Technical Note

LUA 例程-绘图 API 函数使用说明 V1.0

工程技术笔记

类别	内容
关键词	绘图 API 函数
摘要	

修订历史

版本	日期	原因	编制	审查
V1.0	2019/02/27	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

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。

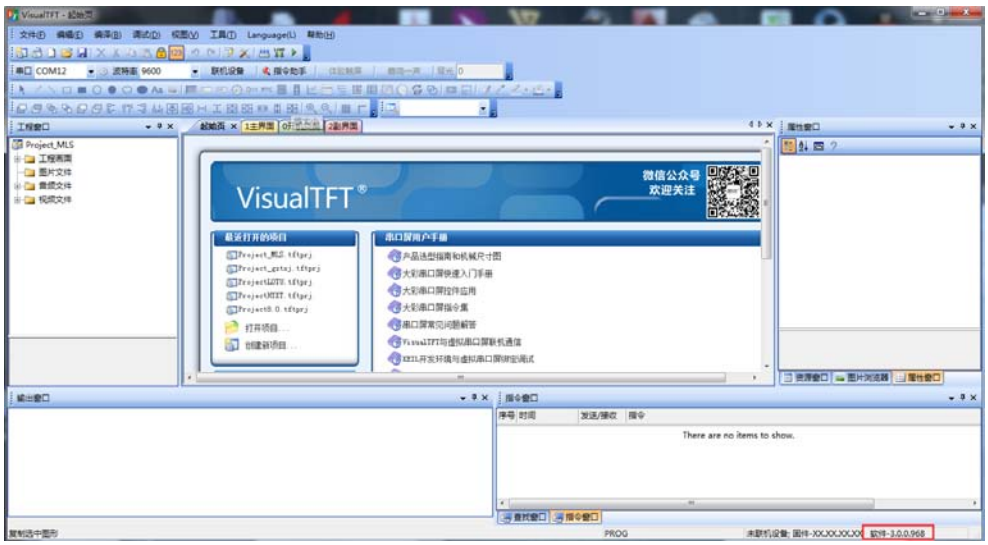

图 2-2 软件版本

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。

本文将介绍大彩 LUA 脚本 API 函数中的绘图 API 函数使用方法，以及使用函数的注意事项。

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 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

