

类别	内容
关键词	LUA、串口、自定义指令
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.....	6
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    设置自定义串口.....	9
5.3.2    设置按钮状态.....	10
5.3.3    设置文本内容.....	11
5.3.4    设置蜂鸣器响.....	12
5.3.5    发送按钮指令.....	12
5.3.6    发送键盘输入内容.....	13
5.4    下载工程.....	14
5.4.1    下载.....	14
6. 完整程序清单.....	16
7. 免责声明.....	21



## 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。

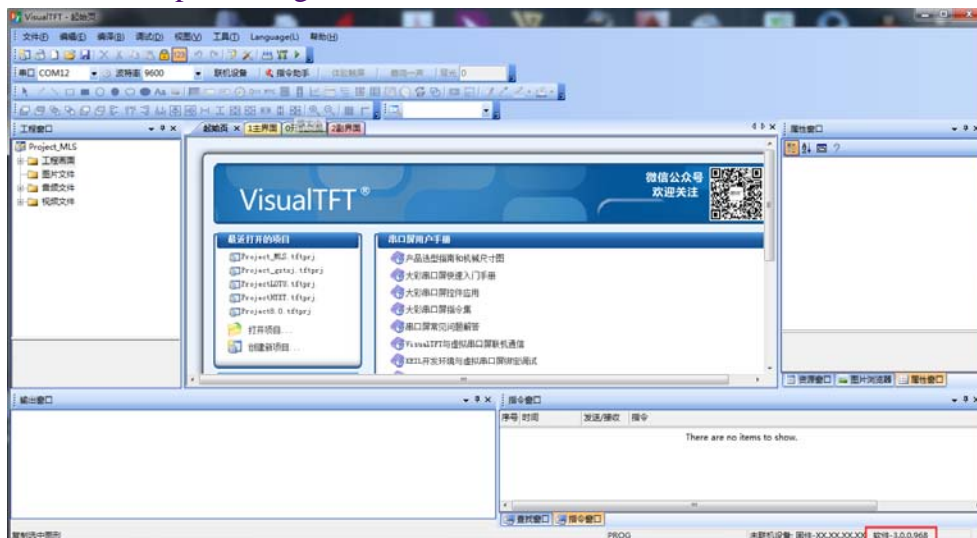


图 2-2 软件版本

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。

本文将介绍通过 LUA 脚本的自定义串口指令设置按钮按下、设置文本、设置蜂鸣器响。并在按下按钮或通过键盘输入数据后发送自定义指令。

## 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:  
<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

本章节主要讲述通过 LUA 实现自定义串口指令设置按钮按下、设置文本、设置蜂鸣器响。并在按下按钮或通过键盘输入数据后发送自定义指令。本文将分为以下是 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。

