技术笔记

Technical Note

4G 应用-HTTP 下载文件 V1.1

工程技术笔记

类别	内容
关键词	4G、HTTP 下载文件
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	2020/07/02	创建文档	陈鹏	
V1.1	2021/12/14	修正软件版本	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.....5
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.....20
6. 免责声明.....22

1. 适合范围

本文档适合大彩 PM 系列串口屏产品使用。

2. 开发环境版本

1. VisualTFT 软件版本：V3.0.0.1133 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。

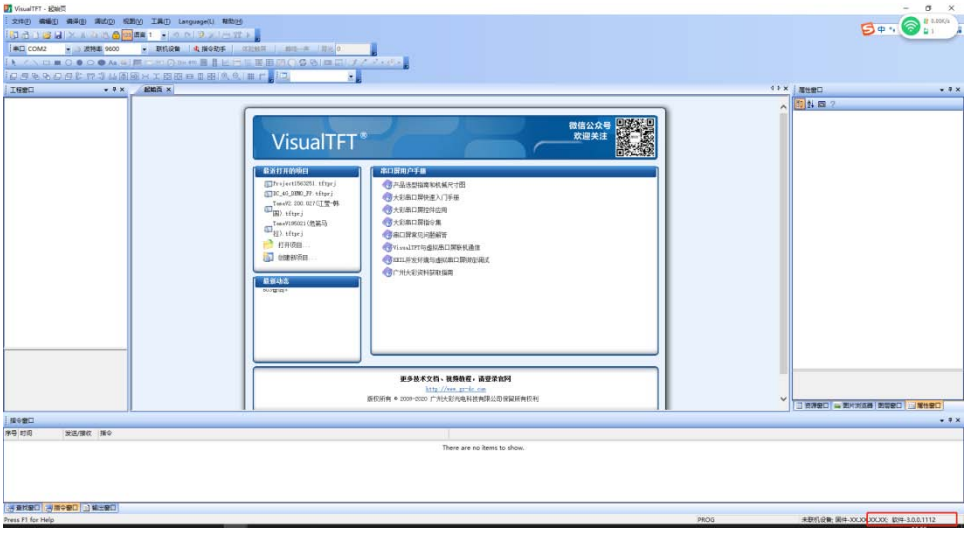


图 2-2 软件版本

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 的 HTTP 下载的应用，通过下载图片、视频、音频说明下载流程。

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. Air 系列 4G 模块 AT 版本 HTTP 应用说明（若连接失效请在 LUAT 社区中查找）
   https://ask.openluat.com/article/626/0

