技术笔记

Technical Note

4G 应用- TCP 单连接 V1.0

工程技术笔记

类别	内容
关键词	4G、TCP
摘要	

广州大彩光电科技有限公司

修订历史

版本	日期	原因	编制	审查
V1.0	2020/08/31	创建文档	陈鹏	

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. 教程实现.....4
5.1.1 准备工程素材.....5
5.1.2 软件平台.....5
5.2 配置串口屏工程.....6
5.2.1 画面配置.....6
5.2.2 LUA编辑.....6
5.3 下载工程.....18
6. 免责声明.....20

1. 适合范围

本文档适合大彩 PM 4G 系列串口屏产品使用。

2. 开发环境版本

1. VisualTFT 软件版本：V3.0.0.1112 及以上的版本。

版本查看：

1) 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

2) 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。

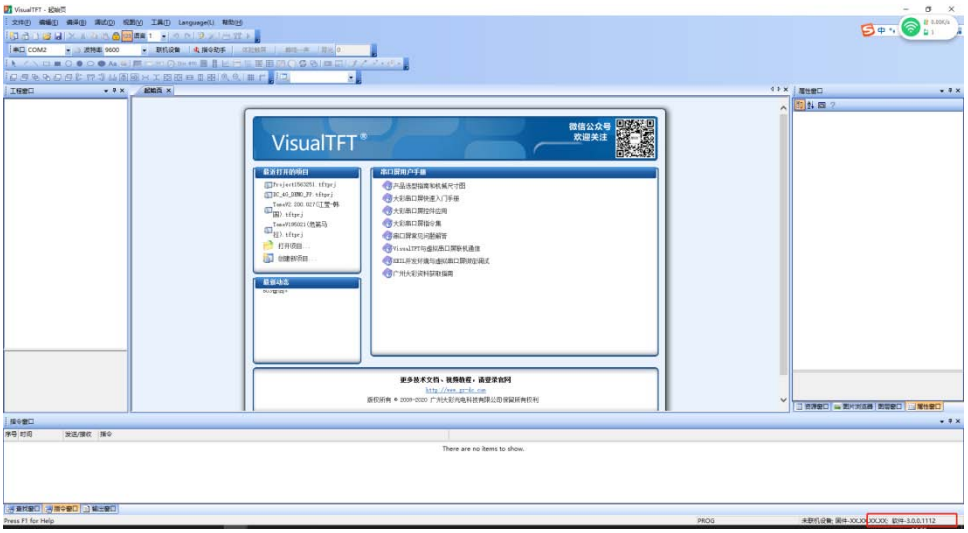


图 2-2 软件版本

2. 串口屏硬件版本： M 系列固件 >= V6.3.257.00。

版本查看：

1) 查看屏幕背面版本号贴纸；

2) VisualTFT 与屏幕联机成功后，右下角显示的版本号。

3. 概述

本例程中，介绍 4G TCP 单连接的应用。屏幕作为客户端，和服务器进行数据收发。

服务器TCP测试工具：本例程中使用合宙提供的TCP 透传云 做测试验证。该测试平台IP固定为 180.97.81.180，Demo中只需要更改端口号，在连接即可测试验证。

注意：UDP或多链接模式可参考《Luat 4G模块AT命令手册V4.2.2.pdf》即可

该平台创建服务器必须在 3 分钟内连上，且长时间未交互，服务器会自动关闭。

4. 参考资料

1. 《LUA 脚本 API V1.4》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
4. AT 指令，可以通过下面子连接了解
   http://www.openluat.com/Product/file/asr1802/Luat%204G 模块 AT 命令手册 V4.2.8.pdf
5. 合宙 TCP 测试平台
   http://tcplab.openluat.com