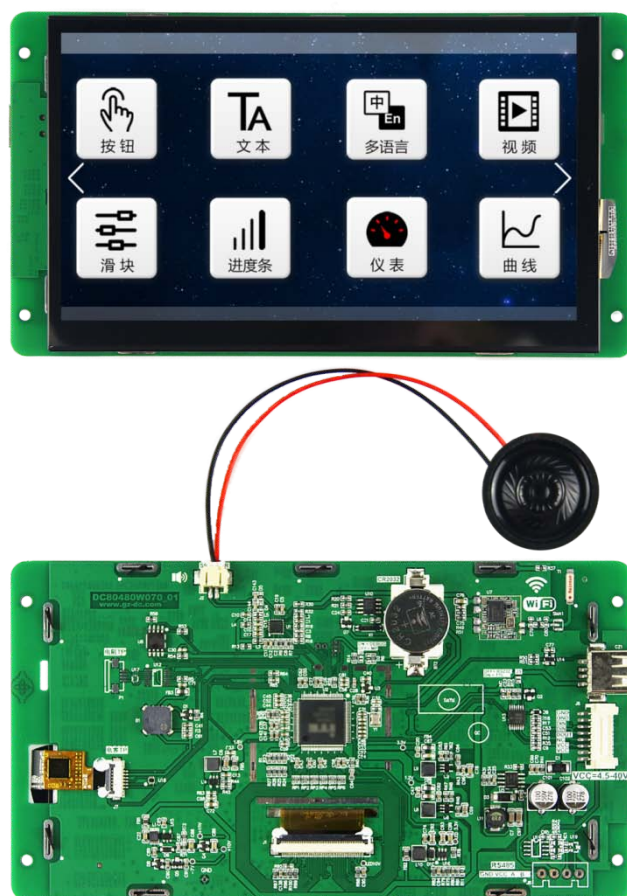


图 5-1 物联型 7 寸

### 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。

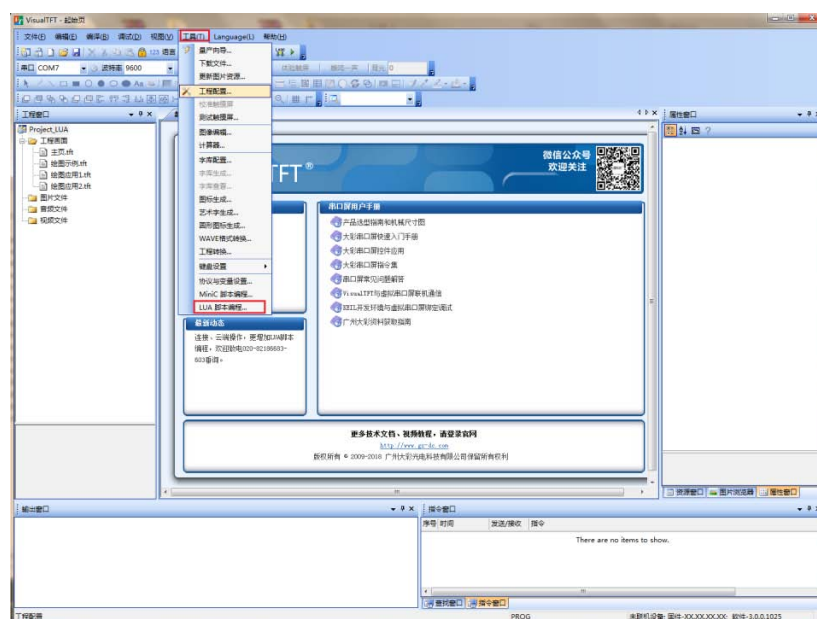


图 5-3 打开 LUA 编辑器

LUA 编辑器的开发环境，如图 5-4 所示。



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 on\_uart\_recv\_data(packet), baudrate 表示设置的波特率值。

注释：此函数可以在用户触摸修改控件后被回调

#### 3. 函数 uart\_set\_timeout(timeout, timeout\_inter)

注释：此函数可以在用户触摸修改控件后被回调。函数有两个参数：

timeout 表示接收总超时

timeout\_inter 表示字节间隔超时

#### 4. 函数 beep(time), time 表示蜂鸣器响的时间长度，单位 ms。

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

注释：此函数可以设置控件数值。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。

#### 6. 函数 set\_text(screen, control, text)

注释：此函数设置文本控件的内容。函数有三个参数：

screen 表示被触发控件的页码

control 表示控件的编号

text 表示要设置的内容

#### 7. 函数 uart\_send\_data(packet), packet 表示数据，数据类型：表。

注释：此函数通过串口发送数据。

#### 8. 函数 get\_text(screen, control)

注释：此函数获取文本控件的内容。函数有两个参数：

screen 表示被触发控件的页码

control 表示控件的编号

返回文本的数据

#### 9. 函数 string.len(arg), arg 表示字符串。

注释：计算字符串长度。

返回文本的长度值

#### 10. 函数 string.byte(arg[, int])

注释：转换字符为整数值(可以指定某个字符，默认第一个字符)。

arg 表示字符串

[,int] 表示索引

返回字符对应整数值

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

串口屏中使用自定义串口指令，通过发送自定义的指令可以设置按钮按下、设置文本、设置蜂鸣器响。并在按下按钮或通过键盘输入数据后发送自定义指令。教程中每个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 设置自定义串口

