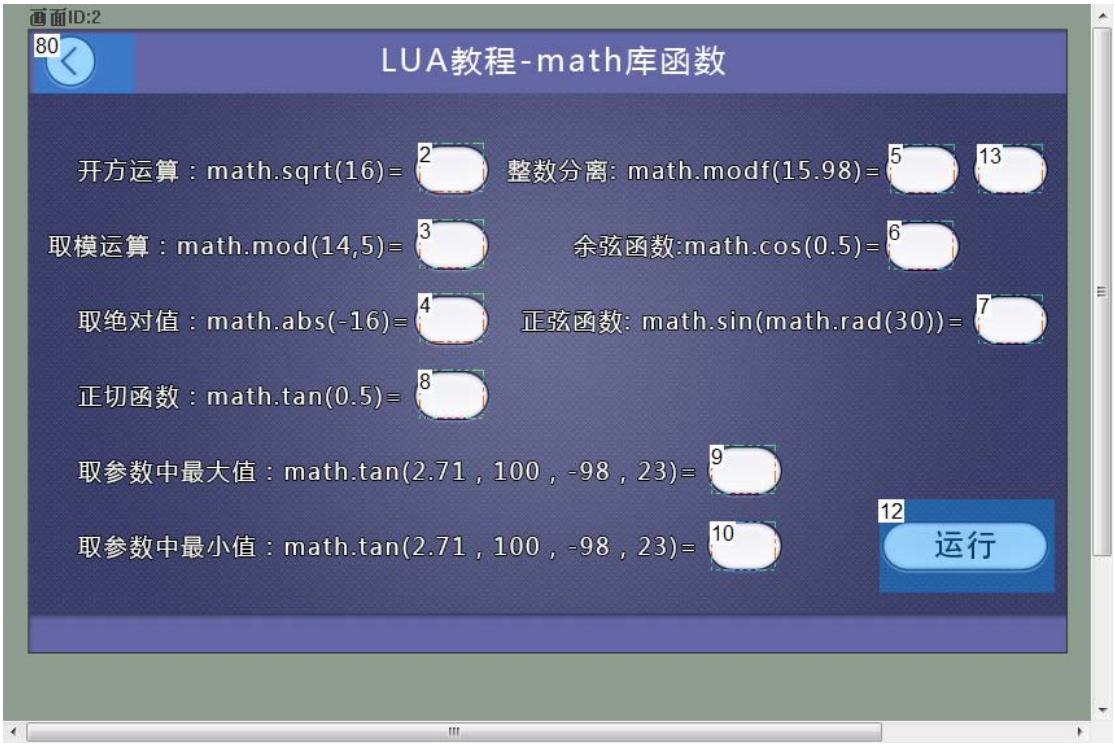

图 5-10 math 库函数

2. 完成配置控件属性后，打开 LUA 编辑器，编写程序实现调用 math 函数进行运算及在画面中显示运算结果。

LUA 程序说明：判断当前画面的按钮按下后，利用 API 函数 set_value 将 math 库函数的运算结果在文本框中显示出来，调用函数及显示运算结果的程序如程序清单 2 所示；