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；

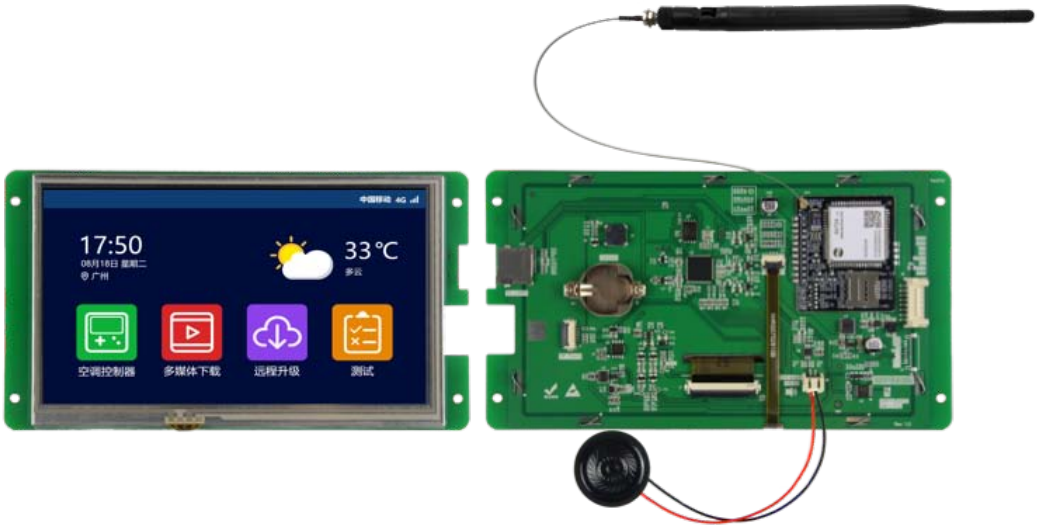


图 5-1 M 系列 7 寸串口屏

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；

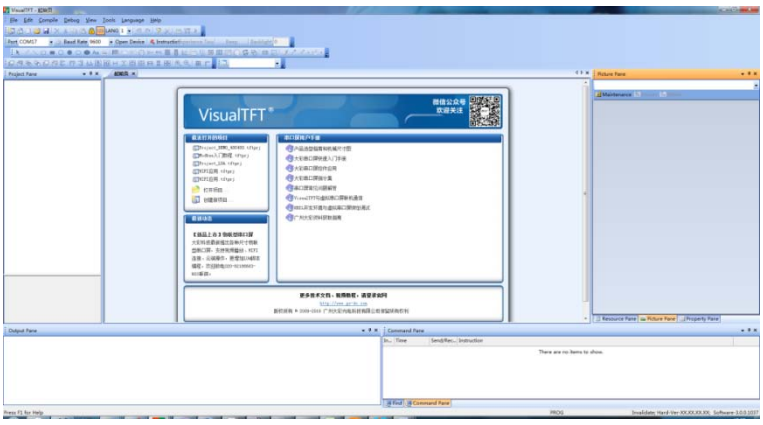


图 5-2 下载软件

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，触发下载、下载过程、4G 信号值及运营商 3 部分组成。

触发下载：3 个按钮控件，作为触发下载条件。

- 控件 ID1~3 依次为图片下载、视频下载以及音频下载。

下载过程：控件 ID4~8 作为下载过程的体现：

- 文本控件 ID4 用于显示下载速度（min/Kb）
- 文本控件 ID5 显示下载信息
- 进度条控件 ID6 为显示下载进度
- 文本控件 ID7 用于显示下载进度百分比
- 文本控件 ID8 显示 ‘当前已下载大小/总下载大小’，单位 Kb

