技术笔记

Technical Note

4G 应用-连接阿里云 V1.0

工程技术笔记

类别	内容
关键词	4G、阿里云
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	

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 阿里云平台配置.....6
5.3 配置串口屏工程.....6
5.3.1 画面配置.....6
5.3.2 LUA 编辑.....7
5.4 下载工程.....22
6. 免责声明.....24

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：V3.0.1.1133 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。

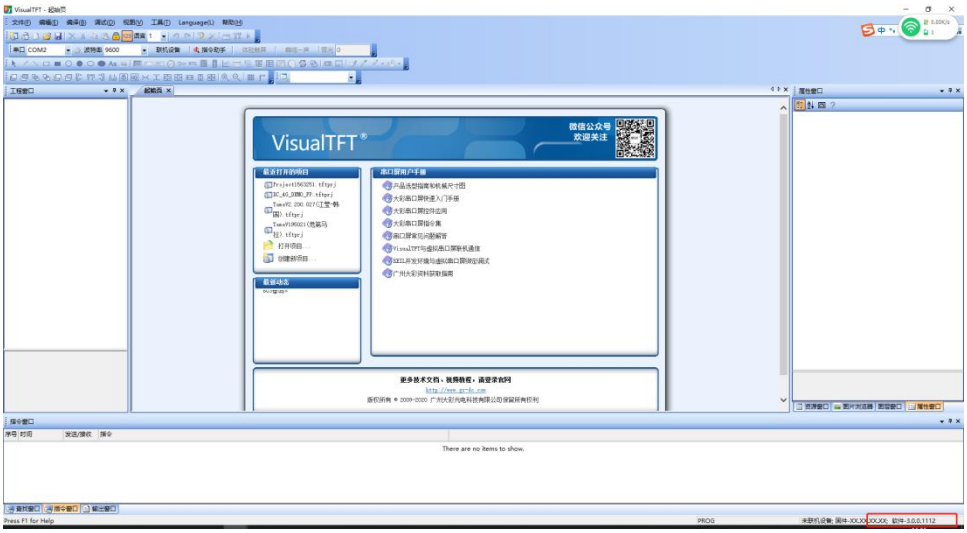


图 2-2 软件版本

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

阿里云 IoT 提供的一款针对生活领域的物联网平台，以解决家电设备快速智能化的问题。平台针对家电智能化的设备连接、移动端控制、设备管理、数据统计等问题，提供了一整套配置化方案，大幅减低设备-云-APP 的开发成本。了解阿里云平台更多信息，详细介绍请参考阿里云官网上《关于生活物联网平台》文档。

本例程主要介绍，PW-4G 系列如何用一机一密方式连接阿里云。以空调为例，对空调的温度、开关机、模式、滤网时间等参数进行数据交互。

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. 连接阿里云 AT 流程说明（一机一密认证方案）.pdf
   http://www.openluat.com/Product/file/rda8910/连接阿里云 AT 流程说明（一机一密认证方案）.pdf
6. 连接阿里云 AT 流程说明（一型一密认证方案）.pdf
   http://www.openluat.com/Product/file/rda8910/连接阿里云 AT 流程说明（一型一密认证方案）.pdf