5. 教程实现

本文主要将以下 2 点进行说明：

1. 准备工程素材；
2. 配置串口屏工程；

5.1.1 准备工程素材

在实现例程前需要作以下 3 个准备：

1. 硬件平台；
2. 软件平台；
3. UI 素材；

该例程使用大彩 M 系列 7 寸串口屏 DC80480M070_1111_0T 为验证开发平台。如图 5-1 所示；

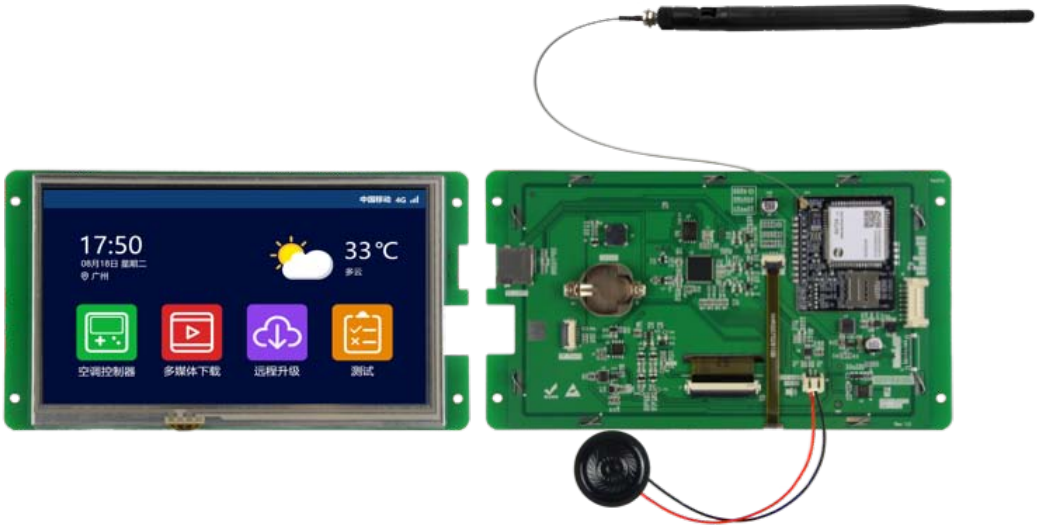


图 5-1 M 系列 7 寸串口屏

其他尺寸的串口屏均可借鉴此教程。

5.1.2 软件平台

使用大彩自主研发的上位机软件VisualTFT配置工程，登录http://www.gz-dc.com/下载。如图 5-2 所示；

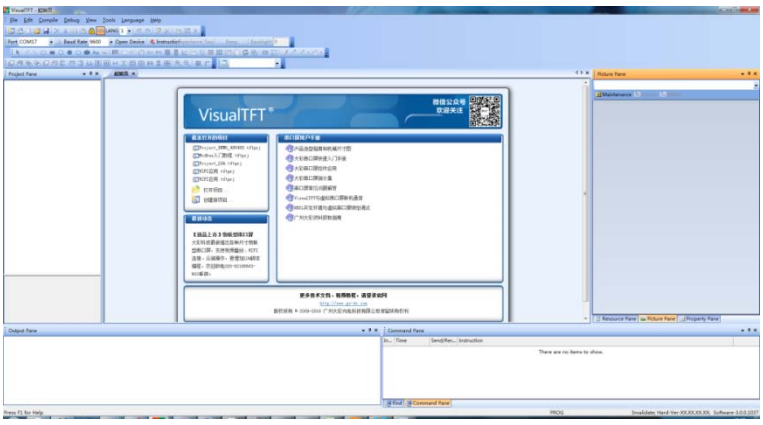


图 5-2 下载软件

5.2 配置串口屏工程

本文主要介绍以下 2 点：

(1) 画面配置

(2) LUA 编辑

5.2.1 画面配置

在画面 ID0 中，客户端、服务端和信号值/运营商 3 部分组成。

客户端：

- 文本控件 ID1：屏幕公网的 IP
- 文本控件 2~3：分别服务器 IP 地址、端口号。
- 按钮控件 ID4：当输入端口号后，点击连接。屏幕向该服务器发起请求信息。