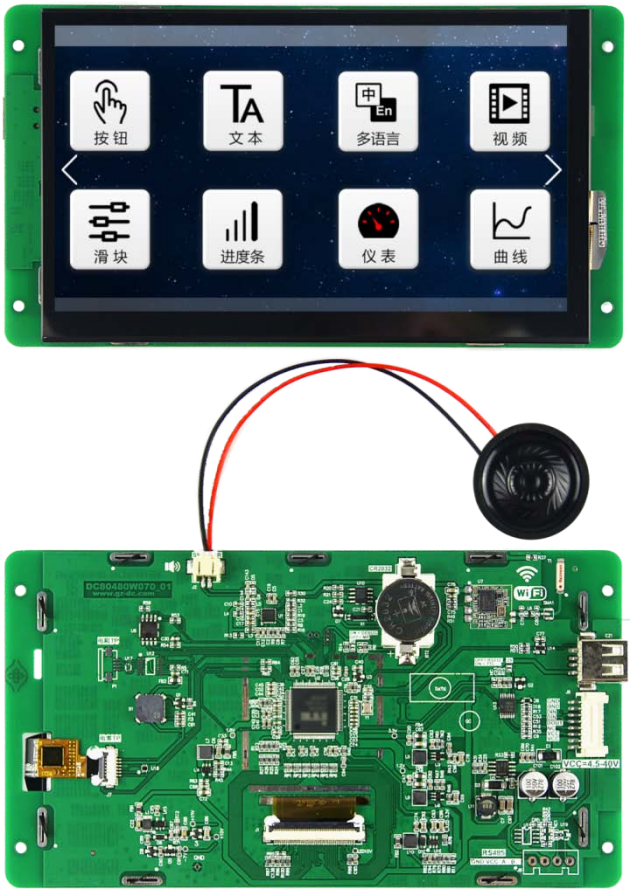

图 5-1 物联型 7 寸串口屏

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。

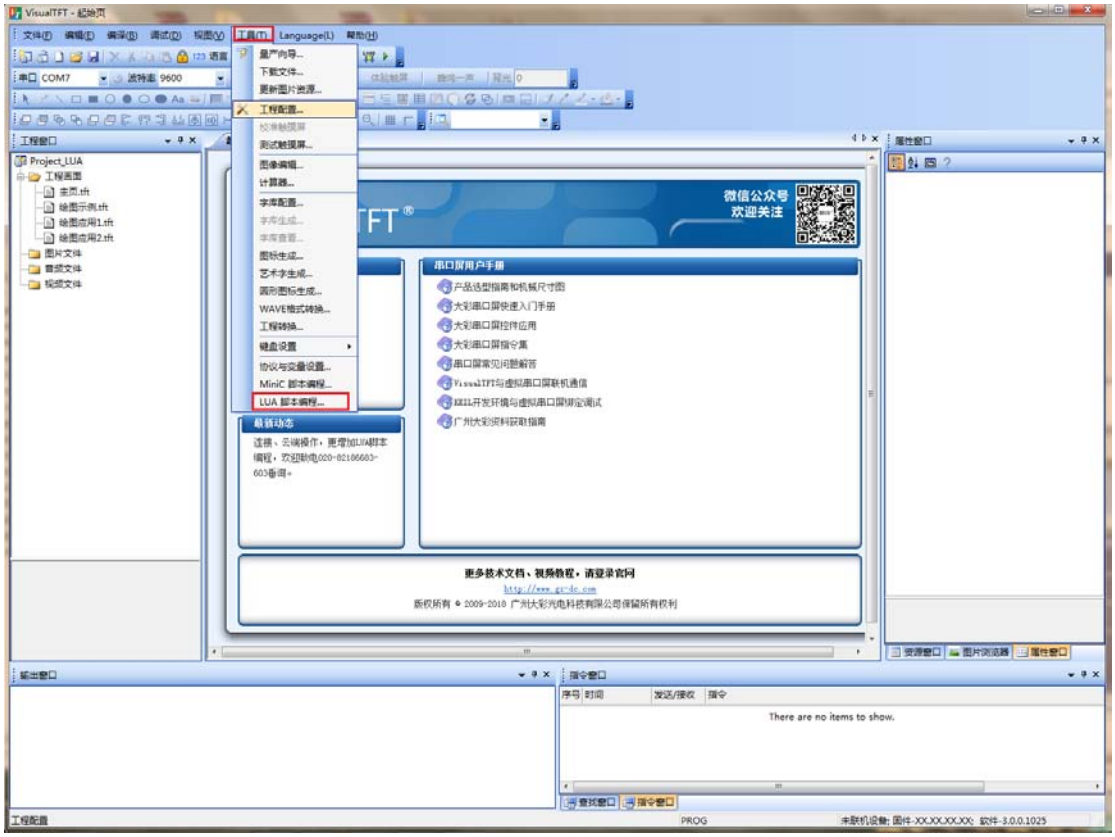

图 5-3 打开 LUA 编辑器

打开编辑器后，Visual TFT 画面如图 5-4 所示；

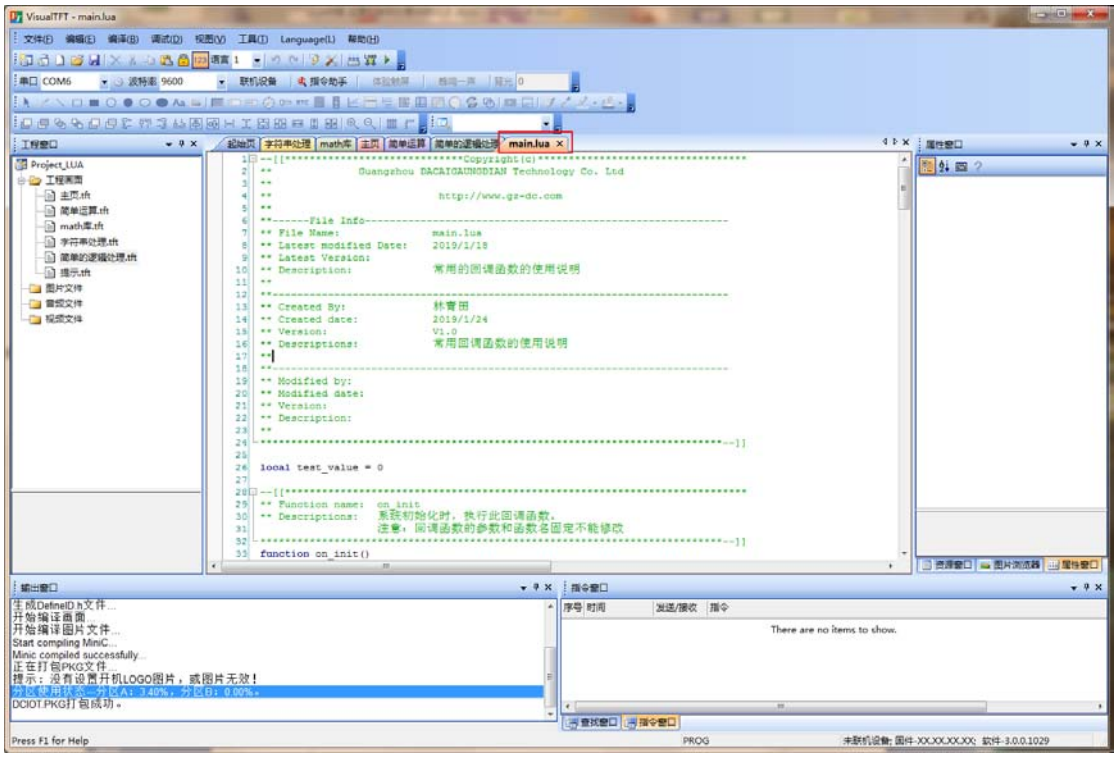

图 5-4 LUA 编辑器

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。

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 redraw()

注释：申请调用回调函数 on_draw；

2. 函数 start_timer(timer_id, timeout, countdown, repeat)

注释：打开定时器。

参数： timer_id 定时器 ID
timeout 超时时间
countdown 1 或 0 顺计时 1，倒计时 0
repeat 计时器重复次数 0 为无限循环

3. 函数 load_surface (filepath)

注释：加载图片。

参数 filepath 图片路径
返还值 surface 图层指针，指向所加载的图片

4. 函数 draw_line(x1,y1,x2,y2,level)

注释：画直线

参数：

x1	直线起始点 x 轴的坐标
y1	直线起始点 y 轴的坐标
x2	直线结束点 X 轴的坐标
y2	直线结束点 y 轴的坐标
level	直线的曲线等级，1~10 级

5. 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

6. 函数 on_draw(screen)

注释：当调用 redraw（）或屏幕刷新时会触发此函数，在目标画面中进行绘图

参数：screen 表示目标画面 ID

所有的绘图 API 函数需要写在回调函数 on_draw（）中才能生效，写在该函数外的绘图 API 函数均无效，但函数 redraw（）除外，函数 redraw（）为申请重新调用回调函数 on_draw。