5. 教程实现

本章节主要串口屏-阿里云-APP 数据交互的配置流程，将分为以下 3 个阶段讲述教程 DEMO 是如何实现：

1. 准备工程素材；
2. 阿里云平台配置；
3. 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；

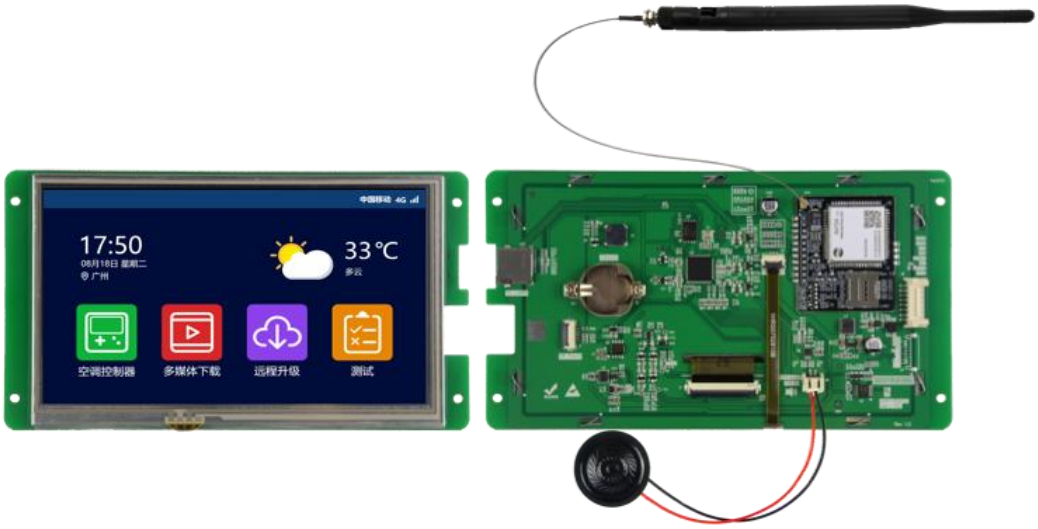


图 5-1 M 系列 7 寸串口屏

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；

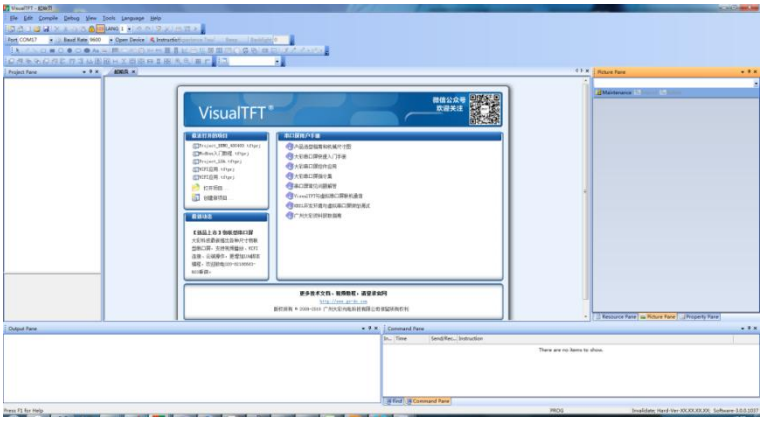


图 5-2 下载软件

5.2 阿里云平台配置

阿里云生活物联网平台配置流程主要包括以下 7 个步骤：

1. 注册账号
2. 创建工程
3. 创建产品
4. 功能定义
5. 设备调试
6. 人机交互
7. 批量投产

详细到配置流程，可参考我司《WIFI-连接阿里云教程》相应阿里云平台配置章节说明，或可参考直接登录阿里云官网，文档中心查阅更多内容。

5.3 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

5.3.1 画面配置

在画面 ID0 中，信号运营商、APP QR Code、APP 交互变量组成。

信号运营商：图标件 ID12 表示信号等级、文本控件 ID13 表示运营商

APP QR Code：二维码控件，使用手机云智能 APP（阿里云公办 APP）扫码，由于是一型一密，用户扫码需要开发者分享授权。

APP 交互的变量：控件 ID1~ID10 用于显示、设置开/关机、滤网时间、温度、设备模式等。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