服务器：

- 文本控件 ID7~9：当服务器下发数据时，将相关信息显示此处，依次为服务器的 IP、端口、数据等

4G 信号和运营商：

- 图标控件控件 ID10 作为信号显示、文本控件 ID11 用于显示运营商。

画面配置如图 5-3 所示：

注意：其他非关键控件不在一一介绍，下文不在累述

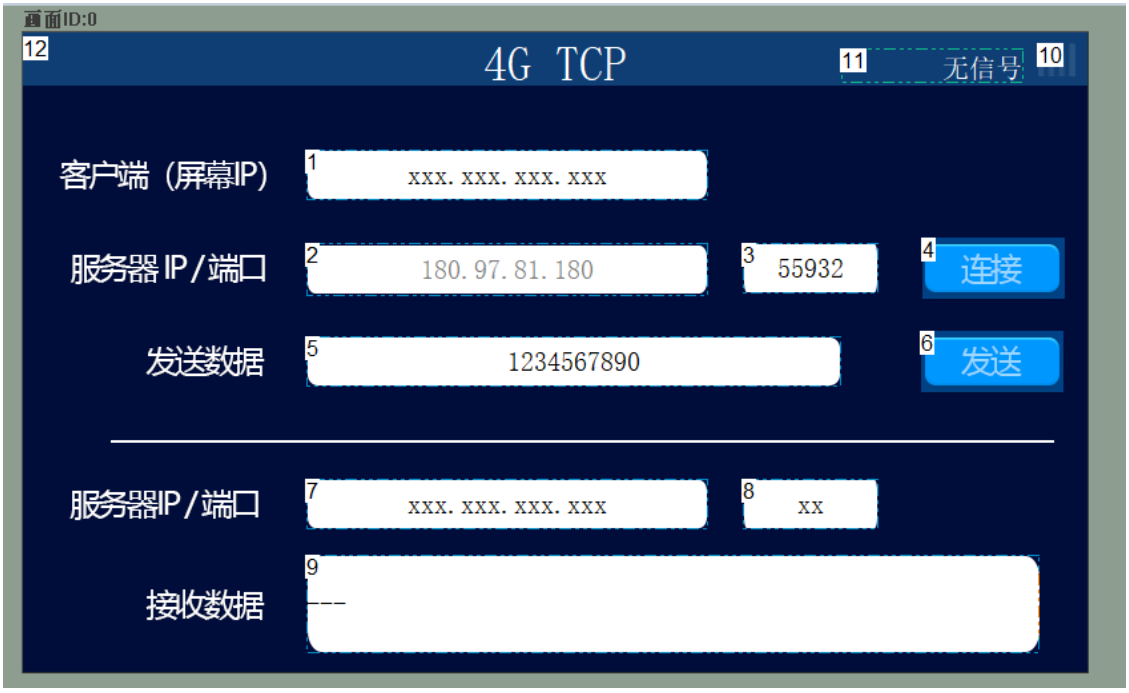


图 5-3 画面配置

5.2.2 LUA 编辑

本例程中，屏幕上电执行初始化操作，如加载 4G AT 指令的库、初始化 4G 模块、定时获取运营商和信号值等。

当用户点击图片下载的时，调用 air_http_download（）开始下载文件。在下载回调函数 on_http_download_file_cb（）里进行数据存储，显示下载信息等。若该文件单次下载不完，会多次回调 on_http_download_file_cb（）函数，直至下载完毕，如图 5-4 所示。