5.3 教程实现

串口屏提供 API 函数绘制的基本图形共有 7 种，分别为直线、矩形、实心矩形、圆形、实心圆形、椭圆形和实心椭圆，另外还提供绘制文字和图片的 API 函数。本章节将结合教程 DEMO 讲述如何调用 API 函数实现绘制基本图形和绘制图片，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 绘制基本图形 API

1. 在教程 DEMO 的绘图示例画面中，点击按钮可以实现绘制指定的基本图形，运行虚拟屏后工程画面如图 5-6 所示

图 5-6 绘制基本图形画面

2. LUA 程序解释：将 7 种图形按按钮控件编号顺序定义，如 1 对应直线、2 对应空心矩形依次类推。按下对应的按钮后获取对应的序号，程序如程序清单 1 所示；

程序清单 1 获取绘图类型

```
function on_control_notify(screen,control,value)
 --*****************************************************
 --功能：     按照定义的顺序选取对应的图形
 --调用函数：redraw()                    重新绘图
 --*****************************************************
 if screen == 1 then
        if (control >= 10 and control <= 16) and value == 1 then          --按下画对应的图形
            draw_type = control                                           --获取按钮编号
            redraw()
        end
        if control == 9 and value == 1 then                               --按下设置画笔颜色
            draw_pen_color = 0xffe0
            redraw()
        end
 end
 …….
end
```