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。

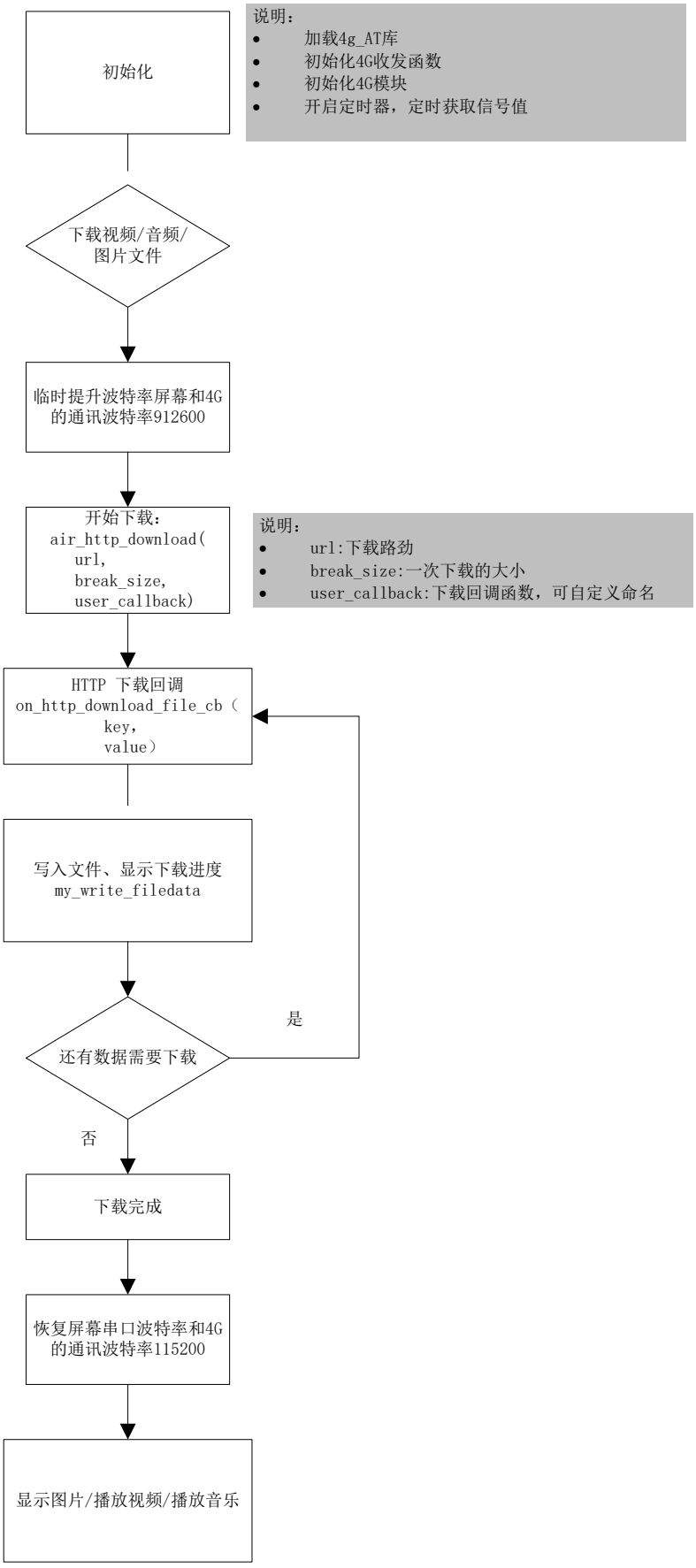

初始化
说明：
加载4g_AT库
初始化4G收发函数
初始化4G模块
开启定时器，定时获取信号值
下载视频/音频/
图片文件
临时提升波特率屏幕和4G
的通讯波特率912600
开始下载：
air_http_download(
url,
break_size,
user_callback)
说明：
url:下载路劲
break_size:一次下载的大小
user_callback:下载回调函数，可自定义命名
HTTP 下载回调
on_http_download_file_cb（
key,
value）
写入文件、显示下载进度
my_write_filedata
还有数据需要下载
是
否
下载完成
恢复屏幕串口波特率和4G
的通讯波特率115200
显示图片/播放视频/播放音乐

图 5-4 下载流程

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)
    set_visiable(screen_mmedia_dl,6,0) --设置进度条不可见
    set_visiable(screen_mmedia_dl,7,0) --设置进度文本不可见
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)

    if value == nil
    then
        return            --value 为空时退出
    end

--*********************************************************************
    --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:     设置 4G 模块波特率成功
    --功能:     设置串口波特率为 921600
    --*****************************************************************
    if string.find(key,'+IPR=921600') ~= nil and
        string.find(value,'OK') ~= nil
    then
        uart_set_baudrate3(921600)
    end
```

```
--*****************************************************************
--条件：     设置 4G 模块波特率成功
--功能：     设置串口波特率为 115200
--*****************************************************************
if string.find(key,'+IPR=115200') ~= nil and
string.find(value,'OK') ~= nil
then
   uart_set_baudrate3(115200)
end

--*****************************************************************
--条件：     4G 模块返回运营商信息
--功能：     设置显示运营商
--*****************************************************************
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：     设置 4G 模块的信号等级
--*****************************************************************
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。在 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. HTTP 下载

用户点击按钮控件 ID1~3，HTTP 协议下载对应的图片、视频、音频文件。文件，每次向服务器读取 7K 大小文件，然后在下载回调函里，每次 以 2K 字节的数据写入文件，代码如程序清单 2 所示：