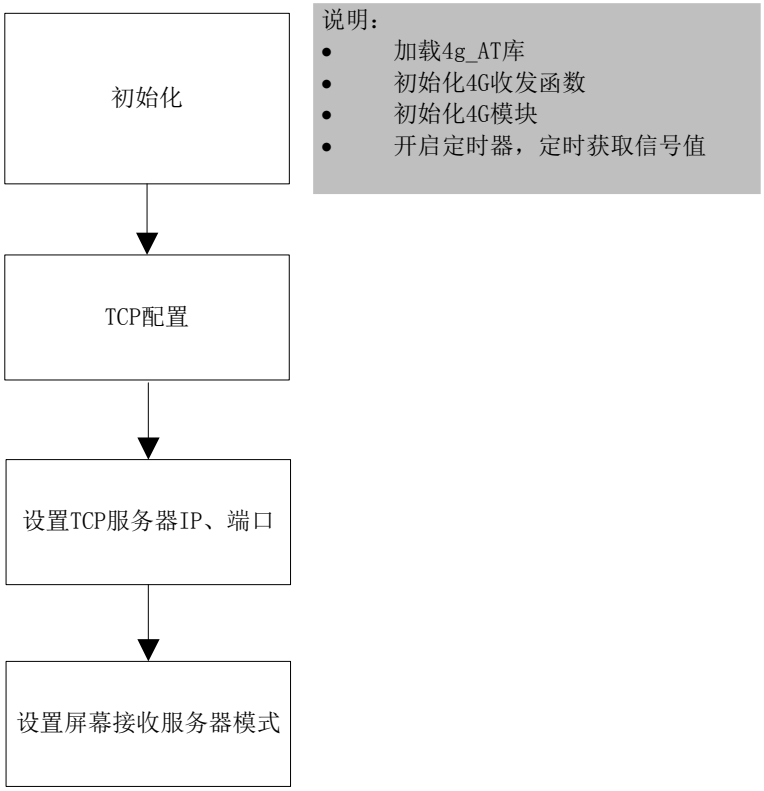


图 5-4 TCP 连接

1. 初始化

调用系统函数 on_init（）执行代码如程序清单 1 所示：

程序清单 1 初始化

```
--[[*****************************************************************
** Function name:  on_init
** Descriptions :  系统初始化时，执行此回调函数。
*****************************************************************--]]
function on_init()
    dofile('air724at.lua') --加载 http.lua 文件
    uart_set_baudrate3(115200) --设置与 4G 模块通讯的串口 3 的波特率为 115200

    --设置 4G 库函数的命令发送函数，命令回调函数、调试信息打印函数
    air_set_callback(on_air_send_cb,on_air_resp_callback,on_air_log_cb)
    air_hw_int() --4G 模块初始化设置
    --开启自动获取型号强度、时间、天气功能
    start_timer(timerId_Sig_Weather, 1000 , 0, 0)
end

--[[*****************************************************************
** Function name:  on_timer
** Descriptions :  定时器超时回到调函数。
```

```
** @ timer_id    :  定时器 ID
**********************************************************************--]]
function on_timer(timer_id)
    on_air_timer(timer_id) --4G 库函数的定时处理

    --定时获取信号强度
    if timer_id == timerId_Sig_Weather
    then
        --定时器计数，timer0_notify_cnt 每秒+1。
        timer0_notify_cnt = timer0_notify_cnt + 1
        if timer0_notify_cnt%15 == 0
        then
            --每 15s 调用一次，更新信号值
            at_cops_csq()
            timer0_notify_cnt = 0 --标志位清空
        end
    end
end

--[[**********************************************************************
** Function name : at_cops_csq
** Descriptions  : 获取运营商信息、信号强度
** @return      : nil,无返回值
**********************************************************************--]]
function at_cops_csq()
    air_cmd_add('AT+COPS?','OK',1000) –获取运营商
    air_cmd_add('AT+CSQ'  ,'OK',1000) –获取信号
end

--[[**********************************************************************
** Function name:  on_air_resp_callback
** Descriptions :  4G 模块-数据回调接口
** @key          :  屏幕向 4G 模块的发送请求
** @value        :  4G 模块返回的数据
**********************************************************************--]]
function on_air_resp_callback(key, value)

--**********************************************************************
    --功能:     判断 key
    --         如果 key 为空，则退出函数。
    --         因为 key 为空时，下方 string.find( key , ) 是不正确的使用。
    --         以下 key 的处理必须不为空，
    --**********************************************************************
    if key == nil
```

```
    then
        return
    end

    ......
      --*********************************************************************
    --条件：     4G 初始化完成
    --功能：     使用使用 AT 指令获取信号强度和运营商。
    --调用函数：at_cops_csq()
    --函数功能：获取信号强度和运营商
    --调用函数：at_tcp_client_cofig()
    --函数功能：TCP 配置
     *********************************************************************
    if string.find(key,'+SAPBR=1,1') ~= nil and string.find(value,'OK') ~= nil
    then
        at_cops_csq()
        at_tcp_client_cofig()
    end

    --*********************************************************************
    --条件：     4G 模块返回运营商信息
    --功能：
    --*********************************************************************
    if string.find(key,'+COPS') ~= nil and string.find(value,'+COPS') ~= nil
    then
        --*****************************************************************
        --value    ：  +COPS: 0,2,"46000",7
        --要提取的值：  46000
        --正则表达式：  '+COPS:.*,.*,"(%d*)"'
        --*****************************************************************
        local regular_e = '+COPS:.*,.*,"(%d*)"'--正则表达式
        --获取的值赋给 my_mobile_MCCMNC
        local my_mobile_MCCMNC = string.match( value, regular_e )
        set_text( screen_main, 2, mobile_MCCMNC[my_mobile_MCCMNC])
    end

    --*********************************************************************
    --条件：  4G 模块返回信号强度信息
    --功能：
    --*********************************************************************
    if string.find(key,'+CSQ')~=nil and string.find(value,'+CSQ')~=nil
    then
        --*****************************************************************
        --value    ：  +CSQ: 15,99
```

```
        --要提取的值： 15
        --正则表达式： '+CSQ: (.*),.*'
        --**********************************************************************
        local regular_e = '+CSQ: (.*),.*' --正则表达式
        local my_csq = tonumber(string.match(value,regular_e))
        if my_csq<=11
        then
           set_value( screen_main, 1, 1) --设置信号图标显示第 1 帧
        elseif my_csq>=12 and my_csq<=13
        then
           set_value(screen_main, 1, 2) --设置信号图标显示第 2 帧
        elseif my_csq>=14 and my_csq<=15
        then
           set_value( screen_main, 1, 3) --设置信号图标显示第 3 帧
        elseif my_csq>=16
        then
           set_value( screen_main, 1, 4) --设置信号图标显示第 4 帧
        end
    end
end

--[[**********************************************************************
** Function name: on_uart_recv_data3
** Descriptions : 接收串口 3 数据回调函数，连接 4G 模块。
**********************************************************************--]]
function on_uart_recv_data3(packet)
    --4G AT 指令库 API
    on_air_recv_data(packet)
end
```

核心 API 函数

1) dofile (filename)

加载文件：本例程中加载 4G AT 指令的库

- filename：文件名

2) uart_set_baudrate3(speed)

设置串口 3 的波特率：串口 3 为屏幕和 4G 模块通讯的串口

- speed：通讯的波特率

3) on_air_recv_data(packet)

串口接收 4G 模块的返回数据的回调。

- packet：形参为表，字节数据

4) air_set_callback (on_air_send_cb,on_air_resp_callback,on_air_log_cb)

设置 4G 库里的回调函数。形参类型为函数，参数依次为命令发送函数，命令

回调函数、调试信息打印函数，可自定义函数名。

- on_air_send_cb：屏幕向 4G 模块发送回调函数
- on_air_resp_callback：4G 向屏幕返回数据回调函数
- on_air_log_cb：用户调试信息回调函数调试

5) air_hw_int()

4G AT 指令的库函数，初始化 4G 模块

6) at_cops_csq()

自定义封装函数，获取运行商和信号值

7) air_cmd_add(sendstr,ackstr,timeout,retry,callback)