程序清单 2 显示运算结果

```
if screen == 2 then
    if control == 12 and value == 1 then                    --按下按钮 12，运行函数
        local value = 0
        local value2 = 0
    --************************************************************
    --功能：开方
    --调用函数：math.sqrt(num)
    --参数 :    num   数值
    --************************************************************
        value = math.sqrt(16)
        set_value(2,2,value)
    --************************************************************
    --功能：取模运算（类求 x/y 余）
    --调用函数：math.fmod(x，y)
    --参数 :    x   数值
    --参数 :    y   数值
    --************************************************************
```

```
	value = math.fmod(14,5)
	set_value(2,3,value)
--********************************************************
--功能：取绝对值
--调用函数：math.abs(x)
--参数 ：	x	数值
--********************************************************
	value = math.abs(-16)
	set_value(2,4,value)
--********************************************************
--功能：获取数值的整数部分及小数部分
--调用函数：math.modf(x)
--参数 ：	x	数值
--********************************************************
	value,value2 = math.modf(15.98)
	set_value(2,5,value)
	set_value(2,13,value2)
--********************************************************
--功能：余弦函数
--调用函数：math.cos(x)
--参数 ：	x			弧度
--********************************************************
	value = math.cos(0.5)
	set_value(2,6,value)
--********************************************************
--功能：正弦函数
--调用函数：math.sin(x)
--		math.rad(30)	角度转换弧度
--参数 ：	30			角度
--********************************************************
	value = math.sin(math.rad(30))
	set_value(2,7,value)
--********************************************************
--功能：正切函数
--调用函数：math.tan(x)
--参数 ：	x	弧度
--********************************************************
	value = math.tan(0.5)
	set_value(2,8,value)
--********************************************************
--功能：筛选出输入的参数中最大的参数
--调用函数：math.max(arg...)
--参数 ：	arg	数值（可多个参数，参数数量不定）
--********************************************************
```

```
        value = math.max(2.71,100,-98,23)
        set_value(2,9,value)
    --************************************************************
    --功能：筛选出输入的参数中最小的参数
    --调用函数：math.min(arg...)
    --参数 ：    arg   数值（可多个参数，参数数量不定）
    --************************************************************
        value = math.min(2.71,100,-98,23)
        set_value(2,10,value)
    end
```

工程编译请参考上一小节 5.3.1 的第 2 步；

3. 点击运行串口屏后切换到当前画面，点击画面中运行按钮后在当前画面中显示函数运行结果，如图 5-11 所示；

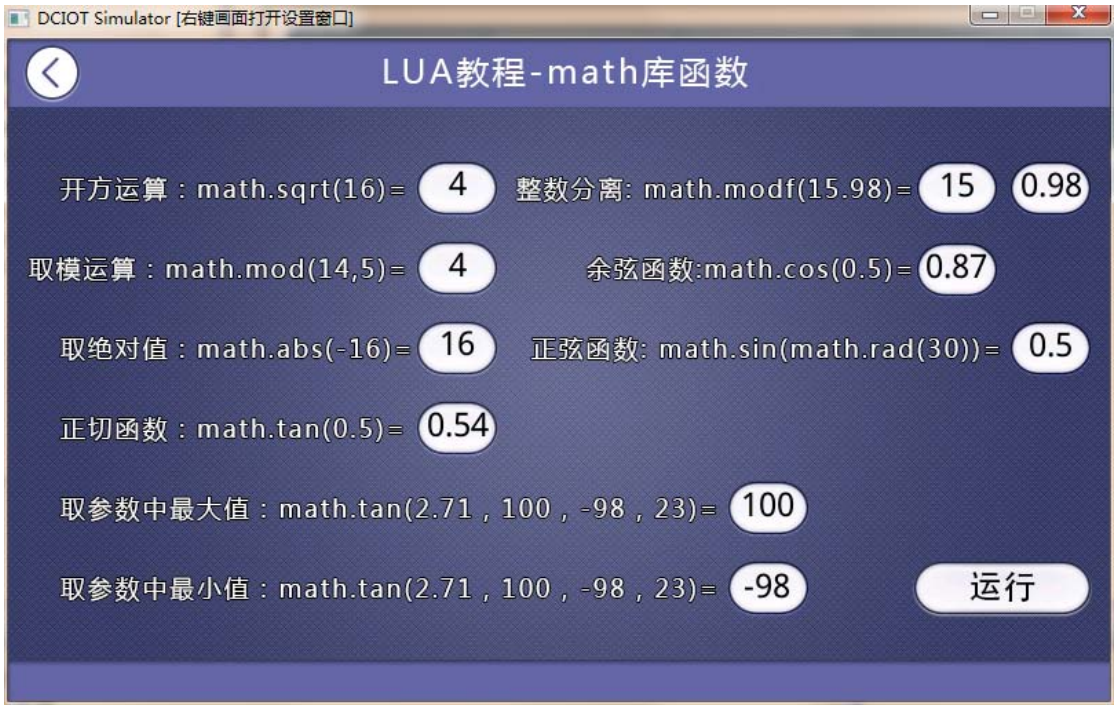

图 5-11 运行虚拟屏

5.3.3 字符串处理

1. 通过LUA脚本string库函数以及字符串特性，可以实现将两个或多个字符串拼接起来在一个文本框中显示，而且不需要外部MCU的参与。教程演示画面如图 5-12 所示；