首先因为在实体屏中数据传输需要时间，所以在 LUA 脚本中设置串口接受超时时间，如程序清单 5-1。

程序清单 5-1 设置串口超时时间

```
function on_init()
    uart_set_timeout(1000,200)
end
```

然后，设置 uart\_free\_protocol 为 1 即进入自由串口协议，如程序清单 5-2。

程序清单 5-2 设置自由串口协议

```
uart_free_protocol=1
function on_uart_recv_data(packet)    --串口函数
```

end

把程序编写入软件的 LUA 编辑器中，如图 5-6 所示。

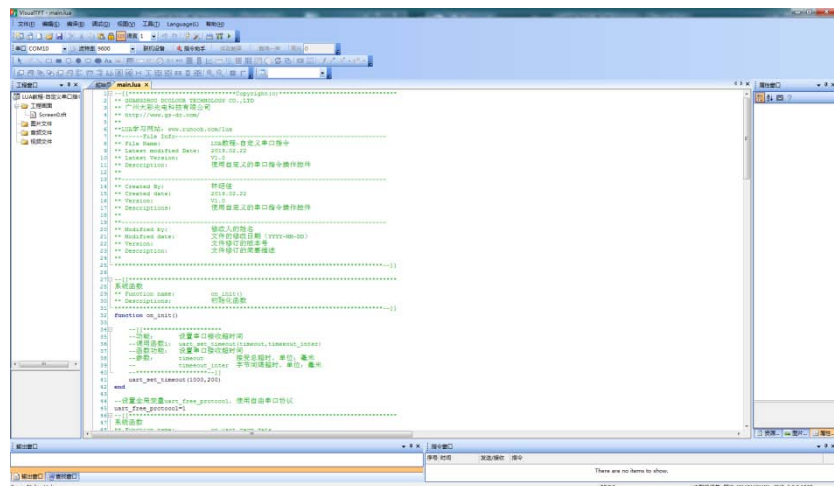


图 5-6 把程序写入 LUA 编辑器

### 5.3.2 设置按钮状态

1. 在工程中设置一个按钮，如需要在状态改变有效果，可以设置按下图片并选择裁剪，如图 5-6。

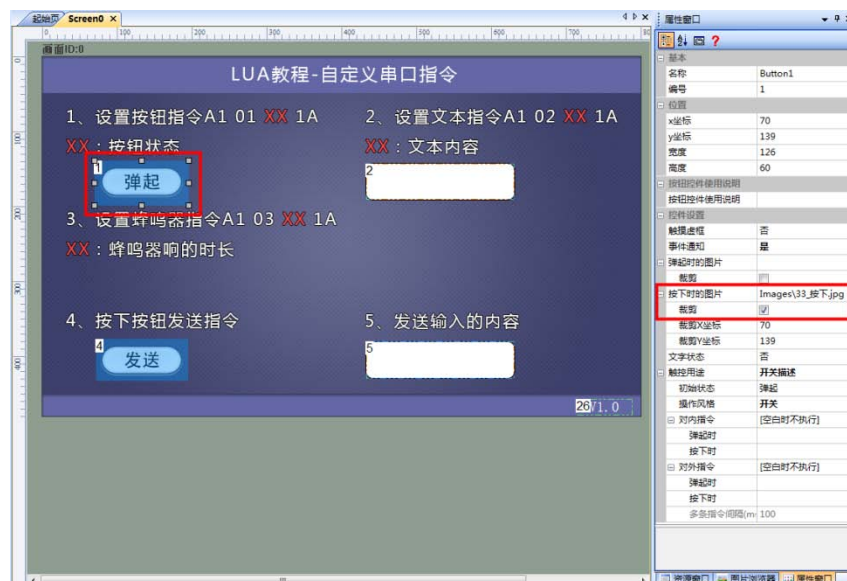


图 5-7 设置按钮

2. 然后在串口函数接收到 A1 01 01 1A 指令时使按钮按下，如程序清单 5-3。

程序清单 5-3 设置按钮状态

```
if packet[0]==0xA1 --帧头为 0xA1
then
    if packet[1]==0x01 --第二个字节对应按钮指令
    then
        if packet[2]==0x01 --第三个字节对应按钮按下的状态
```



```
then
    set_value(0,1,1)
else
    --第三个字节对应按钮弹起的状态
    set_value(0,1,0)
end
end
end
end
```