程序清单 2 HTTP 下载

```
--[[*****************************************************************
** 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。
```

```
** @ screen      : 控件触发所在的页面
** @return       : 控件触发的 ID
** @return       : 控件值
** @return       : nil,无返回值
**********************************************************************--]]
function on_control_notify(screen,control,value)
    if screen == screen_mmedia_dl
    then

        if (control >= 1 and control <= 3) and value == 1
        then
            --停止音乐、视频的播放、下载的图片不显示
            picture_en = 0
            stop_sound()
            stop_video()

            --下载过程失能下载按钮，不允许点击
            set_enable(screen_mmedia_dl, 1, 0)
            set_enable(screen_mmedia_dl, 2, 0)
            set_enable(screen_mmedia_dl, 3, 0)

            set_visiable(screen_mmedia_dl,6,1) --设置进度条可见
            set_visiable(screen_mmedia_dl,7,1) --设置进度文本可见
            set_visiable(screen_mmedia_dl, 9, 0) --设置音乐图标隐藏

            --复位下载相关变量
            file_len         = 0
            file_current_dl    = 0
            file_curt_dl_perct = 0

            临时提升波特率，提高 4G 和屏幕串口传速度
            air_set_baudrate(921600)
            down_mode = control
            set_text(screen_mmedia_dl,5,httpDown_dl_process[down_mode])

            --开始下载
            air_http_download(HttpDownFile_URITb[down_mode],
                              7168,
                              on_http_download_file_cb)
        end
    end
end

--[[**********************************************************************
```

```
** Function name: http_download_process
** Descriptions : 计算当前写入（下载）进度
** @cur_process : 当前下载进度
** @return      : nil, 无返回值
*****************************************************************--]]
function http_download_process( cur_process )

    local allsize =  string.format('%0.2f', (file_len    / 1024))
    local cursize =  string.format('%0.2f', (cur_process / 1024))
    local file_curt_dl_perct = (cur_process/file_len)*100

    file_curt_dl_perct=tonumber(string.format('%0.1f',file_curt_dl_perct))

    set_value(screen_mmedia_dl, 6, file_curt_dl_perct)
    set_text(screen_mmedia_dl, 7, '下载进度: '..file_curt_dl_perct..' %')
    set_text(screen_mmedia_dl, 8, '文件大小:'..cursize..' K/'..allsize..' K')
end

--[[*****************************************************************
** Function name: on_http_download_file_cb
** Descriptions : 下载文件
** @key         : key, HTTP 下载返回标识
** @value       : value,4G 模块屏幕  的应答数据
** @return      : nil, 无返回值
*****************************************************************--]]
function on_http_download_file_cb(key, value)

    if key=='data_len' then
        file_len = value --获取文件长度
        --创建文件
        open_state = file_open( HttpDownSave_FileTb[down_mode], over_write)
        start_timer_download_speed(file_len) --开始计算下载时间
        set_text( screen_mmedia_dl, 4, '')   --速度显示为空

    elseif key=='data'  --文件数据包
    then
        --分段写入数据
        my_write_filedata(HttpDownSave_FileTb[down_mode],value,add_write)

    elseif key=='finish' --下载结束
    then
        --重新开启自动获取信号
        start_timer(timerId_Sig_Weather, 5*1000 , 0, 0)
        file_close() --关闭文件
```

```
--复位相关下载变量
file_len         = 0
file_current_dl   = 0
file_curt_dl_perct = 0
set_value( screen_mmedia_dl, 6, 0) --初始化进度条数值
set_text( screen_mmedia_dl, 7, '下载进度：0%') --初始化进度文本数值
set_text( screen_mmedia_dl , 8, '')
set_visiable( screen_mmedia_dl, 6, 0) --设置进度条不可见
set_visiable( screen_mmedia_dl, 7, 0) --设置进度文本不可见