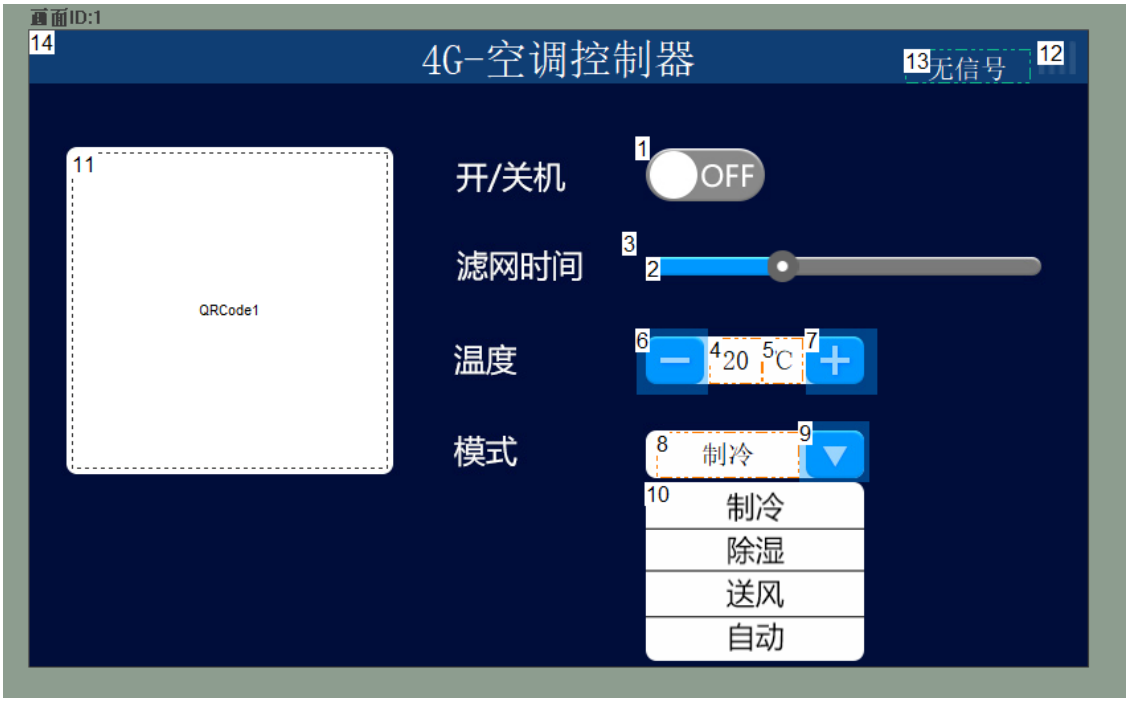
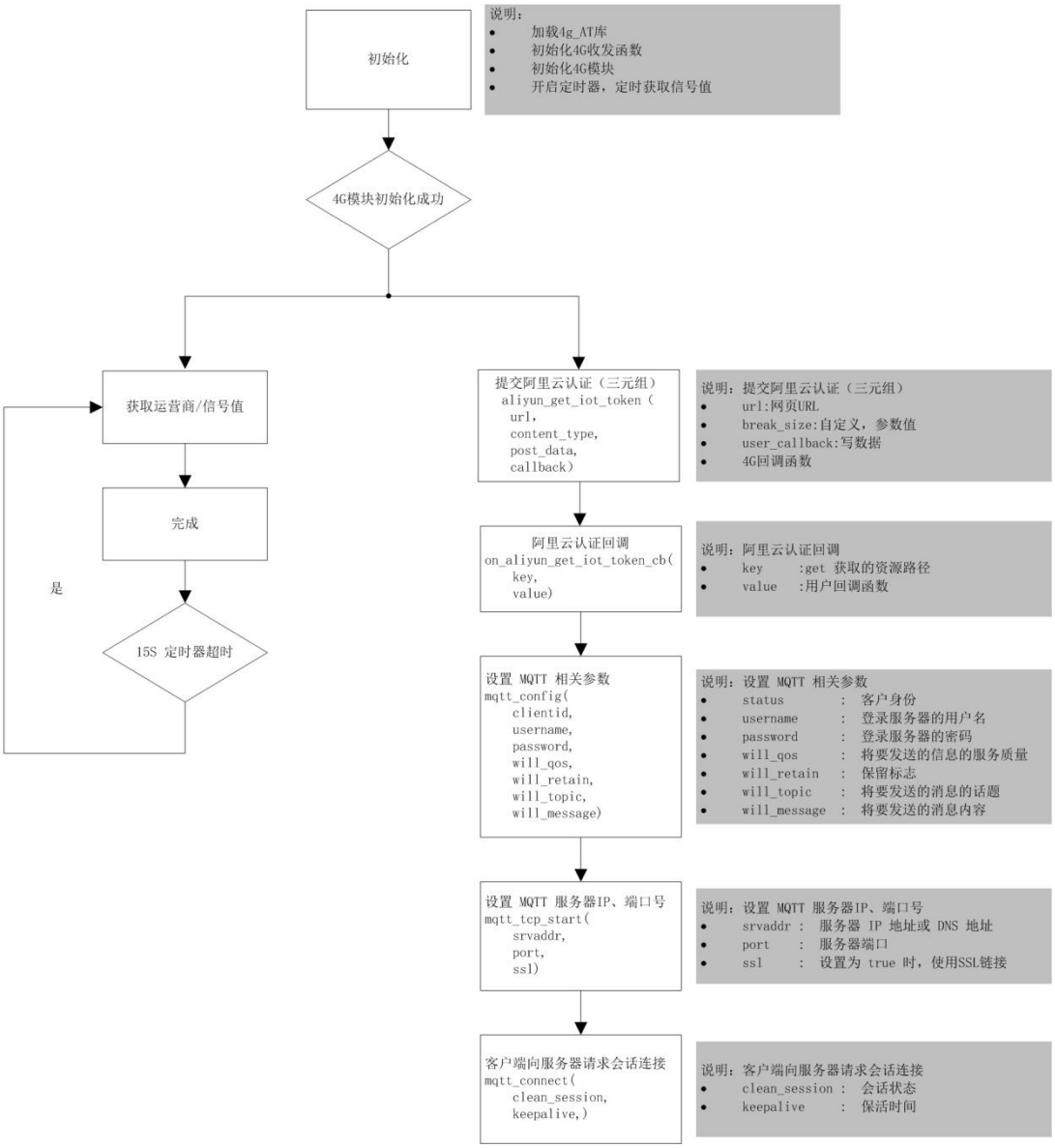


图 5-3 画面配置

5.3.2 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、初始化和 4G 模块的交互的函数、初始化 4G 模块、开启定时器获取运营商和信号值。