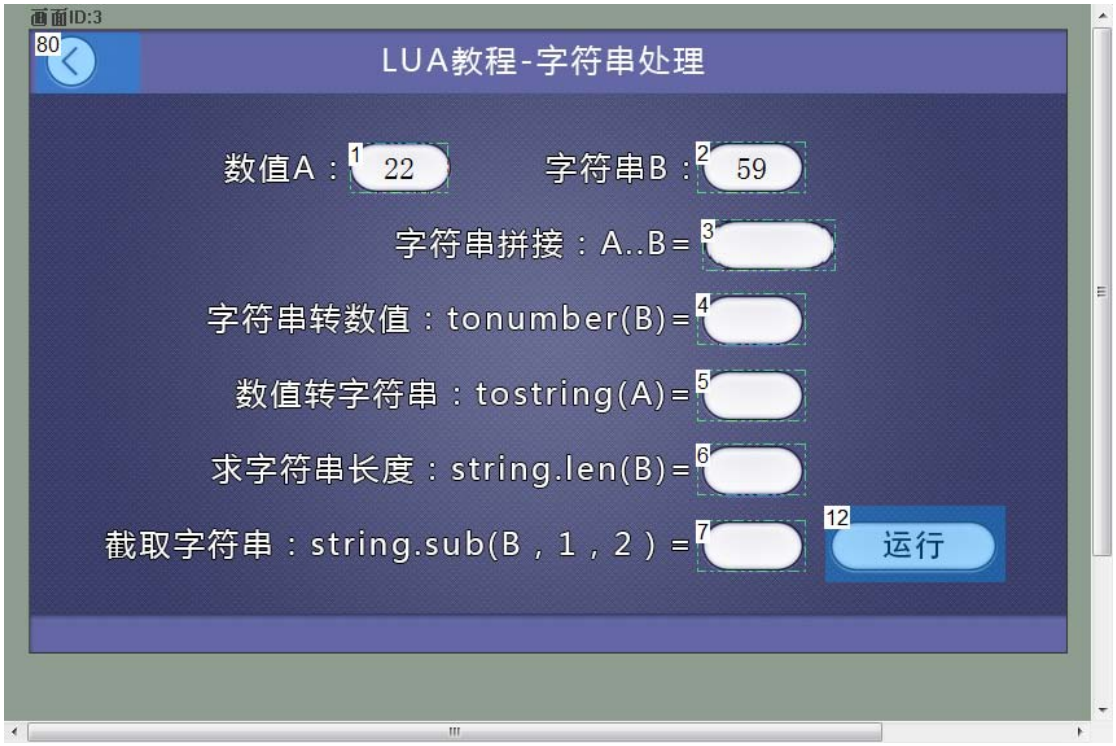

图 5-12 字符串处理

2. 画面配置完成后，打开 LUA 编辑器编写对应程序。

LUA 程序说明：当判断按钮按下后，利用 API 函数 set_value 和 set_value 将字符处理的结果在文本框中显示出来，调用函数及显示结果的程序如程序清单 3 中所示。

程序清单 3 显示图片

```
if screen == 3 then
    if control == 12 and value == 1 then                    --判断按钮按下
        local str_value = 0
        local str_value2 = 0
        str_value2   = get_text(3,2)
        str_value = get_value(3,1)
        set_text(3,3,str_value..str_value2)                 --字符串拼接
--*********************************************************
--功能：尝试将参数 e 转换为数字，当不能转换时返回 nil
--调用函数：tonumber (e [, base])
   --参数 :     e   字符串
--            [，base]（可选参数，不填时默认转十进制数值，填 16 转十六进制进制数）
--*********************************************************
        If   tonumber(str_value2) == nil then               --字符串转数值
            set_value(3,4,0)                                --判断转换失败时显示 0
        else
            set_value(3,4,tonumber(str_value2))
        end
--*********************************************************
```

```
--功能：尝试将参数 e 转换为字符串，当不能转换时返回 nil
--调用函数：tonumber (e)
    --参数 :     e   数值
--*********************************************************
            set_text(3,5,tostring(str_value))                    --数值转字符串
--*********************************************************
--功能：在文本框中显示获取字符串长度
--调用函数：tonumber (str)
    --参数 :     str      字符串
--*********************************************************
            set_value(3,6,string.len(str_value2))                --string.len 获取字符串的长度
--*********************************************************
--功能：在文本框中显示截取字的符串 str_value 的第一个字符
--调用函数：string.sub(str,i,j)
    --参数 :     str   字符串
--              i       起始索引
--              j       截止索引
--*********************************************************
            set_text(3,7,string.sub(str_value2,1,2))             --string.sub 获取字符串的长度
      end
end
```