获取到对应的编号后调用 API 函数在当前画面绘制图形，显示图形部分的程序清单 2 中所示。

程序清单 2 绘制基本图形

```
function on_draw(screen)
     set_pen_color(draw_pen_color)
     if screen == 1   then                          --当前画面 ID
 --*********************************************************************
 --调用函数：draw_line(x1,y1,x2,y2,level)
     --函数功能：     两点间画直线
 --参数：
 --        x1              直线起始点 x 轴的坐标
 --        y1              直线起始点 y 轴的坐标
 --        x2              直线结束点 X 轴的坐标
 --        y2              直线结束点 y 轴的坐标
 --        level           直线的曲线等级，1~10 级
 --*********************************************************************
      if draw_type == 1 then
          draw_line(400,250,550,250,2)
 --*********************************************************************
 --调用函数：draw_rect(x1,y1,x2,y2,fill)
--函数功能：     画矩形
 --参数：
```

```
--          x1              矩形左上脚 x 轴的坐标
--          y1              矩形左上脚 y 轴的坐标
--          x2              矩形右下脚 X 轴的坐标
--          y2              矩形右下角 y 轴的坐标
--          fill            填充 1，不填充 0
--**************************************************************************
        elseif draw_type == 2 then
            draw_rect(400,200,600,300,0)
        elseif draw_type == 3 then
            draw_rect(400,200,600,300,1)
 --**************************************************************************
--调用函数：draw_circle(x1,y1,r,fill)
    --函数功能：      画圆
--参数：
--          x1              直线起始点 x 轴的坐标