屏幕向 4G 模块发送 AT 指令

- sendstr ：屏幕向 4G 模块发送 AT 指令
- ackstr ：4G 模块应答屏幕的请求
- timeou ：应答超时
- retry ： 超时重发次数，可选
- callback ： 应答回调函数，可选

注：如果没有设置超时重发次数，则超时时直接发送队列中的下一条指令。

8) on_air_resp_callback(key, value)

4G 应答屏幕回调函数：屏幕发送 AT 指令，4G 应答后均会回调该函数，初始化设置：air_set_callback(on_air_send_cb,on_air_resp_callback,on_air_log_cb)。

- key ：屏幕向 4G 模块发送请求的 AT 指令
- value ：4G 模块返回的数据

相关 AT 指令：

本例程中，涉及到获取 4G 模块初始化、运营商、信号值等 AT 交互指令回调的判断，在 on_air_resp_callback(key, value)回调函数中，判断 4G 收发的相关 AT 指令，如下所示：

1) 网络数据是否激活：

屏幕发送：AT+SAPBR=1,1。air_hw_int()函数里发送。

屏幕接收：OK。on_air_resp_callback(key, value)函数里执行 4G 返回数据的判断。

2) 获取运营商：

屏幕发送：AT+COPS?。在 at_cops_csq()函数了发送。

屏幕接收：OK。on_air_resp_callback(key, value)函数里执行 4G 返回数据的判断。

3) 获取信号值：

屏幕发送：AT+CSQ。在 at_cops_csq()函数了发送。

屏幕接收：OK。on_air_resp_callback(key, value)函数里执行 4G 返回数据的判断。

2. TCP 配置

当 4G 初始化完毕后，在 on_air_resp_callback(key, value) 函数里调用 at_tcp_client_cofig()进行 TCP 配置，代码如程序清单 2 所示：

程序清单 2 TCP 配置

```
--[[*********************************************************************
** Function name:  on_air_resp_callback
** Descriptions :  4G 模块-数据回调接口
** @key          :  屏幕向 4G 模块的发送请求
** @value        :  4G 模块返回的数据
*********************************************************************--]]
function on_air_resp_callback(key, value)
    ......
    --*****************************************************************
    --条件:     4G 初始化完成
    --功能:     使用使用 AT 指令获取信号强度和运营商。
    --调用函数:at_cops_csq()
    --函数功能:获取信号强度和运营商
    --调用函数:at_tcp_client_cofig()
    --函数功能:TCP 配置
    --*****************************************************************
    if string.find(key,'+SAPBR=1,1') ~= nil and string.find(value,'OK') ~= nil
    then
        .......
        at_tcp_client_cofig()
    end
end

--[[*********************************************************************
** Function name : at_tcp_client_cofig
** Descriptions  : TCP 配置
** @return      : nil,无返回值
********************************************************************* --]]
function at_tcp_client_cofig(connect)
    local function my_get_ip_cb(key, value)

        if string.find(key,'+CIFSR') ~= nil and value ~= nil
        then
        Tcp_ip = value
        set_text(sc_Tcp, 1, Tcp_ip)
        end
    end

    local function my_ip_state(key, value)
        air_log('my debugMsg key   -> '..key)
        air_log('my debugMsg value -> '..value)

        if string.find(key,'+CIPSTATUS') ~= nil
```

```
            and string.find(value,'IP STATUS') ~= nil
        then
            Tcp_state = 1
            air_log('at_tcp_client_cofig success !!!!!!')
        end
    end

    if connect == nil
    then
        connect = 0
    end
    air_cmd_add('AT+CREG=1', 'OK', 1000, 3) --设置当前 GPRS 注册状态
    air_cmd_add('AT+CGREG=1', 'OK', 1000, 3) --设置当前 GPRS 注册状态
    air_cmd_add('AT+CGATT=1', 'OK', 1000, 3) --设置当前 GPRS 附着状态
    air_cmd_add('AT+CIPMUX='.. connect, 'OK', 1000, 3) --设置为单、多链接模式

    --设置为快发模式（推荐使用这种模式）
    air_cmd_add('AT+CIPQSEND=1', 'OK', 1000, 3)
    air_cmd_add('AT+CSTT?', 'OK', 1000, 3) --启动任务，查看 APN
    air_cmd_add('AT+CSTT=\"CMIOT\"', 'OK', 1000, 3) --启动任务，设置 APN

    air_cmd_add('AT+CIICR', 'OK', 1000, 3) --激活移动场景
    air_cmd_add('AT+CIFSR' , 'OK', 1000, 3, my_get_ip_cb) --查询分配的 IP 地址

    air_cmd_add('AT+CIPSTATUS' , 'STATE: IP STATUS', 1000, 3, my_ip_state) --
查询下链接状态

end
```

核心 API 函数

1) at_tcp_client_cofig(connect)

自定义函数，TCP 客户端配置

- connect：0-单链接，非 0 多链接

相关 AT 指令：

本例程中，TCP配置涉及到相关指令有设置为单、多链接模式、快发模式、激活等，相关关AT指令，更多详细的AT指令参考《Luat 4G模块AT命令手册V4.2.2.pdf》，如下所示：

1) 设置链接模式：

屏幕发送：AT+CIPMUX=0。设置单链接，非零（1/2）为多连接模式。

屏幕接收：OK

2) 设置移动激活场景：

屏幕发送：AT+CIICR?