4G 模块初始化完成后，提交阿里云认证，并设置 mqtt 相关参数、服务 IP 和端口等等，屏幕发出请求会话通知。一切就绪后，屏幕和云端可以进行数据交互。



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
```

```
    air_get_iccid()  -- 获取卡号
    --开启自动获取型号强度
    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

    --自动获取型号强度
    if timer_id == timerId_Sig_Weather
    then
        --定时器计数，timer0_notify_cnt 每秒+1。计时长度 30min
        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()
        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->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
    ......
    if key ~= nil
    then
        return              -- key 为空时退出
    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。
    --          使用 MQTT 提交三元组，阿里云认证。
    --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：aliyun_get_iot_token()
    --函数功能：提交阿里云认证
    ==================================================================]]
    if string.find(key,'+SAPBR=1,1') ~= nil and string.find(value,'OK') ~= nil
    then
        at_cops_csq()           --获取信号强度
        aliyun_get_iot_token() -提交阿里云认证
    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 库函数的回调。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) at_cops_csq()

自定义封装函数，获取运行商和信号值

6) 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 ：应道超时时间
- 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。

7) 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，本函数如 air_set_callback(on_air_send_cb,on_air_resp_callback,on_air_log_cb)函数设置。

- key ：屏幕->4G 模块，发送请求的 AT 指令
- value ：4G 模块->屏幕，返回的数据

相关 AT 指令：

本例程中，初始化部分在 on_air_resp_callback(key, value)回调函数中，需要判断 4G 收发的相关 AT 指令，如下所示：

1) 获取运营商：

屏幕发送：AT+COPS?

屏幕接收：OK

2) 获取信号值 AT：

屏幕发送：AT+CSQ

屏幕接收：OK

3) 网络数据是否激活：

屏幕发送：AT+SAPBR=1,1

屏幕接收：OK

2. 提交阿里云认证

当 4G 初始化完成后，提交阿里云认证（三元组），代码如程序清单 2 所示：

程序清单 2 阿里云认证

```
    --出厂工程测试使用的阿里云服务器的登录 id 和三元组
    local clientId     = '862991419835242'           --无限制
    local productKey   = 'a1D2E9vaSuZ'               --三元组的 productKey
    local deviceName   = 'tuOVqg3nrVZnR2oMWwev'      --三元组的 deviceName
    local deviceSecret = '36c51acc5e3a7e410fe55831fd5e899d'         --三元组的
deviceSecret

   --[[*****************************************************************
   ** Function name: aliyun_get_iot_token
   ** Descriptions : 连接 Aliyun
   ** return       : nil,无返回值
   *****************************************************************--]]
    function aliyun_get_iot_token()
       local msg  = 'clientId'..clientId..
                    'deviceName'..deviceName..
                    'productKey'..productKey
       local sign = md5_hmac(msg,deviceSecret)   --计算哈希值
       local http_data = 'productKey='..productKey..
                         '&sign='..sign..
                         '&clientId='..clientId..
                         '&deviceName='..deviceName
       --提交 Aliyun 认证
       air_http_post(
                 'https://iot-auth.cn-shanghai.aliyuncs.com/auth/devicename',
                 'application/x-www-form-urlencoded',
                 http_data,
                 on_aliyun_get_iot_token_cb
                 )

       --设置二维码
       qrstring =
          'https://g.aliplus.com/ilop/static/download/ilopdownload.html?
           locale=zh-CN&pk='..productKey..'&dn='..deviceName
       set_text( screen_ac_control, 11, qrstring)
    end

    --[[*****************************************************************
    ** Function name:  on_aliyun_get_iot_token_cb
    ** Descriptions :  Aliyun 认证回调函数
    **                 成功时发送命令 AT+MCONFIG 和 AT+SSLMIPSTART
```

```
**********************************************************************--]]
function on_aliyun_get_iot_token_cb(key, value)
    if key=='data' then
        local jsondata = cjson.decode(table2str(value)) --json 解码
        if jsondata ~= nil and jsondata['code'] == 200
        then
            iotId = jsondata['data']['iotId']
            iotToken = jsondata['data']['iotToken']

            mqtt_config(clientId,iotId,iotToken)
            mqtt_tcp_start(
                    productKey .. '.iot-as-mqtt.cn-shanghai.aliyuncs.com',
                    1883,
                    True )
        end
    end
end
```

核心 API 函数

1) aliyun_get_iot_token()

提交阿里云认证（三元组）和设置二维码。

2) air_http_post (url,content_type,post_data,callback)

HTTP POST 操作，例程中次数用于提交阿里云认证（三元组）。

- url：网页 URL
- content_type：自定义，参数值
- post_data：写数据
- callback：4G 数据回调函数

3) on_aliyun_get_iot_token_cb(key, value)