3. 虚拟运行，当接收到 A1 01 01 1A 指令时，按钮设置为按下状态，如图 5-7 所示。

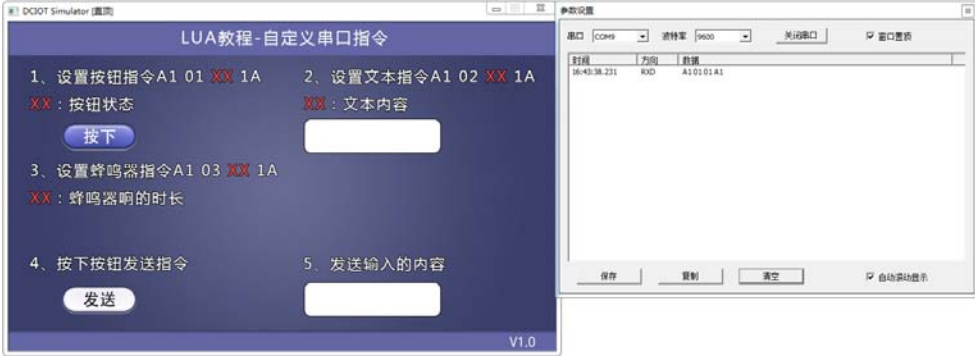


图 5-8 设置按钮按下

5.3.3 设置文本内容

1. 设置一个文本框，如图 5-8。

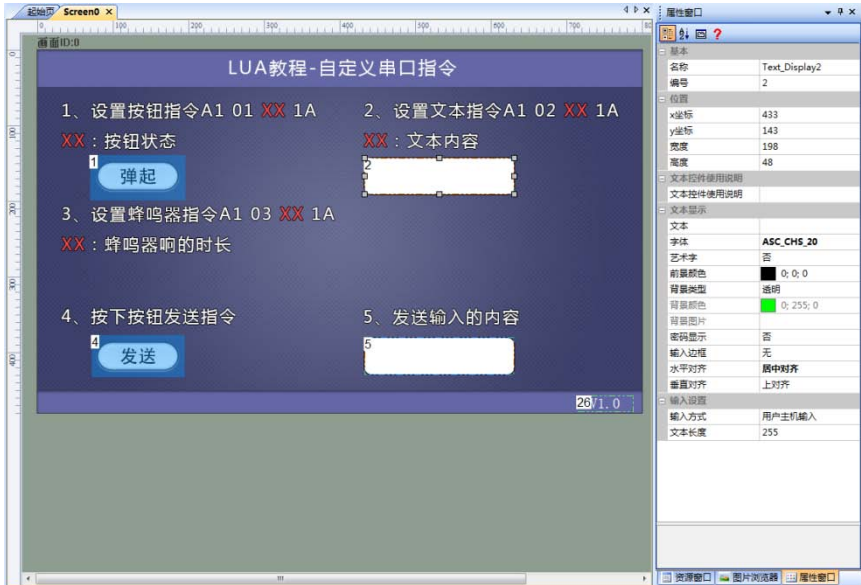


图 5-9 设置文本框

2. 然后在串口函数接收到 A1 02 XX 1A 指令时设置文本值为 XX，如程序清单 5-4。

程序清单 5-4 设置文本框的数值

```
if packet[0]==0xA1
then
    if packet[1]==0x02
    then
        --帧头为 0xA1
        --第二个字节对应文本指令
```

```

set_text(0,2,packet[2])    --设置文本为第三个字节
end
end
end

```

3. 虚拟运行，接收到指令时，设置文本框的值，如图 5-9 所示。

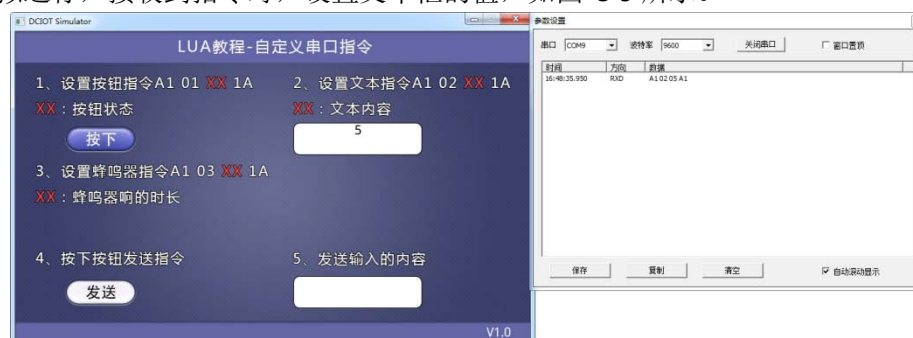


图 5-10 设置文本框的值

### 5.3.4 设置蜂鸣器响

1. 在串口函数接收到 A1 03 XX 1A 指令时设置响为 XX 毫秒，如程序清单 5-5。

程序清单 5-5 设置蜂鸣器响

```

if packet[0]==0xA1        --帧头为 0xA1
then
    if packet[1]==0x03    --第三个字节对应文本指令
    then
        beep(packet[2])   --设置蜂鸣器响，单位：毫米
    end
end
end