--          y1              直线起始点 y 轴的坐标
--          x2              直线结束点 X 轴的坐标
--          r               直线结束点 y 轴的坐标
--          fill            填充 1，不填充 0
--**************************************************************************
        elseif draw_type == 4 then
            draw_circle(500,250,50,0)
        elseif draw_type == 5 then
            draw_circle(500,250,50,1)
      --**************************************************************************
--调用函数：draw_ellipse(x1,y1,x2,y2,fill)
    --函数功能：      画椭圆
--参数：
--          x1              左上角的 x 坐标
--          y1              左上角的 y 坐标
--          x2              右下角的 x 坐标
--          y2              右下角的 y 坐标
--          fill            填充 1，不填充 0
--**************************************************************************
        elseif draw_type == 6 then
            draw_ellipse(400,200,600,300,0)
        elseif draw_type == 7 then
            draw_ellipse(400,200,600,300,1)
        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

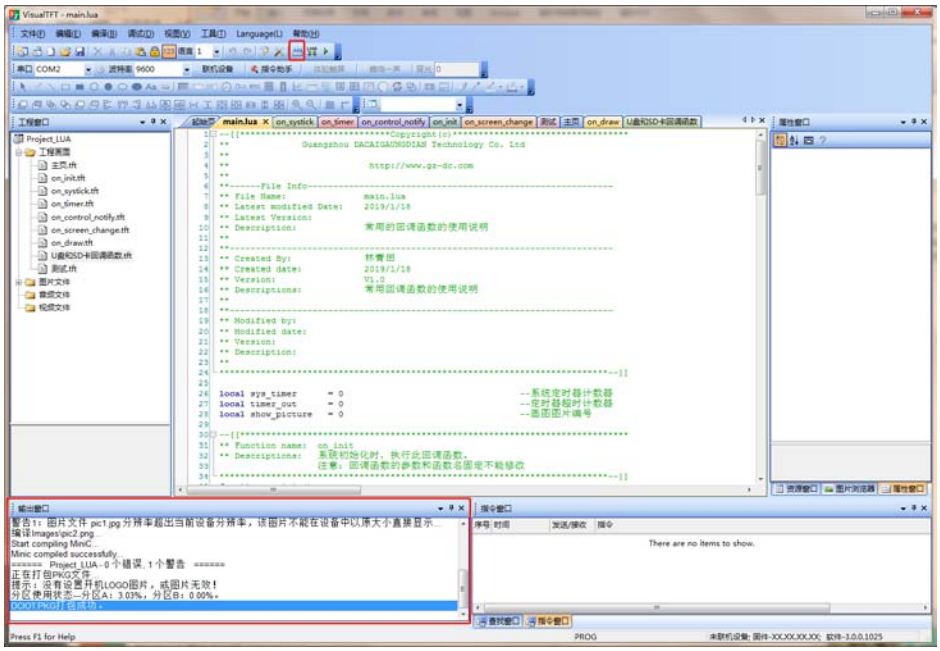

图 5-7 编译工程

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，如图 5-8 所示

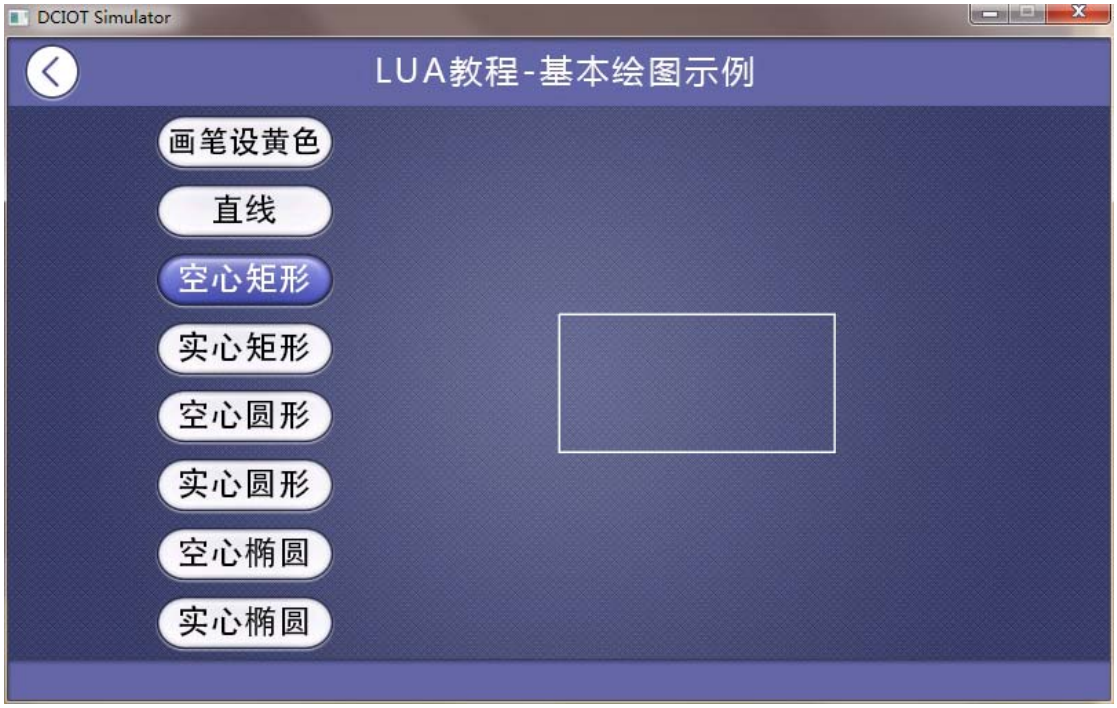

图 5-8 基本绘图示例

5.3.2 绘图应用 1

1. 利用基本图形也可以仿制部分控件的使用，如利用画直线模仿曲线控件和画实心矩形模仿进度条控件，教程 DEMO 中先将曲线坐标轴和空心矩形画在背景图中，然

后利用坐标在对应的位置上画直线和实心矩形，教程 DEMO 画面如图 5-9 所示；

图 5-9 绘图应用 1

2. LUA 程序解释：教程中的曲线数据为固定数据，进入到该画面后按照固定数据显示曲线；显示曲线部分的代码如程序清单 3 所示；

程序清单 3 绘制曲线