工程编译请参考上一小节 5.3.1 的第 2 步

3. 运行虚拟屏后切换到当前画面中，在对应的文本控件中输入值后点击运行，调用函数的处理结果会在画面中显示出来，如所图 5-13 示；

图 5-13 虚拟屏-字符串处理

5.3.4 简单的逻辑运算

1. 教程中列举了一个简单的逻辑运算演示，通过放置两个按钮，一个为按下时候文本中的数值加一，另一个按下时文本中的数值减一，并设置最大值为 10，最小值为 0，达到最大值后跳出对话框提示已达到最大值，教程画面如图 5-14 所示；

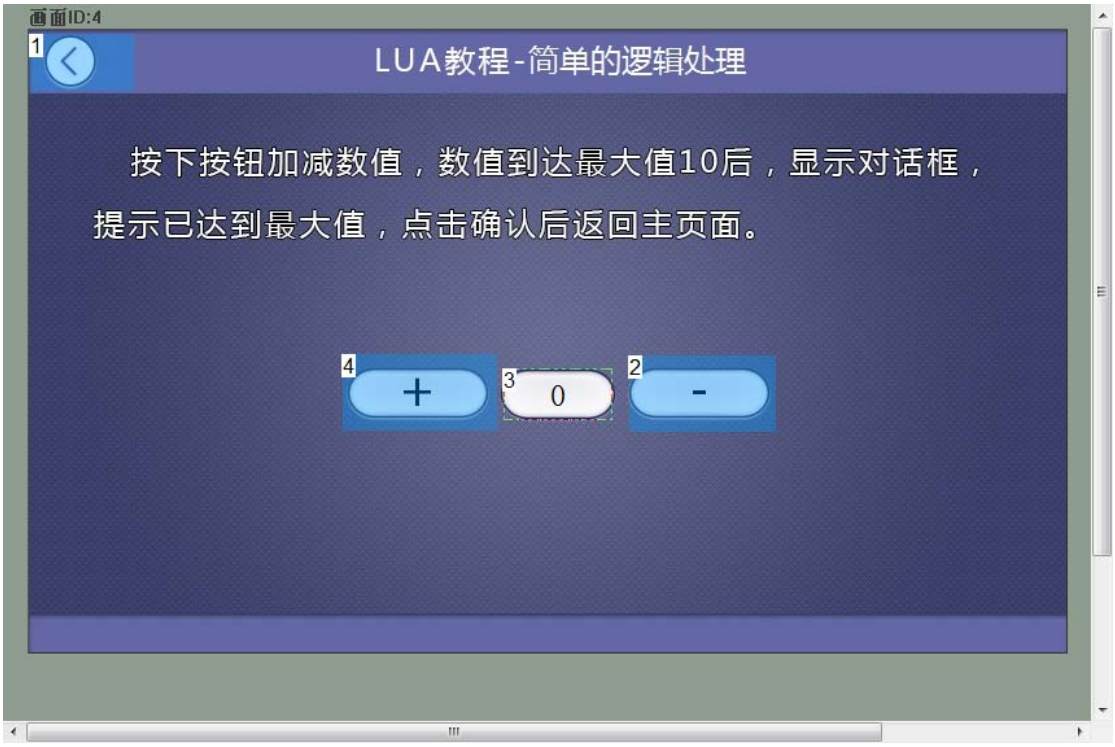

图 5-14 简单的逻辑处理

2. 配置完画面后，打开 LUA 编辑器编写对应功能的程序。

LUA 程序说明：先在程序中定义一个变量，然后在每次按钮按下变量加一或减一，然后加上判断是否达到最大值或最小值，达到最值后调用函数 change_screen 切换到对话框的画面,提示已经到了最大值。按钮加减变量值以及判断变量大小的的代码如所示