```

### 5.3.5 发送按钮指令

1. 在设置一个按钮，事件通知选择是，再设置按下图片并选择裁剪，操作风格选择瞬变，如图 5-10。

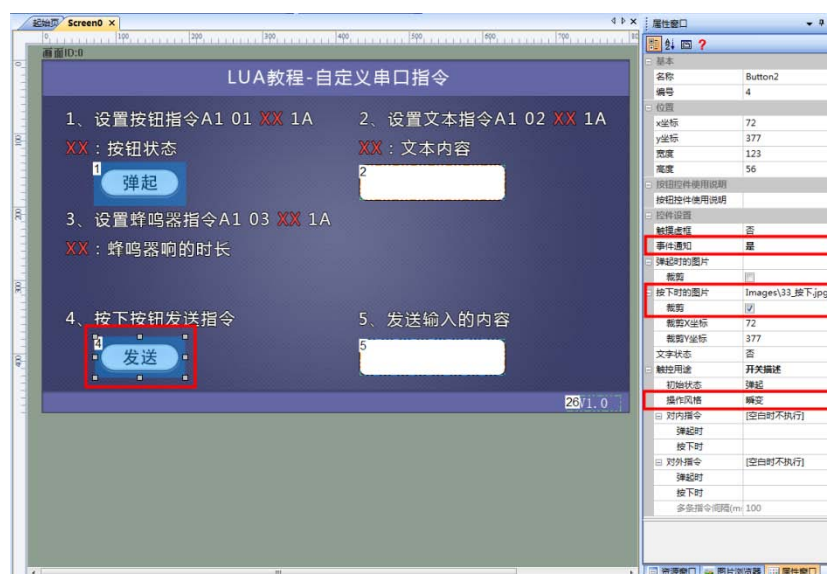


图 5-11 设置按钮



- 在触摸控件回调函数中设置按下按钮是发送指令 A1 04 01 1A，如程序清单 5-6。

程序清单 5-6 发送按钮指令

```
--设置按钮按下时发送的指令: A1 04 01 1A
local send_table1={04,01,26}
send_table1[0]=161
if screen==0 and control==4 and value==1
then
    uart_send_data(send_table1)    --发送指令
end
```