Aliyun 认证回调函数，当云端反馈正常认证后，调用系统 4G AT 指令的库的 mqtt_config（）设置 MTQQ 参数、mqtt_tcp_start（）设置服务器 IP 和端口。

- key：云端恢复的类型
- Value:云端恢复的数据

3. 设置云端 IP 和端口

当云端认证成功和设置 mqtt 相关参数后，调用 mqtt_config() 、mqtt_tcp_start() 设置云端 IP 和端口，调用 mqtt_connect() 与服务器建立会话连接 ，代码如程序清单 3 所示：

程序清单 3 设置云端 IP 和端口

```
--[[**********************************************************************
** Function name:  on_aliyun_get_iot_token_cb
** Descriptions :  Aliyun 认证回调函数
**                 成功时发送命令 AT+MCONFIG 和 AT+SSLMIPSTART
```

```
--[[*****************************************************************--]]
function on_aliyun_get_iot_token_cb(key, value)
    ......
    mqtt_config(clientId,iotId,iotToken)
    mqtt_tcp_start(productKey ..
                   '.iot-as-mqtt.cn-shanghai.aliyuncs.com', 1883, true)
    ......
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)

    my_print('on_air_resp_callback()')
    ......
    --*****************************************************************
    --条件:     连接 Aliyun，AT+SSLMIPSTART 成功
    --功能:     发送 AT+MCONNECT=...
    --*****************************************************************
    if string.find(key, '+SSLMIPSTART') ~= nil and
       string.find(value, 'CONNECT OK') ~= nil
    then
        my_print('mqtt_connect(1,600)')
        mqtt_connect(1,600)
    end
end
```

核心 API 函数

1) mqtt_config(clientid,username,password,will_qos,will_retain,will_topic,will_message)

设置 MQTT 相关参数，本例程，只需要填写客户身份、登陆服务器的用户名以及密码，其中这三个参数是有阿里云认证反馈回来的。

- status：客户身份
- username：登录服务器的用户名
- password：登录服务器的密码
- will_qos：将要发送的信息的服务质量
- will_retain：保留标志
- will_topic：将要发送的消息的话题
- will_message：将要发送的消息内容

2) mqtt_tcp_start(srvaddr,port,ssl)

设置服务器的 IP、端口

- srvad：服务器 IP 地址或 DNS 地址
- port：服务器端口
- ssl：设置为 true 时，使用 SSL 链接

3) mqtt_connect()

客户端向服务器请求会话连接

- clean_session：会话状态
- keepalive ：保持时间

当屏幕向云端发送设置服务器 IP、端口请求的时候，云端回应数据，触发回调函数 on_air_resp_callback(key, value)，应答‘CONNECT OK’后，发起请求连接 mqtt_connect(1,600)

4. 发起连接请求

当客户端向服务器请求会话连接成功后，客户端对服务器订阅主题，代码如程序清单 4 所示：

程序清单 4 发起连接请求

```
--[[*****************************************************************************
** 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)
    ......
    --*************************************************************************
    --条件:      接收 Aliyun 发布消息
    --功能:      在 cloud_on_property_set(payload) 中对消息进行处理。
    --调用函数:  cloud_on_property_set(payload)
    --*************************************************************************
   if string.find(key, '+MCONNECT') ~= nil
      and string.find(value,'CONNACK OK')~=nil
   then
       mqtt_sub(
           '/sys/'..productKey..'/'..deviceName..'/thing/service/property/
set',
           0 )
       mqtt_sub_bit = 'sub_OK'
    end
end
```

核心 API 函数

1) mqtt_sub(topic,qos)