程序清单 4 简单的逻辑处理

```
local    test_value = 0                                    --定义局部变量
--**********************************************************
 --功能：达到最大值后弹出提示框。提示已到达最大值
 --**********************************************************
 if screen == 4 then
 --**********************************************************
 --功能：按一下按钮计数值减一
 --**********************************************************
       if control == 2 and value == 1 then
              test_value = test_value - 1
              if    test_value < 0 then
                     test_value = 0
              end
              set_value(4,3,test_value)
```

```
        end
--*********************************************************
--功能：按一下按钮计数值加一
--*********************************************************
        if control == 4 and value == 1 then
                test_value = test_value + 1
                if test_value > 10 then
                        change_screen(5)
                        test_value = 10
                end
                set_value(4,3,test_value)
        end
end
```

工程编译请参考上一小节 5.3.1 的第 2 步

3. 运行虚拟屏后切换到当前画面会，点击加减按钮实现对文本控件中的数值，如图 5-15 和图 5-16 所示；

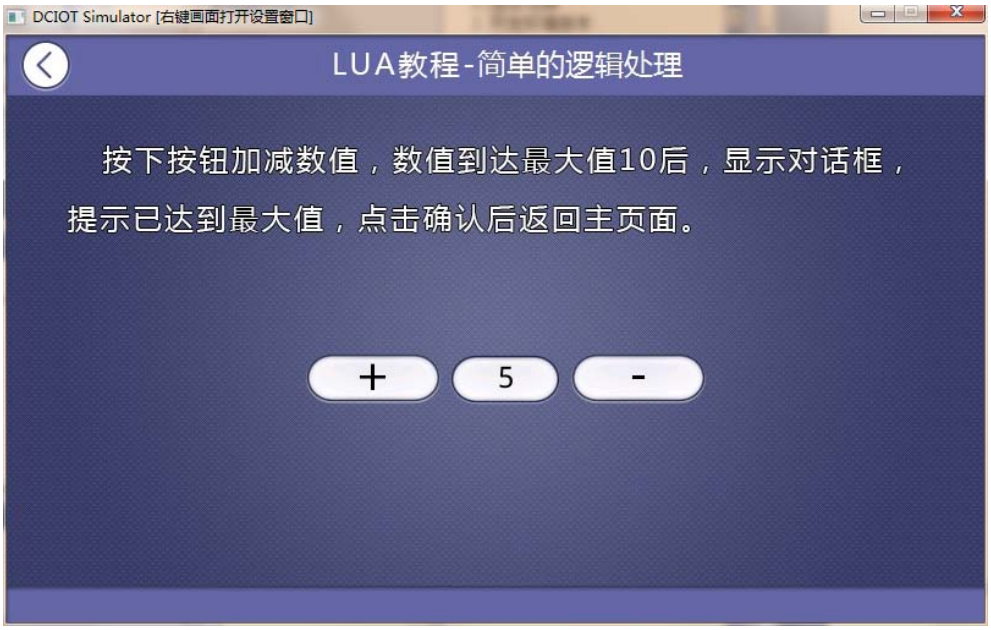

图 5-15 点击按钮

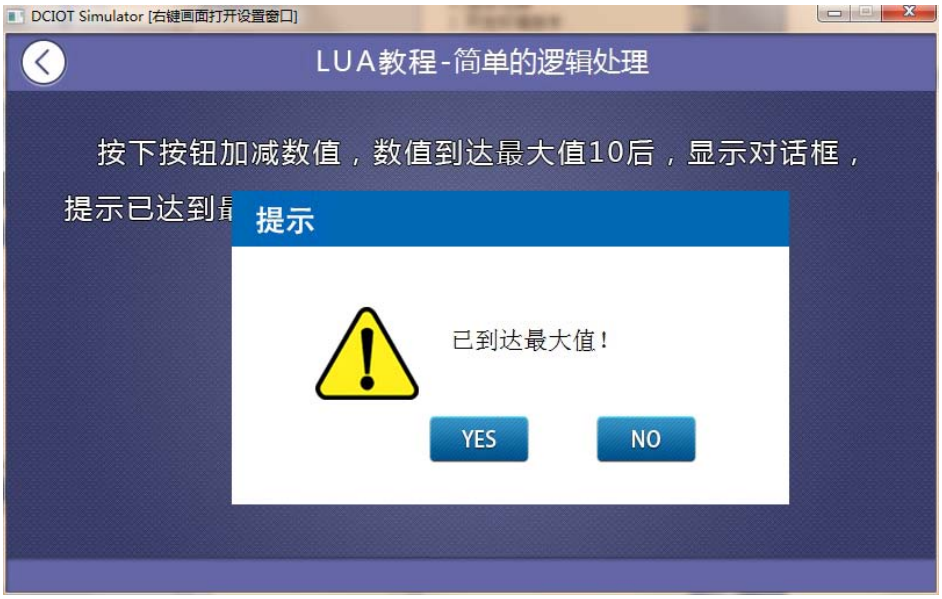

图 5-16 提示框

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。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 5-17 和图 5-18 所示；然后把 U 盘接上串口屏重新上电，等待提示烧录工程成功后，拔掉 U 盘重新上电即可。

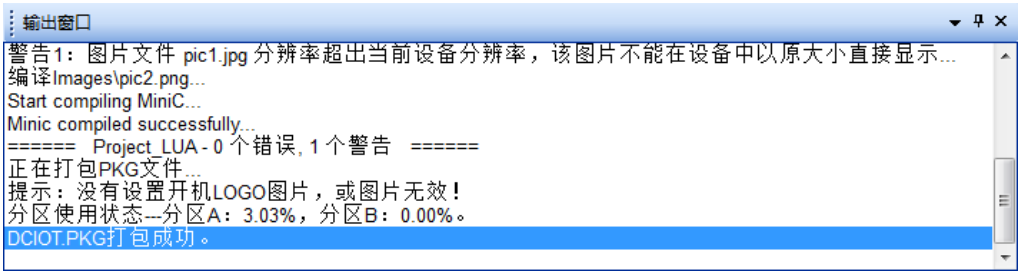

图 5-17 编译成功

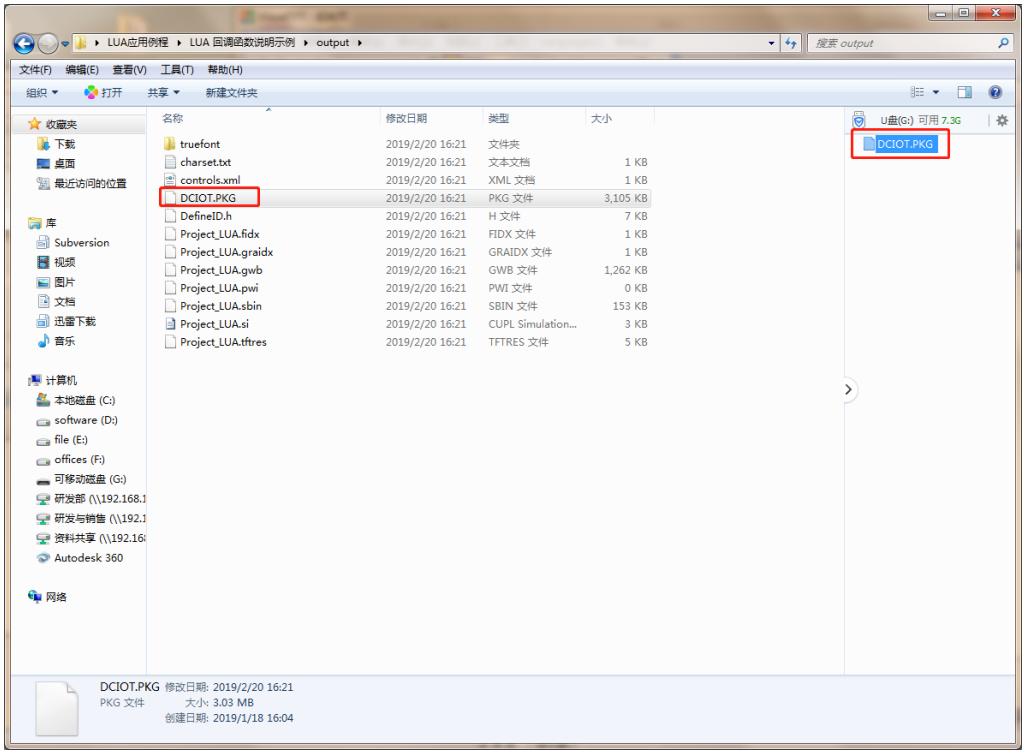
LUA应用例程 ▸ LUA 回调函数说明示例 ▸ output ▸
搜索 output
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 打开 共享 新建文件夹
名称 修改日期 类型 大小
truefont 2019/2/20 16:21 文件夹
charset.txt 2019/2/20 16:21 文本文档 1 KB
controls.xml 2019/2/20 16:21 XML 文档 1 KB
DCIOT.PKG 2019/2/20 16:21 PKG 文件 3,105 KB
DefineID.h 2019/2/20 16:21 H 文件 7 KB
Project_LUA.fidx 2019/2/20 16:21 FIDX 文件 1 KB
Project_LUA.graidx 2019/2/20 16:21 GRAIDX 文件 1 KB
Project_LUA.gwb 2019/2/20 16:21 GWB 文件 1,262 KB
Project_LUA.pwi 2019/2/20 16:21 PWI 文件 0 KB
Project_LUA.sbin 2019/2/20 16:21 SBIN 文件 153 KB
Project_LUA.si 2019/2/20 16:21 CUPL Simulation... 3 KB
Project_LUA.tftres 2019/2/20 16:21 TFTRES 文件 5 KB
U盘(G:) 可用 7.3G
DCIOT.PKG
DCIOT.PKG 修改日期: 2019/2/20 16:21
PKG 文件 大小: 3.03 MB
创建日期: 2019/1/18 16:04