--停止下载计时，比返回总用时秒数，每秒下载速度
local dl_sec,dl_speed = stop_timer_download_speed()
set_text(screen_mmedia_dl,4,
        '下载速度:'..string.format('%0.2f',(dl_speed*60))..'KB/min')
--重新设置 4G 模块和串口 3 波特率为 115200
air_set_baudrate(115200)

--下载模式为 1 时显示图片
if down_mode == 1
then
   set_text(screen_mmedia_dl,5,httpDown_dl_process_end[down_mode])
   if surface ~= 0
   then
      destroy_surface(surface)
      surface = 0
   end
   picture_en=1
   surface = load_surface(HttpDownSave_FileTb[down_mode])
   redraw()

--下载模式为 2 时播放视频
elseif down_mode == 2
then
   set_text(screen_mmedia_dl,5,httpDown_dl_process_end[down_mode])
   picture_en = 0
   stop_video()
   play_video(HttpDownSave_FileTb[down_mode], 40, 75, 480, 320)

--下载模式为 3 时播放音乐
elseif down_mode == 3
then
   set_text(screen_mmedia_dl,5,httpDown_dl_process_end[down_mode])
   set_visiable(screen_mmedia_dl, 9, 1) --设置音乐图标显示
   picture_en = 0
```

```
    stop_video()
    stop_sound()
    play_sound(HttpDownSave_FileTb[down_mode])
end

set_value(screen_mmedia_dl   , 1, 0)
set_value(screen_mmedia_dl   , 2, 0)
set_value(screen_mmedia_dl   , 3, 0)
set_enable(screen_mmedia_dl  , 1, 1)
set_enable(screen_mmedia_dl  , 2, 1)
set_enable(screen_mmedia_dl  , 3, 1)

elseif key=='timeout' --下载超时
then
    start_timer(timerId_Sig_Weather, 5*1000 , 0, 0)

    set_text(screen_mmedia_dl  , 5, '下载超时，请重新下载! ')
    dl_reset_flag = 1
    set_value(screen_mmedia_dl , 1, 0)
    set_value(screen_mmedia_dl , 2, 0)
    set_value(screen_mmedia_dl , 3, 0)
    set_enable(screen_mmedia_dl, 1, 1)
    set_enable(screen_mmedia_dl, 2, 1)
    set_enable(screen_mmedia_dl, 3, 1)

    air_set_baudrate(115200) --重新设置 4G 模块和串口 3 波特率为 115200
    stop_timer_download_speed() --停止下载计时

elseif key=='dl_file_read_retry' -- HTTPREAD 重下
then
--*****************************************************************
--value    :  retry: -1,601
--要提取的值:  -1
--            601
--正则表达式:       'retry: (%d*),(%d*)'
--*****************************************************************

local httpread_retry=''
local status=''
local regular_e = 'retry: (.*),(%d*)' --正则表达式
--获取的值赋给
local httpread_retry,status = string.match(value,regular_e)
air_log('user r retry: '..httpread_retry..','..status)
if httpread_retry == '-1' and ( status == '408' or
```

```
                                        status == '601' or
                                        status == '603' )
        then
            reset_4G()
            --提示
            set_text( screen_mmedia_dl, 5, '下载超时，发送网络错误，重新设置中...')
            dl_reset_flag = 1
        end

    elseif key=='dl_file_head_retry' -- HTTPHEAD 重下
    then
        --*************************************************************
        --value    :     retry: -1,601
        --要提取的值:       -1
        --                  601
        --正则表达式:       'retry: (%d*),(%d*)'
        --*************************************************************
        local httphead_retry=''
        local status=''
        local regular_e = 'retry: (.*),(%d*)'  --正则表达式
        local httphead_retry,status = string.match(value,regular_e)
        air_log('user h retry: '..httpread_retry..','..status)
        if httphead_retry == '-1' and ( status == '408' or
                                        status == '601' or
                                        status == '603' )
        then
            reset_4G()
            set_text( screen_mmedia_dl, 5, '下载超时，发送网络错误，重新设置中...')
            dl_reset_flag = 1
        end
    end
end
```