- 虚拟运行，按下按钮时，发送 A1 04 01 1A 指令，如图 5-11 所示。

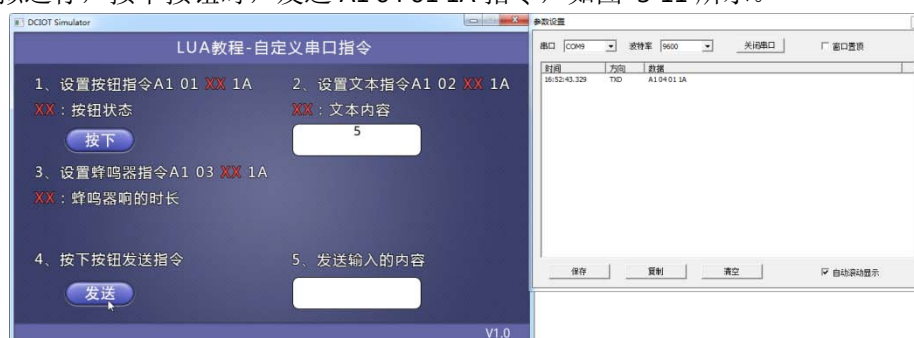


图 5-12 按下按钮发送指令

### 5.3.6 发送键盘输入内容

- 在设置一个文本框，文本框属性的键入方式选择弹出系统键盘输入，输入方式选择全键盘，如图 5-12 所示。

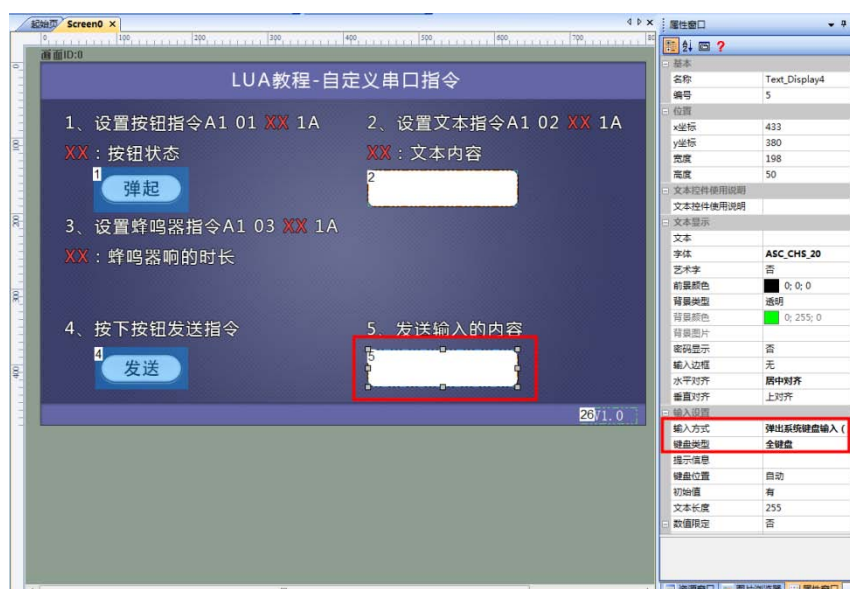


图 5-13 设置键盘输入

- 在触摸控件回调函数中设置键盘输入完成后是发送指令 A1 05 XX 1A，XX 表示键盘输入的内容，如程序清单 5-7 所示。

程序清单 5-7 发送键盘输入的内容

```

if screen==0 and control==5
then
    send_table2={}                --表初始化
    send_table2[0]=161            --设置指令头, 0xA1
    send_table2[1]=0x05           --设置键盘指令
    send_text2=get_text(0,5)      --获取文本的内容
    for i=1,string.len(send_text2) do    --把字符串转为表
        send_table2[i+1]=string.byte(send_text2,i)
    end
    send_table2[string.len(send_text2)+2]=26    --设置指令末, 0x1A
    uart_send_data(send_table2)            --发送指令
end
end

```

虚拟运行，键盘输入完成后发送键盘输入的内容，如图 5-13 所示。

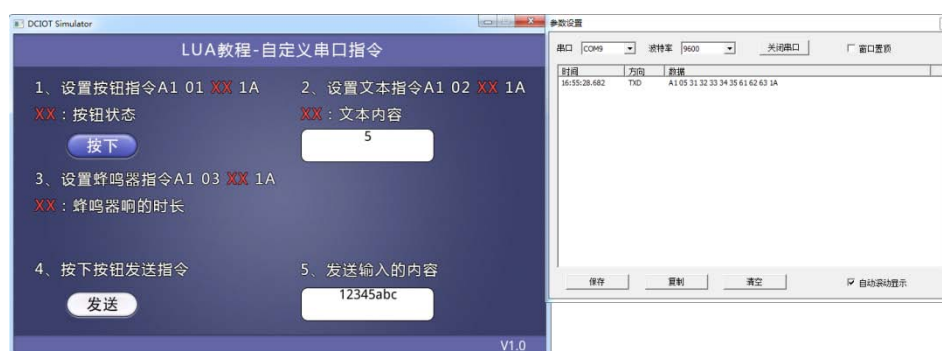


图 5-14 发送键盘输入的内容

## 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 5-15 和图 5-16 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。