屏幕接收：OK

3) 查询分配的 IP 地址：

屏幕发送：AT+CIFSR

屏幕接收：OK

3. 连接服务器

当用户输入端口号后，点击联机，调用 at_tcp_client_connect()并设置服务器接收模式，代码如程序清单 3 所示：

程序清单 3 连接服务器

```
--[[*********************************************************************
** Function name : at_tcp_client_connect
** Descriptions  : 设置接收模式
** @head        : 是否显示头部
** @showtp      : 是否显示协议
** @srip        : 是否显示 IP、端口
** @return      : nil,无返回值
**********************************************************************--]]
function at_set_tcp_recv_mode(head, showtp,srip)
   air_cmd_add('AT+CIPHEAD='..head,     'OK', 1000, 3) --显示头部
   air_cmd_add('AT+CIPSHOWTP='..showtp,  'OK', 1000, 3) --显示协议
   air_cmd_add('AT+CIPSRIP='..srip,     'OK', 1000, 3) --显示 IP、端口
end

--[[*********************************************************************
** Function name : at_tcp_client_connect
** Descriptions  : 连接服务器
** @mode        : 连接模式，TCP\UPD
** @server_ip    : 服务器地址
** @port        : 服务器端口
** @return      : nil,无返回值
**********************************************************************--]]
function at_tcp_client_connect(mode, server_ip, port)
   --连接服务器
   air_cmd_add(
         'AT+CIPSTART=\"'..mode..'\",'..'\"'..server_ip..'\",'..port,
         'OK',
         1000,
         3)
   --查询当前 GPRS 注册状态
   air_cmd_add('AT+CIPSTATUS', 'STATE: CONNECT OK', 1000, 3)

end
--[[*********************************************************************
** Function name : on_control_notify
```

```
    ** Descriptions  : 系统回调函数，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。
    ** @return       : nil,无返回值
    ********************************************************************--]]
    function on_control_notify(screen,control,value)
       if screen == sc_Tcp
       then
          if control == 4 and value == 0 and Tcp_state == 1
          then
             Tcp_server_ip = get_text(sc_Tcp, 3)
             Tcp_server_port = get_text(sc_Tcp, 3)
             at_tcp_client_connect('TCP', Tcp_server_ip, Tcp_server_port) --
连接服务器
             at_set_tcp_recv_mode(1, 1, 1)            --设置接收模式
          ......
          end

       end
    end
```

核心 API 函数

1) at_set_tcp_recv_mode(head, showtp,srip)

设置接收模式

- head: 是否显示头部
- showtp: 是否显示协议
- srip: 是否显示 IP、端口
- return: nil,无返回值

2) at_tcp_client_connect(mode, server_ip, port)

连接服务器

- mode: 连接模式，TCP\UPD
- server_ip : 服务器地址
- port : 服务器端口

相关 AT 指令：

1) 请求连接：

屏幕发送：AT+CIPSTART="TCP","60.166.18.9",7500。

屏幕接收：OK

2) 设置服务器数据包含头：

屏幕发送：AT+CIPHEAD=1

屏幕接收：OK

3) 设置服务器协议包含协议：

屏幕发送：AT+CIPSHOWTP=1

屏幕接收：OK

4) 设置服务器协议包含 IP、端口：
屏幕发送：AT+CIPSRIP=1
屏幕接收：OK

4. 发送数据到服务器

用户输入文本数据点击发送后，调用 at_tcp_send ()发送数据到服务端，代码如程序清单 4 所示：

程序清单 4 发送数据

```
--[[*********************************************************************
** Function name : on_control_notify
** Descriptions  : 系统回调函数，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。
** @return       : nil,无返回值
*********************************************************************--]]
function on_control_notify(screen,control,value)
   if screen == sc_Tcp
   then
      ......
         local send_data = get_text(sc_Tcp, 5)
         at_tcp_send(send_data)
      .......
   end
end

--[[*********************************************************************
** Function name : at_tcp_send
** Descriptions  : TCP、发送数据到服务端
** @str          : 发送数据
** @return       : nil,无返回值
*********************************************************************--]]
function at_tcp_send(str)
   air_cmd_add('AT+CIPSEND='..string.len(str) , '>', 1000, 3)
   air_cmd_add(str, 'DATA ACCEPT', 1000, 3)
end
```