本命令从客户端到服务器，用于一个或多个订阅主题，当向服务器请求

- topic：应用程序消息的主题
- qos：申请消息的服务质量

5. 接收云端信息

当用户手机 APP 设置数据的时候，云端自动将数据下发给屏幕，代码如程序清单 5 所示：

程序清单 5 接收云端数据

```
--[[*********************************************************************
** 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
        return            --value 为空时退出
    end

    if key == nil
    then
        return
    end

    --*****************************************************************
    --条件:      接收 Aliyun 发布消息
    --功能:      在 cloud_on_property_set(payload) 中对消息进行处理。
    --调用函数:  cloud_on_property_set(payload)
    --*****************************************************************
    if string.find(value,'+MSUB') ~= nil and key == nil
    then
        local index = string.find(value,'{')

        my_print('index: '..index)

        if index ~= nil
        then
            local payload = string.sub(value, index,-1)
            if payload ~= nil
            then
                cloud_on_property_set(payload)
            end
```

```
        end
    end
end

--[[*****************************************************************************
** Function name:  cloud_on_property_set
** Descriptions :  接收云端数据
** @payload     :  数据字符串，Json 格式
*****************************************************************************--]]
function cloud_on_property_set(payload)
    my_print('cloud_on_property_set')

    --payload = string.gsub(payload,'\\22','\"');

    local data = cjson.decode(payload) --解析 JSON 数据

    if data == nil
    then
        return
    end

    if data['params'] == nil
    then
        return
    end

    --开关机按钮
    local powerstate = tonumber(data['params']['powerstate'])
    if powerstate ~= nil
    then
        set_value(screen_ac_control,1,powerstate)
    end

    --滤网时间进度条
    local filter_time = tonumber(data['params']['filter_time'])
    if filter_time ~= nil
    then
        set_value(screen_ac_control,3,filter_time)
    end

    --温度文本
    local targetTemperature = tonumber(data['params']['targetTemperature'])
    if targetTemperature ~= nil
    then
```

```
            set_value(screen_ac_control,4,targetTemperature)
        end

        --模式选择控件
        local mode = tonumber(data['params']['mode'])
        if mode ~= nil
        then
            set_text(screen_ac_control,8,my_resp_mode[mode])
        end
    end
```

核心 API 函数

1) cloud_on_property_set(payload)

接收云端的数据

- payload：Json 格式的字符串

本例程中，假设用户通过 APP 设置设备的模式，下发 Json 格式的字符串，数据原型如下所示：

```
+MSUB:"/sys/a1D2E9vaSuZ/PlYPGnoYWeUJujyOli2a/thing/service/property/set",95
byte,{"method":"thing.service.property.set","id":"2032042748","params":{"mode
":3},"version":"1.0.0"}
```

通过 Json 库函数 cjson.decode(payload)解压数据，则 data['params']['mode'] 为模式的值，在将值设置到对应的滑动选择控件上：

```
    set_text(screen_ac_control,8,my_resp_mode[mode])
```

6. 发送云端信息

当用户触碰控件修改值时候，屏幕会自动将数据打包成 Json 格式发给云端，代码如程序清单 6 所示。

程序清单 6 发送数据到云端

```
    local params = {}
    local property = {}
    local post_id = 1                --屏发布的消息次数 ID
    --[[*********************************************************************
    ** Function name:  aliyun_pub
    ** Descriptions :  屏发布消息时，此函数处理消息内容，并发送
    ** @screen      : aliyun 画面 ID
    ** @control     : 控件 ID
    ** @value       : 控件值
    ** @return      : nil,无返回值
    *********************************************************************--]]
    function aliyun_pub(screen, control, value)
        params   = nil
        property = nil
        params   = {}
        property = {}
```

```
    property['method'] = 'thing.event.property.post'
    property['id'] = ''..post_id
    property['version'] = '1.0'
    post_id = post_id + 1

    if screen == screen_ac_control
    then
        --按下开关按钮
        if control == 1
        then
            local PowerSwitchName = 'powerstate'
            params[PowerSwitchName] = value --在表写入按钮数值
            property['params'] = params
            local jsonStr = cjson.encode(property)
            --发布消息
            mqtt_pub(
'/sys/'..productKey..'/'..deviceName..'/thing/event/property/post',
            0,
            0,
            jsonStr
            )
        end
    end
    if screen == screen_ac_control
    then
        --按下滑块
        if control == 3
        then
            local PowerSwitchName = 'filter_time'
            params[PowerSwitchName] = value --在表写入滑块数值
            property['params'] = params
            local jsonStr = cjson.encode(property)
            --发布消息
            mqtt_pub(
'/sys/'..productKey..'/'..deviceName..'/thing/event/property/post',
            0,
            0,
            jsonStr
            )
        end
    end

    if screen == screen_ac_control
    then
```

```
            --按下温度设置按钮，温度减
            if control == 6 and value == 1
            then
                temp_value = tonumber(get_text(screen_ac_control,4))
                temp_value = temp_value-1
                if temp_value<16 then
                    temp_value = 16
                end
                set_text(screen_ac_control,4,temp_value)

            --按下温度设置按钮，温度加
            elseif control == 7 and value == 1
            then
                temp_value = tonumber(get_text(screen_ac_control,4))
                temp_value = temp_value+1
                if temp_value>32 then
                    temp_value = 32
                end
                set_text(screen_ac_control,4,temp_value)
            end
            local PowerSwitchName = 'targetTemperature'
            params[PowerSwitchName] = temp_value --在表写入温度数值
            property['params'] = params
            local jsonStr = cjson.encode(property)
            --发布消息
            mqtt_pub(

'/sys/'..productKey..'/'..deviceName..'/thing/event/property/post',
            0,
            0,
            jsonStr
            )

        end

        if screen == screen_ac_control
        then
            if control == 10
            then
                local PowerSwitchName = 'mode'
                params[PowerSwitchName] = value
                property['params'] = params
                local jsonStr = cjson.encode(property)
                mqtt_pub(
```

```
'/sys/'..productKey..'/'..deviceName..'/thing/event/property/post',
              0,
              0,
              jsonStr
              )
          end
       end
    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)

      --*********************************************************************
      --位置:     空调控制器
      --*********************************************************************
      if screen == screen_ac_control
      then
         if control == 1 or
            control == 3 or
            control == 6 or
            control == 7 or
            control == 10
         then
            --屏发布消息
            aliyun_pub(screen,control,value)
         end
      end
   end
```