图 5-18 拷贝

6. 完整程序清单

LUA 教程-运算和字符处理 demo 的完整程序，如程序清单 5 所示，查看程序时请结合教程 demo 理解：

程序清单 5 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:                  运算以及字符串处理说明
**
**--------------------------------------------------------------------------
** Created By:                  林青田
** Created date:                 2019/1/24
** Version:                      V1.0
** Descriptions:                运算以及字符串处理说明
**
**--------------------------------------------------------------------------
** Modified by:
** Modified date:
** Version:
** Description:
**
*************************************************************************--]]

Local    test_value = 0

--[[*************************************************************************
** 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)
 --调用函数：get_value(screen,control)
      --Input value :     screen    画面 ID
 --                        control  控件 ID
 --                        value     控件值
 --*****************************************************
 if screen == 1 then
      if control == 13 and value == 1 then                          --按下按钮显示运算结果
            local A = get_value(1,1)
            local B = get_value(1,2)
            set_value(1,3,A+B)                                      --显示 A+B 的结果
            set_value(1,4,A-B)                                      --显示 A-B 的结果
             set_value(1,5,A*B)                                     --显示 A*B 的结果
             set_value(1,6,A/B)                                     --显示 A/B 的结果
             if A < B    then                                       --显示 A 与 B 的比较
                   set_value(1,7,1)
             else
                  set_value(1,7,0)
               end
```

```
            if A > B    then
                set_value(1,8,1)
            else
                set_value(1,8,0)
            end
            if A <= B    then
                set_value(1,9,1)
            else
                set_value(1,9,0)
            end
            if A >= B    then
                set_value(1,10,1)
            else
                set_value(1,10,0)
            end
            if A == B    then
                set_value(1,11,1)
            else
                set_value(1,11,0)
            end
            if A ~= B    then
                set_value(1,12,1)
            else
                set_value(1,12,0)
            end
        end
    end
--*****************************************************
--功能：文本控件画面读写控件件值
--调用函数：set_value(screen,control,value)
--          set_text(screen,control,value)
--          get_value(screen,control)
--          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 == 2 then
    if control == 12 and value == 1 then                    --按下按钮 12，运行函数
        local value = 0
        local value2 = 0
        --*********************************************************
        --功能：开方
        --调用函数：math.sqrt(num)
```