```
local    graph_x = {70,120,160,200,240,280}                    --曲线数据
local    graph_y = {300,300,260,260,300,300}                   --曲线数据
function on_draw(screen)
……
      if screen == 2    then
         draw_rect(460,320,rectangle_x,rectangle_y,1)          --进度条
          for i = 1 ,5 do
               draw_line(graph_x[i],graph_y[i],graph_x[i+1],graph_y[i+1],1)      --曲线
         end
      end
……
end
```

按下画面中开始按钮后启动定时器 2 定时刷新进度条的进度。代码如程序清单 4 中所示

程序清单 4 定时刷新进度条

```
--*****************************************************
--功能：      打开定时器，定时绘图
--调用函数：start_timer(timer_id, timeout, countdown, repeat)      打开定时器
    --参数 ：      timer_id                    定时器 ID
```

```
--                  timeout                     定时时间
--                  countdown                   1 或 0   顺计时  1 ，倒计时  0
--                  repeat                      计时器重复次数     0 为无限循环
--调用函数：stop_timer(timer_id)              停止定时器
--调用函数：redraw()                          重新绘图
--*****************************************************
if screen == 2 then
    if control == 8 and value == 1 then                                    --按下启动定时器 2
        start_timer(2,300,1,0)
    elseif control == 8 and value == 0 then
        stop_timer(2)                                                      --按下按钮停止定时器 2
    end
end
function on_draw(screen)
……
    if screen == 2   then
      draw_rect(460,320,rectangle_x,rectangle_y,1)                         --进度条
      for i = 1 ,5 do
          draw_line(graph_x[i],graph_y[i],graph_x[i+1],graph_y[i+1],1)     --曲线
      end
    end
……
End
```

工程编译请参考上一小节的第 2 步；

3. 点击运行串口屏后切换到当前画面，画面中已经在指定的坐标把曲线画出来，点击画面中开始按钮后刷新进度条的值，如图 5-10 所示；

图 5-10 绘图应用 1

5.3.3 绘图应用 2

1. 该画面为使用 LUA API 函数显示图片的教程，在教程画面中的按钮代表对应的显示图片的 API 函数，空心矩形框的位置是图片显示的位置。教程演示画面如图 5-11 所示；

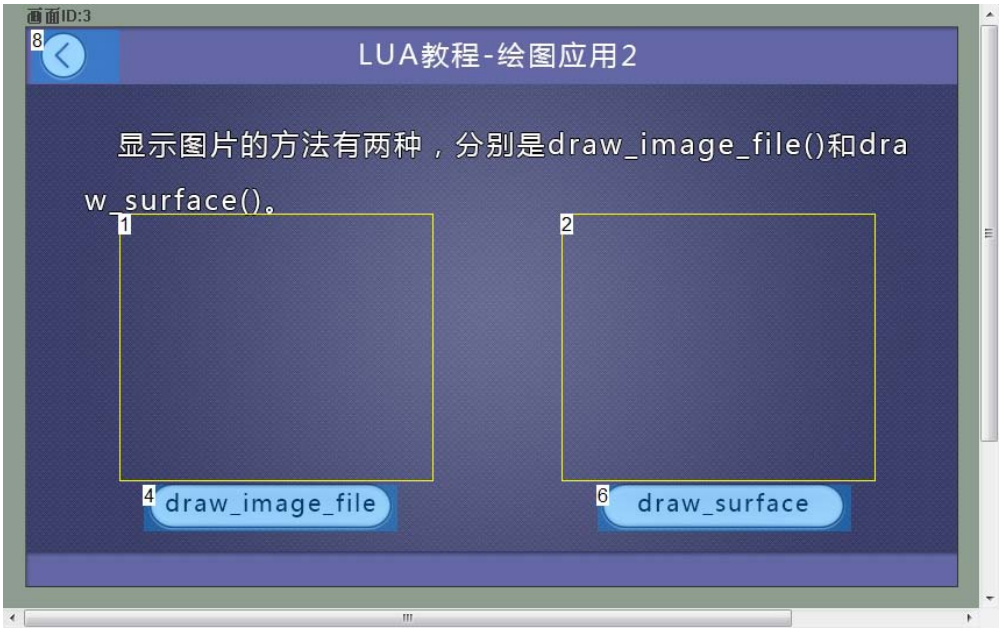

图 5-11 绘图应用

2. 按下画面中的按钮，分别调用两种 API 函数显示指定的图片，显示图片的程序如程序清单 5 中所示。

程序清单 5 显示图片

```
if    screen == 3 and control == 4 and value == 1 then
        redraw()
        draw_pic1 = 1
 end
 if    screen == 3 and control == 6 and value == 1 then
--*****************************************************
--功能：      销毁图层指针
--调用函数：destroy_surface(surface)     销毁图层指针
    --参数 ：      surface              图层指针
--调用函数：surface = load_surface (filepath)     加载图片到图层
--参数          filepath                    图片路径
--返还值       surface                     图层指针        指向所加载的图片
--调用函数：redraw()                       重新绘图
--*****************************************************
 if    surface ~= 0 then
        destroy_surface(surface)
        surface = 0
 end
        surface = load_surface ("A:/0.jpg")
 end
```