核心 API 函数

1) air_http_download(url,break_size,user_callback)

4G 库函数，HTTP 下载文件。

- url：HTTP 下载资源连接，本例程中，将图片、视频以及音频的资源路径放在数组缓冲区中，如下所示：

```
local HttpDownFile_URITb =
                    {'http://video.gz-dc.com/logo_480320_165kV2.jpg',
                     'http://video.gz-dc.com/video2_62M9s.mp4',
                     'http://video.gz-dc.com/music503k32s.mp3'}
```

- break_size：单次网络下载的包大小，单位字节（BYTE），本例程中为 7K
- user_callback：改形参是一个函数变量，HTTP 下载应答屏幕的回调函数。

名称可自定义命名，如本例程为 on_http_download_file_cb。

2) on_http_download_file_cb(key, value)

用户自定义函数，HTTP 下载回调函数。

- key：HTTP 响应数据类型
- value：HTTP 响应的内容

若 key = ‘data_len’，表示 HTTP 下载的文件大小；value 表示为文件大小，单位 byte。

若 key = ‘data’，表示 HTTP 响应数据内容；value 为数据内容，类型为字节数组，大小为 air_http_download(url,break_size,user_callback)设置的 break_size。

若 key = ‘finish’，表示 HTTP 下载完成；value 为文件总大小，单位 byte

若 key = ‘timeout’，表示 HTTP 下载超时。

若 key = ‘dl_file_head_retry’，表示 HTTPACTION 查询头信息超时。若超时，内部已经处理了 3 次重发数据请求。

若 key = ‘dl_file_read_retry’，表示 HTTPACTION 查询读取数据超时。若超时，内部已经处理了 3 次重发数据请求。

3) my_write_filedata(file, data, open_mode)

用户自定义函数，写文件。本例程，将 4G HTTP 下载返回的数据写在文件中。

- file：存储路径，本例程中，将图片、视频以及音频下载后的文件指定存储在屏内（3 为屏内盘符），路径定义如下所示：

```
local HttpDownSave_FileTb = {'3:/image.jpg','3:/bench.mp4', '3:/my dream.mp3'}
```

- data：存储的数据，类型可以是‘数组’或‘字符串’，本例程中，4G HTTP 下载返回的数据类型是数组
- open_mode：文件打开模式：本例程中，每次写文件，将数据写在文件尾。

注意：本文不再阐述文件读写的详细说明，可参考相应的资料，如《Lua 应用-文件读写 V1.0.pdf》

4) load_surface（filename）

加载图片到图层。

- filename 图片文件路径，支持 JPEG/PNG

本例程中，当下载完成 key= ‘finish’ 时，且当前下载的是图片文件（down_mode = 1），调用此 API 申请图层绘图，并调用 redraw()刷新触发 on_draw()显示。

注意：本文不再阐述绘图的详细说明，可参考相应的资料，如《LUA教程 7-绘图API使用说明.pdf》

5) destroy_surface (surface)

销毁图层，W 系列支持、M 系列 >= V6.1.241.00、F 系列不支持

- surface 图层资源指针

6) play_sound(filename)

播放指定的声音文件

- filenam：音频文件路径，支持 MP3/WAV

本例程中，当下载完成 key= ‘finish’ 时，且当前下载的是音频文件（down_mode = 3），调用 此 API 播放音频。若播放完毕，触发音频播放结束通知 on_audio_callback（），在重复播放该音频文件。

注意：本文不再阐述音频的详细说明，，可参考相应的资料，如 《M 系列-播放 SD 卡里视频、音频文件 V1.0.pdf》

7) play_video(file,left,top,width,height)

播放指定的视频文件

- file：文件路径，支持 MP4
- left：起始坐标 x
- top：起始坐标 y
- width：视频显示的宽度
- height：视频显示的高度

本例程中，当下载完成 key= ‘finish’ 时，且当前下载的是视频文件（down_mode = 2），调用 此 API 播放视频，播放过程触发 on_video_notify(msg，v1, v2)。若 msg=0，表示播放完毕，在重复播放该视频文件。

注意：本文不再阐述视频的详细说明，，可参考相应的资料，如 《M 系列-播放 SD 卡里视频、音频文件 V1.0.pdf》

8) on_video_notify(msg， v1, v2)