核心 API 函数

1) at_tcp_send(str)
发送数据到服务器
str: 字符串

相关 AT 指令：

1) 发送数据（确定长度）：

屏幕发送：AT+CIPSEND=10
屏幕接收：>

2) 发送数据（数据内容）
屏幕发送：1234567890
屏幕接收：DATA ACCEPT:10

5. 接收服务器的数据

本例程中，接收服务器的信息是含接收员（目标服务器的 IP、端口）、帧头、协议类型、数据内容等，在函数 at_set_tcp_recv_mode(head, showtp,srip)配置。程序清单如程序清单 5 所示。

程序清单 5 接收服务器数据

```
--[[*********************************************************************
** Function name: split
** Descriptions : 切割字符串
** @ str         : 源字符串
** @ pat         : 分割符号
*********************************************************************--]]
function split(str, pat)
    local t = {}
    local last_end = 0
    local s, e = string.find(str, pat, 1) --第一个分割号
    local i = 1
    while s do
        table.insert(t, string.sub(str, last_end + 1, last_end + s - last_end
- 1))
        last_end = e
        s, e = string.find(str, pat, last_end + 1)
        i = i + 1
    end
    if last_end <= #str then
        cap = string.sub(str, last_end + 1)
        table.insert(t, cap)
    end
    return t
end

--[[*********************************************************************
** Function name:  on_air_resp_callback
** Descriptions :  4G 模块-数据回调接口
** @key        :  屏幕->4G 模块的发送请求
** @value      :  4G 模块->返回的数据
*********************************************************************--]]
```

```
function on_air_resp_callback(key, value)

    ......
    if value ~= nil
    then
        if string.find(value,'FROM') ~= nil
        then
            --+RECV FROM: 60.166.12.210:7500
            --分割成 3 份，分别为+RECV FROM、60.166.12.210、7500
            local serverInfo = split(value,':')
            set_text(sc_Tcp, 7, serverInfo[2]) -- ip
            set_text(sc_Tcp, 8, serverInfo[3]) --端口

        elseif  string.find(value,'IPD') ~= nil
        then
            --+IPD,TCP,10:123456789
            --分割成 2 份，分别为+IPD,TCP,10、123456789
            local serverData = split(value,':')
            set_text(sc_Tcp, 9, serverData[2]) -- 数据内容
        end
    end
    ......
end
```

相关 AT 指令：

1) 接收数据源（服务器 IP、端口）：
屏幕接收：+RECV FROM: 60.166.12.210:7500

2) 接收数据（帧头+协议+数据长度+数据内容）：
屏幕接收：+IPD,TCP,10:123456789

5.3 下载工程

工程编译成功后在输出窗口会提示编译成功，如图 5-5 所示；

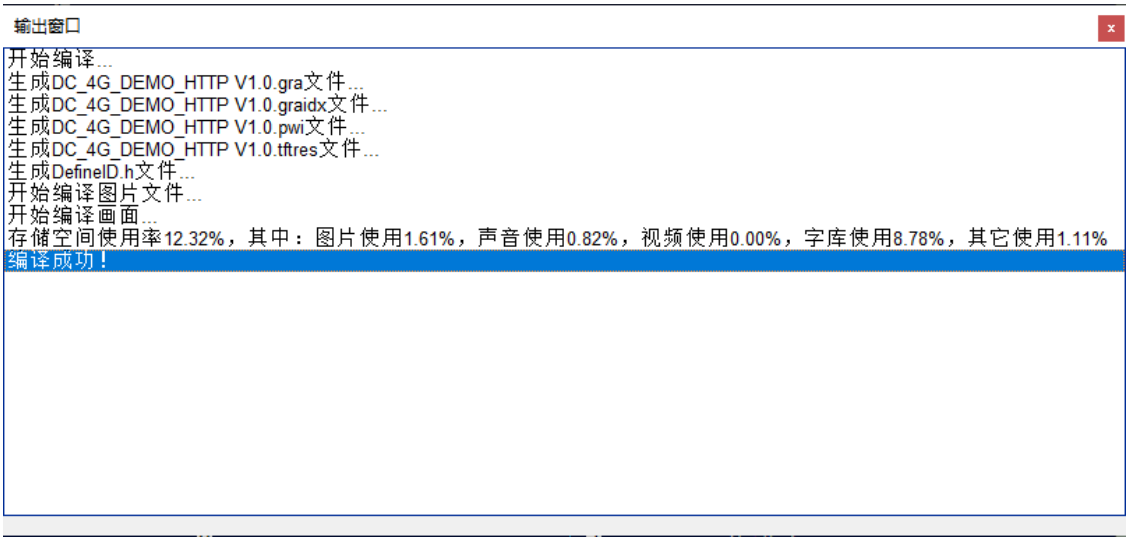


图 5-5 编译成功

在菜单栏中，文件→打开工程目录，在‘dciot_build’目录的 ‘private’拷贝到 SD 卡中，如图 5-6 和图 5-7 所示；把 SD 卡接上串口屏后重新上电，等到提示烧录工程成功后，拔掉 SD 卡重新上电即可。