```
--参数 :      num   数值
--*********************************************************
      value = math.sqrt(16)
      set_value(2,2,value)
--*********************************************************
--功能：取模运算（类求 x/y 余）
--调用函数：math.fmod(x，y)
--参数 :      x   数值
--参数 :      y   数值
--*********************************************************
      value = math.fmod(14,5)
      set_value(2,3,value)
--*********************************************************
--功能：取绝对值
--调用函数：math.abs(x)
--参数 :      x   数值
--*********************************************************
      value = math.abs(-16)
      set_value(2,4,value)
--*********************************************************
--功能：获取数值的整数部分及小数部分
--调用函数：math.modf(x)
--参数 :      x   数值
--*********************************************************
      value,value2 = math.modf(15.98)
      set_value(2,5,value)
      set_value(2,13,value2)
--*********************************************************
--功能：余弦函数
--调用函数：math.cos(x)
--参数 :      x                  弧度
--*********************************************************
      value = math.cos(0.5)
      set_value(2,6,value)
--*********************************************************
--功能：正弦函数
--调用函数：math.sin(x)
--              math.rad(30)    角度转换弧度
--参数 :      30                 角度
--*********************************************************
      value = math.sin(math.rad(30))
      set_value(2,7,value)
--*********************************************************
--功能：正切函数
```

```
        --调用函数：math.tan(x)
        --参数 ：      x   弧度
        --*********************************************************
              value = math.tan(0.5)
              set_value(2,8,value)
        --*********************************************************
        --功能：筛选出输入的参数中最大的参数
        --调用函数：math.max(arg...)
        --参数 ：      arg   数值（可多个参数，参数数量不定）
        --*********************************************************
              value = math.max(2.71,100,-98,23)
              set_value(2,9,value)
        --*********************************************************
        --功能：筛选出输入的参数中最小的参数
        --调用函数：math.min(arg...)
        --参数 ：      arg   数值（可多个参数，参数数量不定）
        --*********************************************************
              value = math.min(2.71,100,-98,23)
              set_value(2,10,value)


      end
end
--*********************************************************
--功能：简单的字符串处理
--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 == 12 and value == 1 then
              local str_value = 0
              local str_value2 = 0
              str_value2   = get_text(3,2)
              str_value = get_value(3,1)
              set_text(3,3,str_value..str_value2)                    --字符串拼接
--*********************************************************
--功能：尝试将参数 e 转换为数字，当不能转换时返回 nil
--调用函数：tonumber (e [, base])
    --参数 ：      e   字符串
--              [，base]（可选参数，不填时默认转十进制数值，填 16 转十六进制进制数）
--*********************************************************
           if tonumber(str_value2) == nil then                       --字符串转数值
```

```
                set_value(3,4,0)
            else
                set_value(3,4,tonumber(str_value2))
            end
--*********************************************************
--功能：尝试将参数 e 转换为字符串，当不能转换时返回 nil
--调用函数：tonumber (e)
    --参数 :     e   数值
--*********************************************************
            set_text(3,5,tostring(str_value))                    --数值转字符串
--*********************************************************
--功能：在文本框中显示获取字符串长度
--调用函数：tonumber (str)
    --参数 :     str     字符串
--*********************************************************
            set_value(3,6,string.len(str_value2))                --string.len 获取字符串的长度
--*********************************************************
--功能：在文本框中显示截取字的符串 str_value 的第一个字符
--调用函数：string.sub(str,i,j)
    --参数 :     str   字符串
--              i       起始索引
--              j       截止索引
--*********************************************************
            set_text(3,7,string.sub(str_value2,1,2))             --string.sub 获取字符串的长度
      end
end
--*********************************************************
--功能：达到最大值后弹出提示框。提示已到达最大值
--*********************************************************
if screen == 4 then
--*********************************************************
--功能：按一下按钮计数值减一
--*********************************************************
      if control == 2 and value == 1 then
            test_value = test_value - 1
            if test_value < 0 then
                test_value = 0
            end
            set_value(4,3,test_value)

      end
--*********************************************************
--功能：按一下按钮计数值加一
--*********************************************************
```

```
        if control == 4 and value == 1 then
            test_value = test_value + 1
            if test_value > 10 then
                change_screen(5)
                test_value = 10
            end
            set_value(4,3,test_value)
        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所有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，若在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。