视频播放回调函数

- msg：1-播放中，0-播放完毕
- v1:当前播放进度，当前已播时长，单位 s
- v2:播放总进度，当前视频总时长，单位 s

9) on_audio_callback (state)

声音播放结束回调通知，state 保留未使用。

10) on_draw(screen)

当界面的显示内容需要更新时，系统自动调用此函数，用户在此函数中添加自定义的绘图操作。用户绘制的内容叠加在画面内容之上。此函数为系统回调函数，用户不要直接调用。

下面几种情况会触发此函数：

- 界面有动画播放、视频播放、RTC 时间显示的动态刷新；
- 用户操作屏幕控件控件；
- 通过 LUA 脚本或串口指令更新控件；

- 通过执行 redraw；

界面上有任何变化，都会触发此函数

11) draw_surface (surface,dstx,dsty,width,height,srcx,srcy)

绘制图层，相比于 draw_image_file，此方法效率较高，W 系列支持、M 系列 >= V6.1.241.00 支持（裁剪显示）、F 系列不支持

- surface 图层资源指针
- dstx 图片显示 X 坐标
- dsty 图片显示 Y 坐标
- width 图片显示宽度[可选]
- height 图片显示高度[可选]
- srcx 图片裁剪 X 坐标[可选]
- srcy 图片裁剪 Y 坐标[可选]

例如：

平铺显示：draw_surface(surface, dstx, dsty)

缩放显示：draw_surface(surface, dstx, dsty, width, height)，此方法 M 型不支持

裁剪显示：draw_surface(surface, dstx, dsty, width, height ,srcx, srcy)