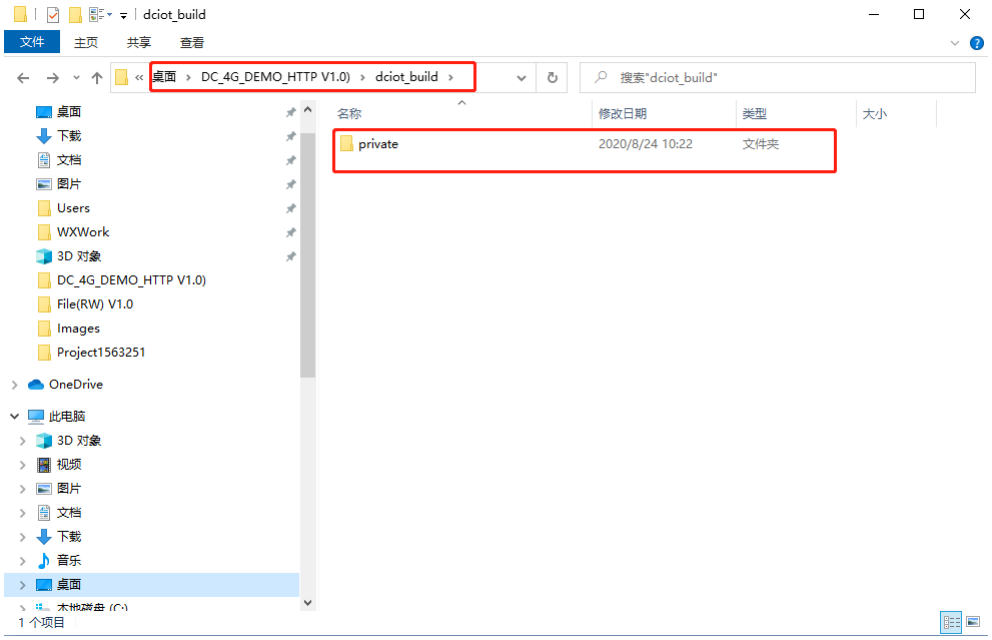


图 5-6 下载文件

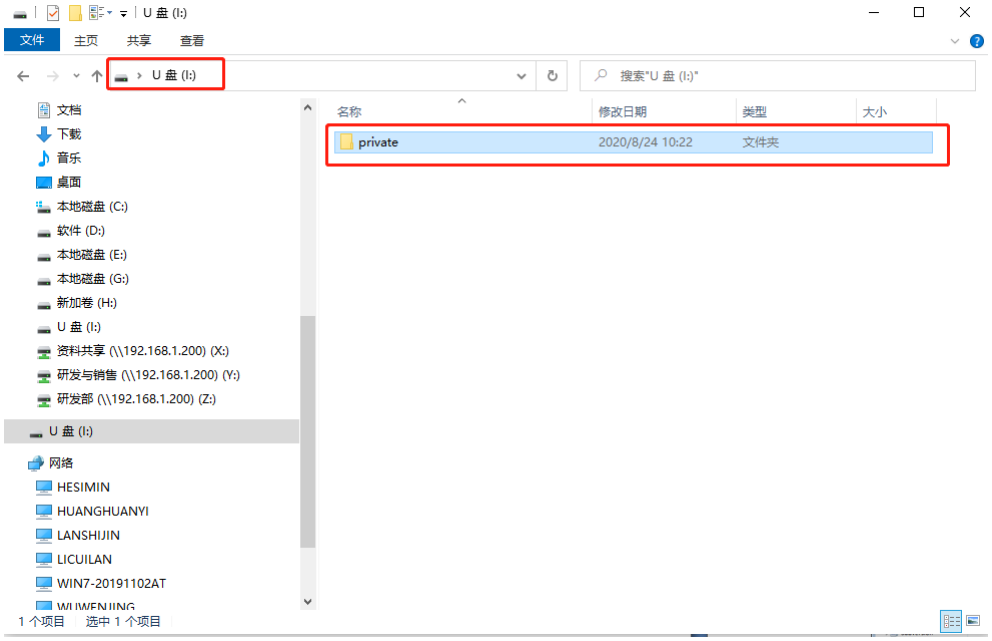


图 5-7 拷贝到 SD 卡

6. 免责声明

本文档提供有关广州大彩光电科技有限公司（以下简称：大彩科技）产品的信息，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。大彩科技产品不能在用于军事、医疗、救生或维生等用途中作为唯一控制设备。

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除大彩科技在其产品的销售条款和条件中声明的责任之外，大彩科技概不承担任何其它责任。并且，大彩科技对大彩科技产品的销售和/或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。大彩科技可能随时对产品规格及产品描述做出修改，恕不另行通知。