核心 API 函数

1) mqtt_pub (topic,qos,retain,message)

发送数据到云端，传送应用消息

- topic ：应用程序消息的主题
- qos ：申请消息的服务质量
- retain ：保留标志
- message：消息内容

本例程中，假设用户修改模式，则将数据 Json 格式上传到云端，，数据原型如

下所示：

```
AT+MPUB="/sys/a1D2E9vaSuZ/PlYPGnoYWeUJujyOli2a/thing/event/property/post",0,0
,"{\22version\22:\221.0\22,\22params\22:{\22mode\22:0,\22targetTemperature\22
:27},\22id\22:\2261\22,\22method\22:\22thing.event.property.post\22}"
```

通过 Json 库函数 `jsonStr = cjson.encode(payload)`打包成 Json 格式的数据，则 `jsonStr` 为上传的数据。

5.4 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 5-4 所示。编译成功后点击菜单栏中【工具】→【量产向导】，如图 5-5 所示；

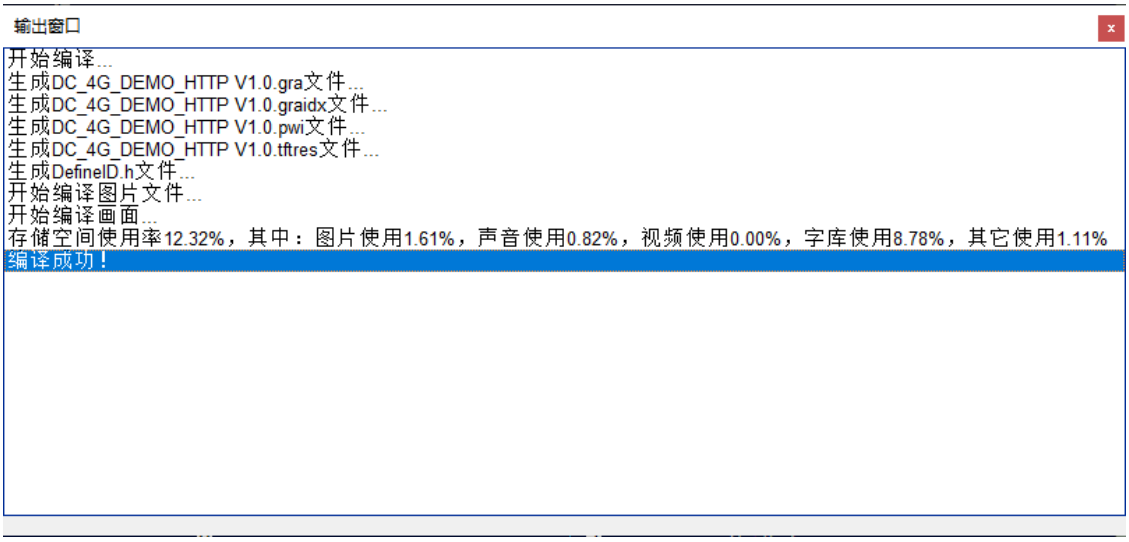


图 5-4 编译成功

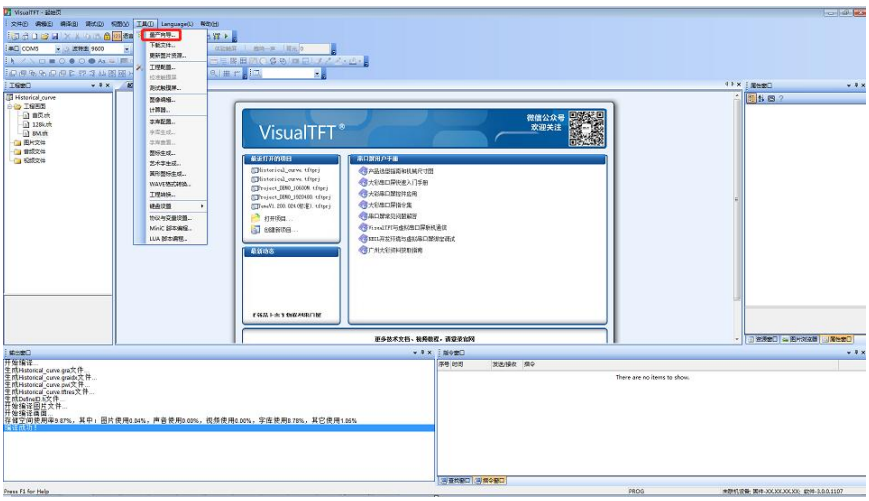


图 5-5 量产向导

在菜单栏中，文件→打开工程目录，在‘dciot_build’目录的‘private’文件夹拷贝到 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。