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；

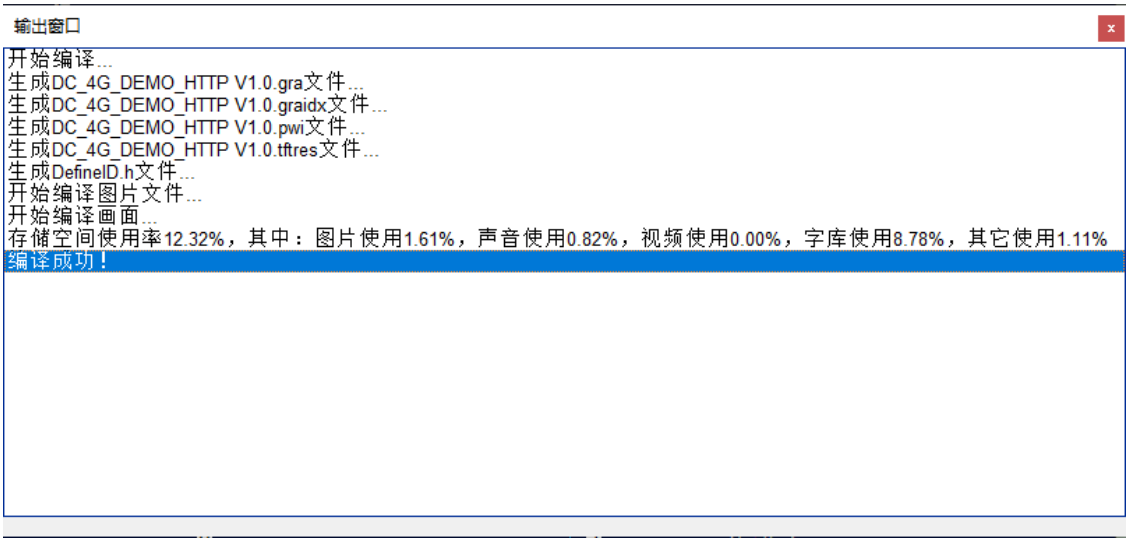


图 5-5 编译成功

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。

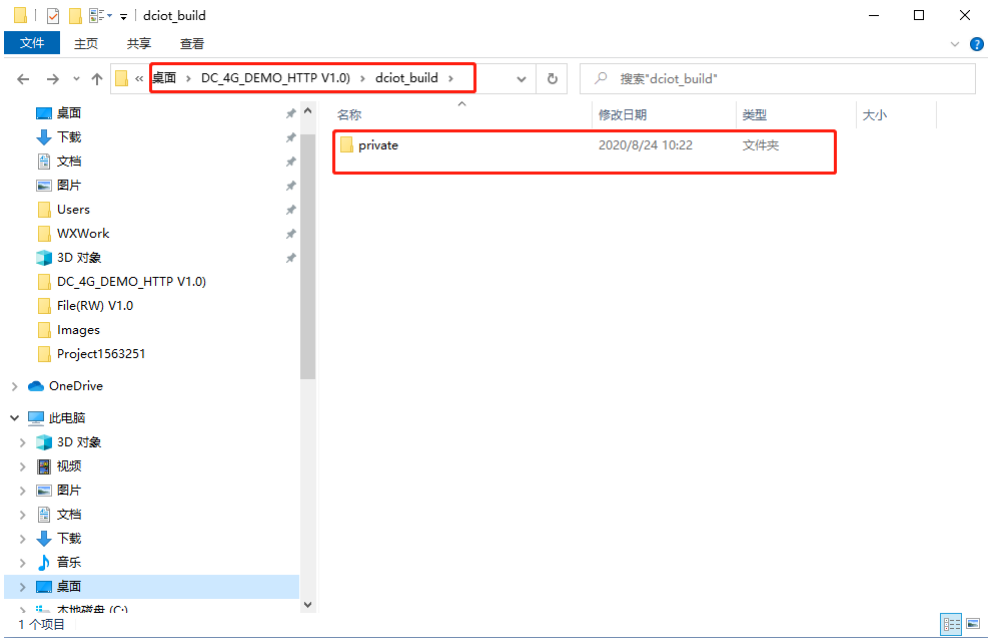


图 5-6 量产向导

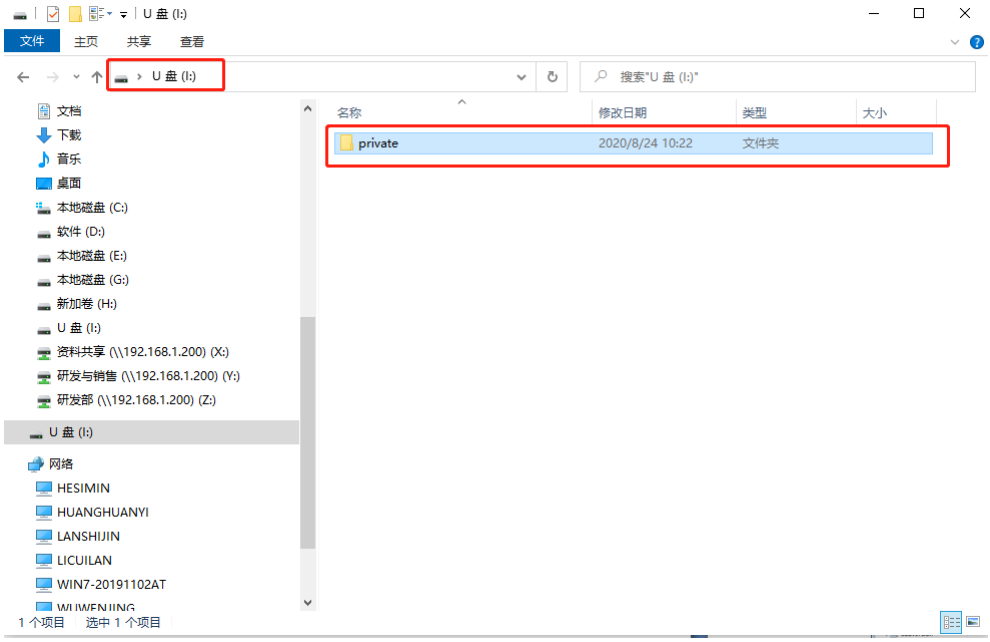


图 5-7 拷贝到 SD 卡

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。