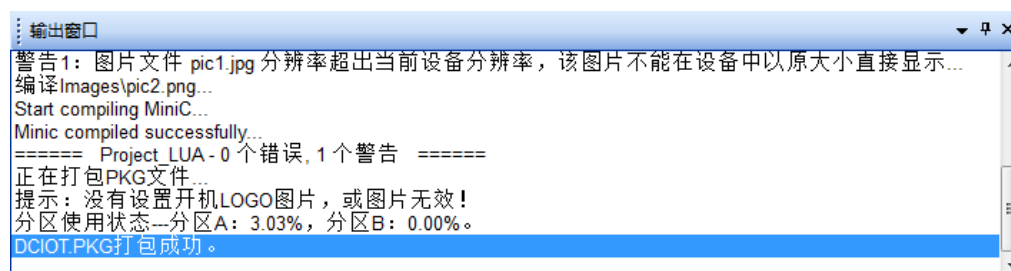


图 5-15 编译成功

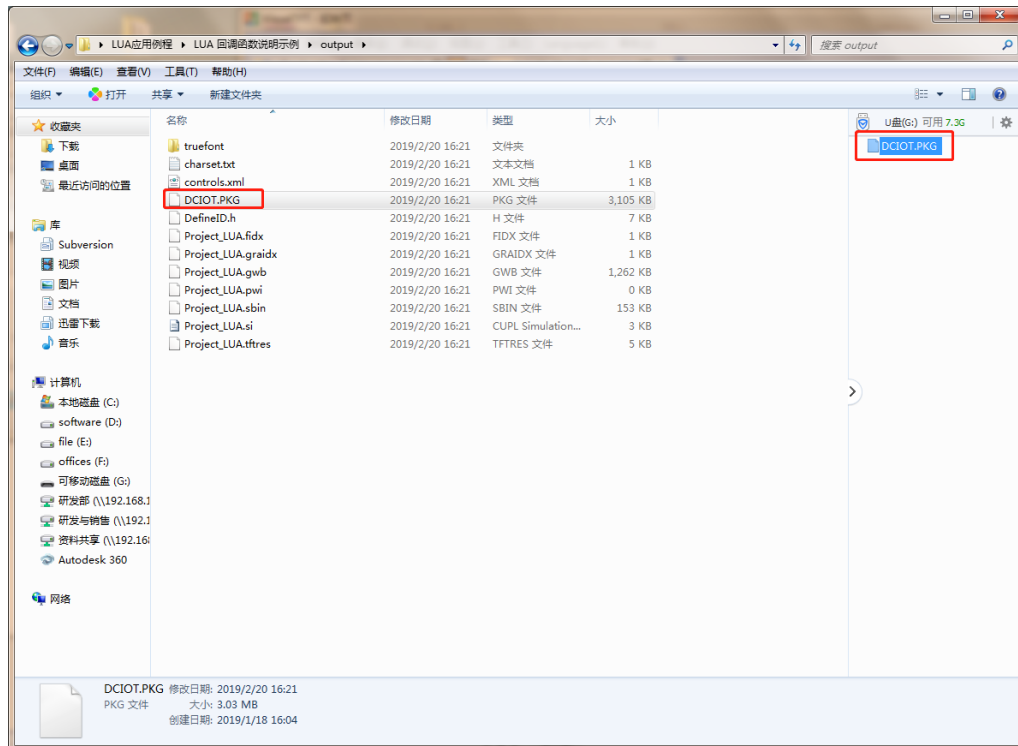


图 5-16 拷贝

## 6. 完整程序清单

自定义串口指令 demo 的完整程序，如程序清单 8 所示，查看程序时请结合教程 demo 理解：

程序清单 8 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-自定义串口指令
** Latest modified Date: 2019.02.22
** Latest Version:     V1.0
** Description:        使用自定义的串口指令操作控件
**
**-----
** Created By:         林绍佳
** Created date:       2019.02.22
** Version:            V1.0
** Descriptions:       使用自定义的串口指令操作控件
**
**-----
** 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
**
**-----
*****--]]

--[[*****
系统函数
** Function name:      on_init()
** Descriptions:       初始化函数
*****--]]

function on_init()

    --[[*****
    --功能:             设置串口接收超时时间
    --调用函数 1:      uart_set_timeout(timeout,timeeout_inter)
    --函数功能:        设置串口接收超时时间
    --参数:             timeout          接受总超时，单位：毫米
```

```

--          timeout_inter 字节间隔超时，单位：毫米
--*****--]]
uart_set_timeout(1000,200)
end