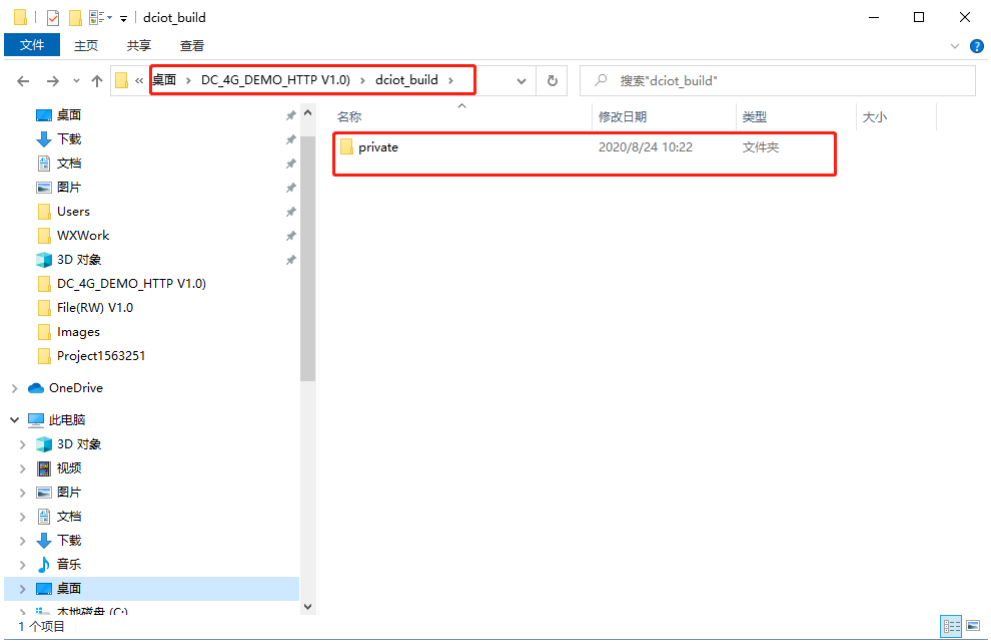


图 5-6 量产向导

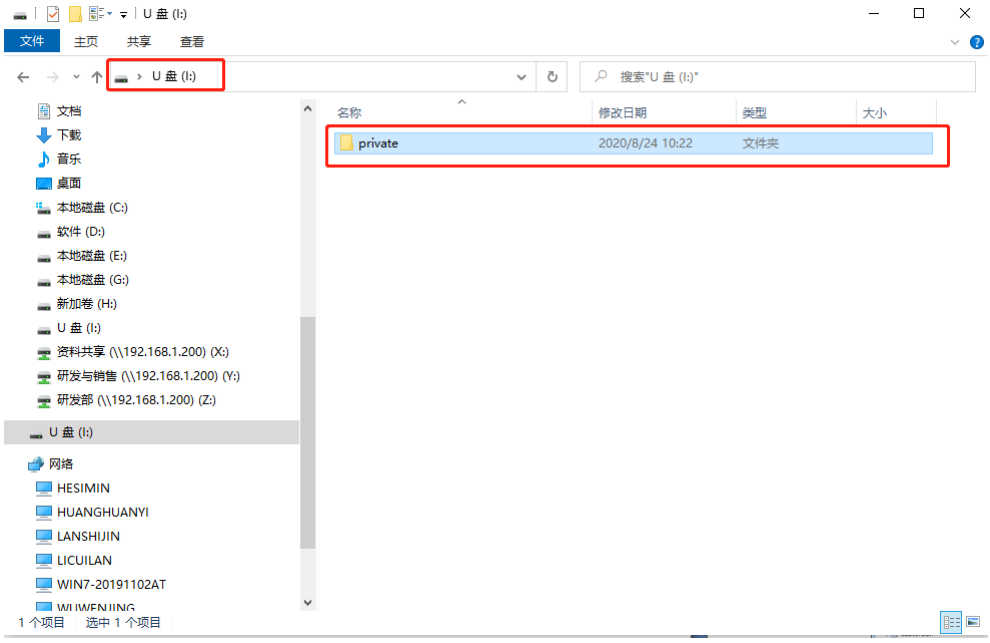


图 5-7 拷贝到 SD 卡

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。