```
function    on_draw(screen)
    if    screen == 3    then
        if    draw_pic1 == 1    then
            draw_image_file("a:/1.png",80,160,270,230,0,100)        --显示内部存储 a 区中的 1.png 图片
        end
        draw_surface(surface,460,160,270,230,0,0)                   --显示内部存储 a 区中的 0.jpg 图片
    end
end
```

工程编译请参考上一小节的第 2 步；

3. 该画面需要在实体屏中，才可以将图片显示，下载工程方法请参考 5.4。

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 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。

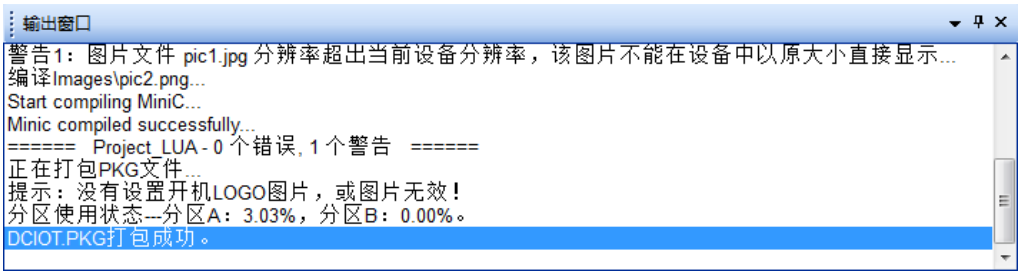

图 5-12 编译成功

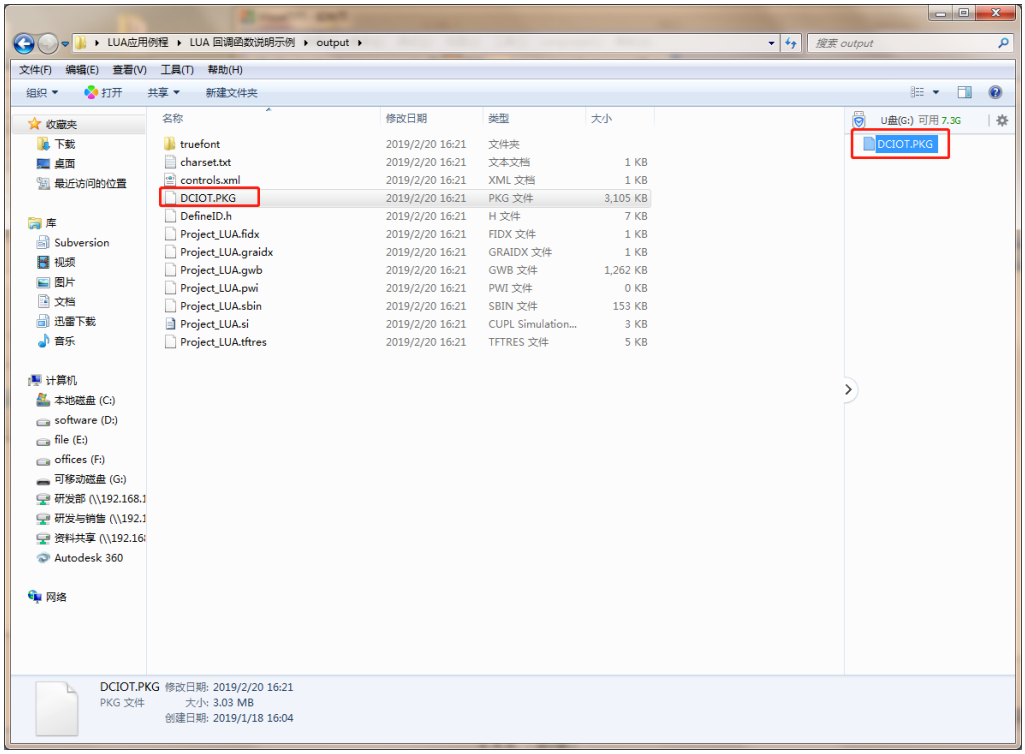
LUA应用例程 ▸ LUA 回调函数说明示例 ▸ output ▸
搜索 output
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 打开 共享 新建文件夹
名称 修改日期 类型 大小
truefont 2019/2/20 16:21 文件夹
charset.txt 2019/2/20 16:21 文本文档 1 KB
controls.xml 2019/2/20 16:21 XML 文档 1 KB
DCIOT.PKG 2019/2/20 16:21 PKG 文件 3,105 KB
DefineID.h 2019/2/20 16:21 H 文件 7 KB
Project_LUA.fidx 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 5-13 拷贝

6. 完整程序清单

LUA 教程-绘图 API 函数说明 demo 的完整程序，如程序清单 6 所示，查看程序时请结合教程 demo 理解：

程序清单 6 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:     2019/2/22
** Latest Version:
** Description:                绘图 API 使用教程说明
**
**--------------------------------------------------------------------------
** Created By:                 林青田
** Created date:             2019/2/22
** Version:                    V1.0
** Descriptions:             绘图 API 使用教程说明
**
**--------------------------------------------------------------------------
** Modified by:
** Modified date:
** Version:
** Description:
**
************************************************************************--]]
local draw_type = 0                                  --画的类型
local draw_pen_color = 0xFFFF                        --画笔颜色为白色，RGB565
local graph_x = {70,120,160,200,240,280}             --曲线数据
local graph_y = {300,300,260,260,300,300}            --曲线数据


local progress = 0                                   --进度条百分比
local rectangle_x = 460                              --进度条起始 x
local rectangle_y = 354                              --进度条起始 y
local draw_pic1 = 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)
        if    timer_id == 2 then
            rectangle_x = rectangle_x +30
            progress = progress + 10
            if    rectangle_x > 760 then
                rectangle_x = 460
                progress = 0
            end
            set_value(2,3,progress)
        end
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：       选取对应的图形
 --调用函数：redraw()                    重新绘图
 --*****************************************************
 if screen == 1 then
```

```
    if (control >= 10 and control <= 16) and    value    == 1 then                    --按下画对应的图形
        draw_type = control
        redraw()
    end
    if control == 9 and value == 1 then                                              --按下设置画笔颜色
        draw_pen_color = 0xffe0
        redraw()
    end
end
--*****************************************************
--功能：      打开定时器，定时绘图
--调用函数：start_timer(timer_id, timeout, countdown, repeat)     打开定时器
    --参数 ：      timer_id              画面 ID
--              timeout               超时时间
--                countdown           1 或 0   顺计时  1 ，倒计时  0
--                repeat              计时器重复次数     0 为无限循环
--调用函数：stop_timer(timer_id)     停止定时器
--调用函数：redraw()                   重新绘图
--*****************************************************
if   screen == 2 then
    if control == 8 and value == 1 then                                              --按下启动定时器 2
        start_timer(2,300,1,0)
    elseif control == 8 and value == 0 then
        stop_timer(2)                                                                --按下按钮停止定时器 2
    end
end
if   screen == 3 and control == 6 and value == 1 then