--设置全局变量 uart_free_protocol，使用自由串口协议
uart_free_protocol=1
--[[*****
系统函数
** Function name:          on_uart_rcv_data
** Descriptions:          串口自定义函数
** input parameters:      packet
*****--]]
function on_uart_rcv_data(packet)

--[[*****
--自定义指令
--使用自定义指令来设置按钮、文本、蜂鸣器
--方法如下：
--显示图片命令格式(16 进制): A1 A B 1A
--帧头: A1
--帧尾: 1A
--A: 01-按钮; 02-文本; 03-蜂鸣器;
--B: A 为 1 时，表示按钮状态;
--   A 为 2 时，表示文本内容;
--   A 为 3 时，表示蜂鸣器响的时长;
--例 1: A1 01 01 1A
--      表示设置按钮 1 按下
--例 2: A1 02 05 1A
--      表示设置文本 2 的内容为 05
*****--]]

if packet[0]==0xA1          --帧头为 0xA1
then

--[[*****
--功能:          控制按钮状态
--调用函数 1:    set_value(screen,control,value)
--函数功能:      设置控件数值
--参数:          screen: 控件的页码
--               control: 控件的编号
--               value:   控件被修改后的值
*****--]]
if packet[1]==0x01          --第二个字节对应按钮指令

```

```
    then
        if packet[2]==0x01          --第三个字节对应按钮按下的状态
        then
            set_value(0,1,1)
        else
            --第三个字节对应按钮弹起的状态
            set_value(0,1,0)
        end
    end
end

--[[*****
--功能:      设置文本
--调用函数 1:  set_text(screen,control,text)
--函数功能:   设置控件显示内容(字符串)
--参数:      screen:   被触发控件的页码
--            control:  控件的编号
--            text:    字符串
--*****--]]
if packet[1]==0x02          --第二个字节对应文本指令
then
    set_text(0,2,packet[2]) --设置文本为第三个字节
end

--[[*****
--功能:      设置蜂鸣器响
--调用函数 1:  beep(time)
--函数功能:   设置蜂鸣器响
--参数:      time:    响的时长, 单位: 毫秒
--*****--]]
if packet[1]==0x03          --第三个字节对应文本指令
then
    beep(packet[2])          --设置蜂鸣器响, 单位: 毫米
end
end
end

--设置按钮按下时发送的指令: A1 04 01 1A
local send_table1={04,01,26}
send_table1[0]=161

--创建表, 用于存储键盘输入完成后发送的指令
local send_table2={}

--创建字符串
local send_text2
```

```
--[[*****
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:  控件的页码
**                        control:  控件的编号
**                        value:    控件被修改后的值
*****--]]

function on_control_notify(screen,control,value)

    --[[*****
    --功能:          按下按钮时，通过串口发送指令
    --调用函数 1:    uart_send_data(packet)
    --函数功能:      通过串口发送数据
    --参数:          packet 数据，类型：表
    _*****--]]
    if screen==0 and control==4 and value==1
    then
        uart_send_data(send_table1)        --发送指令
    end

    --[[*****
    --功能:          发送键盘输入的内容
    --调用函数 1:    uart_send_data(packet)
    --函数功能:      通过串口发送数据
    --参数:          packet 数据，类型：表
    --调用函数 2:    get_text(screen,control)
    --函数功能:      获取控件字符串内容(字符串)
    --参数:          screen:  被触发控件的页码
    --                control:  控件的编号
    --返回:          字符串
    --调用函数 3:    string.len(arg)
    --函数功能:      计算字符串长度
    --参数:          arg:      字符串
    --返回:          整数值
    --调用函数 4:    string.byte(arg[,int])
    --函数功能:      转换字符为整数值(可以指定某个字符，默认第一个字符)
    --参数:          arg:      字符串
    --                [,int]:  索引
    --返回:          整数值
    _*****--]]
    if screen==0 and control==5
```

```
    then
        send_table2={ }                --表初始化
        send_table2[0]=161              --设置指令头, 0xA1
        send_table2[1]=0x05             --设置键盘指令
        send_text2=get_text(0,5)        --获取文本的内容
        for i=1,string.len(send_text2) do --把字符串转为表
            send_table2[i+1]=string.byte(send_text2,i)
        end
        send_table2[string.len(send_text2)+2]=26 --设置指令末, 0x1A
        uart_send_data(send_table2)      --发送指令
    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。