--*****************************************************
--功能：      销毁图层指针
--调用函数：destroy_surface(surface)     销毁图层指针
    --参数 ：      surface             图层指针
--调用函数：surface = load_surface (filepath)     加载图片到图层
--参数          filepath                  图片路径
--返还值       surface                    图层指针        指向所加载的图片
--调用函数：redraw()                   重新绘图
--*****************************************************
    if surface ~= 0 then
        destroy_surface(surface)
        surface = 0
    end
    surface = load_surface ("A:/0.jpg")
end
 if   screen == 3 and control == 4 and value == 1 then
    redraw()
```

```
        draw_pic1 = 1
    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
--[[*********************************************************************************
** Function name:    on_draw
** Descriptions:     画面刷新时，或者使用 API 函数 redraw 申请重绘，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_draw(screen)
    set_pen_color(draw_pen_color)
    if screen == 1    then
    --*************************************************************************
 --调用函数：draw_line(x1,y1,x2,y2,level)
    --函数功能：       两点间画直线
 --参数：
 --          x1                直线起始点 x 轴的坐标
 --          y1                直线起始点 y 轴的坐标
 --          x2                直线结束点 X 轴的坐标
 --          y2                直线结束点 y 轴的坐标
 --          level             直线的曲线等级，1~10 级
 --*************************************************************************
      if draw_type == 1 then
            draw_line(400,250,550,250,2)
    --*************************************************************************
 --调用函数：draw_rect(x1,y1,x2,y2,fill)
    --函数功能：       画矩形
 --参数：
 --          x1                矩形左上脚 x 轴的坐标
 --          y1                矩形左上脚 y 轴的坐标
 --          x2                矩形右下脚 X 轴的坐标
 --          y2                矩形右下角 y 轴的坐标
 --          fill              填充 1，不填充 0
 --*************************************************************************
       elseif draw_type == 2 then
```

```
            draw_rect(400,200,600,300,0)
        elseif draw_type == 3 then
            draw_rect(400,200,600,300,1)
    --*************************************************************************
--调用函数：draw_circle(x1,y1,r,fill)
    --函数功能：      画圆
--参数：
--          x1              直线起始点 x 轴的坐标
--          y1              直线起始点 y 轴的坐标
--          x2              直线结束点 X 轴的坐标
--          r               直线结束点 y 轴的坐标
--          fill            填充 1，不填充 0
--*************************************************************************
        elseif draw_type == 4 then
            draw_circle(500,250,50,0)
        elseif draw_type == 5 then
            draw_circle(500,250,50,1)
    --*************************************************************************
--调用函数：draw_ellipse(x1,y1,x2,y2,fill)
 --函数功能：      画椭圆
--参数：
--          x1              左上角的 x 坐标
--          y1              左上角的 y 坐标
--          x2              右下角的 x 坐标
--          y2              右下角的 y 坐标
--          fill            填充 1，不填充 0
--*************************************************************************
        elseif draw_type == 6 then
            draw_ellipse(400,200,600,300,0)
        elseif draw_type == 7 then
            draw_ellipse(400,200,600,300,1)
        end
      end
      if   screen == 2   then
        draw_rect(460,320,rectangle_x,rectangle_y,1)                    --进度条
        for i = 1 ,5 do
            draw_line(graph_x[i],graph_y[i],graph_x[i+1],graph_y[i+1],1)    --曲线
        end
end
 if screen == 3   then
        if draw_pic1 == 1 then
            draw_image_file("a:/1.png",80,160,270,230,0,100)        --显示内部存储 a 区中的 1.png 图片
        end
       draw_surface(surface,460,160,270,230,0,0)                    --显示内部存储 a 区中的 0.jpg 图片
```

```
 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。