



# A76xx\_Series\_Open\_SDK\_ 用户开发与DEMO使用指南

LTE Module

## **SIMCom Wireless Solutions Limited**

Building B, SIM Technology Building, No.633, Jinzhong Road

Changning District, Shanghai P.R. China

Tel: 86-21-31575100

support@simcom.com

www.simcom.com

<b>文档名称:</b>	A76xx_Series_Open_SDK_用户开发与 DEMO 使用指南
<b>版本:</b>	1.00.01
<b>日期:</b>	2021.5.6
<b>状态:</b>	已发布

## 版权声明

本手册包含芯讯通无线科技（上海）有限公司（简称：芯讯通）的技术信息。除非经芯讯通书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权，芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通，任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

### 芯讯通无线科技(上海)有限公司

上海市长宁区金钟路 633 号晨讯科技大楼 B 座 6 楼

电话：86-21-31575100

邮箱：simcom@simcom.com

官网：www.simcom.com

### 了解更多资料，请点击以下链接：

<http://cn.simcom.com/download/list-230-cn.html>

### 技术支持，请点击以下链接：

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 [support@simcom.com](mailto:support@simcom.com)

版权所有 © 芯讯通无线科技(上海)有限公司 2020，保留一切权利。

## 前言

感谢使用 SIMCom 提供的 A76xx 系列模块。该文档适用于 SIMCom A76xx Series, including A76xx-xxxx, A7620。用于说明 Open SDK 二次开发环境的使用及编译方法。

使用前请仔细阅读用户手册，您将领略其完善的功能和简洁的操作方法。

此模块主要用于语音或者数据通讯，本公司不承担由于用户不正常操作造成的财产损失或者人身伤害责任。请用户按照手册中的技术规格和参考设计开发相应的产品。同时注意使用移动产品应该关注的一般安全事项。在未声明之前，本公司有权根据技术发展的需要对本手册内容进行修改。

SIMCom  
Confidential

## 版本历史

日期	版本	章节	变更描述
V1.00.00	2020.7.6		新版本
V1.00.01	2021.5.6		更新文本支持模块 系列为 A76xx

SIMCom  
Confidential

# 目录

版权声明 .....	1
版本历史 .....	3
目录.....	4
<b>1 引言 .....</b>	<b>5</b>
<b>2 用户新增文件 .....</b>	<b>5</b>
2.1 添加编译目录.....	5
2.2 目录下新增文件.....	6
2.3 在 APP 中添加 Task .....	6
<b>3 Demo 介绍及使用 .....</b>	<b>7</b>
3.1 DEMO UI 界面介绍 .....	8
3.2 DEMO 操作方法 .....	8

# 1 引言

本文档详细介绍了用户如何添加代码以及 demo（UI 界面）的使用方法，可以让用户快速上手，增加开发效率。

## 2 用户新增文件

### 2.1 添加编译目录

当用户需要新添加目录的时候，需要在 sc\_app 目录下的 makefile 里面添加目录名。

```
#-----
# Configure compile directories
#-----
PACKAGE_DIRS:= \
    simcom_application \
    simcom_demo
    XXX 用户添加的目录名
```

图 1 sc\_app 路径下的 makefile

用户新增加的目录需要在 sc\_app 下的路径里面添加。

build	2020/7/31 16:02	文件夹	
script	2020/7/30 16:43	文件夹	
simcom_application	2020/7/30 13:53	文件夹	
simcom_demo	2020/7/30 13:54	文件夹	
simcom_lib	2020/7/30 16:43	文件夹	
build.bat	2020/7/30 15:33	Windows 批处理...	1 KB
Makefile	2020/7/30 15:33	文件	1 KB

图 2 sc\_app 的目录结构

## 2.2 目录下新增文件

执行完第一步后，在新增加的目录添加代码和 makefile。参考 simcom\_application 目录下的 makefile。

```

#-----
# Configure source code files
#-----
SRC_FILES+= \
    simcom_application.c
    xxx.c 在此添加用户的文件
#-----
# the include directories
#-----
INC_DIRS+= \
    -I${TOP_DIR}/simcom_lib/inc \
    在此添加需要依赖的头文件
#-----
# Configure compile flag
#-----
C_FLAGS :=
    在此添加编译选项
#-----
# Configure default macro define
#-----
DFLAGS+=
    在此添加宏定义

```

图 3 simcom\_application 目录下的 makefile

### NOTE

只要使用到提供的 API，则必须要在代码处引入 simcom\_api.h 的头文件。

## 2.3 在 APP 中添加 Task

用户需要在 sc\_app\simcom\_appication\simcom\_application.c 中的 Application 函数里添加 task。该函数将会在模块初始化期间被调用。

参考 sc\_app\simcom\_demo\demo\_helloworld.c 添加 task。添加 task 需要使用 sAPI\_TaskCreate，有关 task 操作的使用方法请参考文档 A76xx\_Series\_APIs\_Programming\_User\_Guide.docx。

```

#include "simcom_api.h" 必须包含simcom_api.h
sTaskRef helloWorldProcessor;
static UINT8 helloWorldProcessorStack[1024];

void sTask_HelloWorldProcessor(void* argv) task主函数
{
    sAPI_Debug("Task runs successfully");
}

void sAPP_HelloWorldDemo(void)
{
    SC_STATUS status = SC_SUCCESS;
    // 创建task的api, 与task相关操作请参考api介绍文档 task优先级
    status = sAPI_TaskCreate(&helloWorldProcessor, helloWorldProcessorStack, 1024, 150, "helloWorldProcessor", sTask_HelloWorldProcessor, (void *)0);
    if(SC_SUCCESS != status)
    {
        sAPI_Debug("Task create fail,status = [%d]", status);
    }
}

```

图 4 demo\_helloworld.c

```

static void Application(void * argv)
{
    // 入口函数
    /* Do not comment the initialization code below */
    unsigned long *apiTable = (unsigned long*)argv;
    if(apiTable != NULL)
    {
        get_sAPI(apiTable);
        sAPI_Debug("sc_Init successfull...");
    }
    /*End of initialization*/

    //sAPI_enableDUMP();
    sAPI_Debug("sAPI_Malloc:%p", sAPI_Malloc);
    sAPI_Debug("sAPI_Debug is OK...");
    sAPI_Debug("helloworld is finish...");

    sAPP_SimcomUIDemo();
    sAPP_UartTask();
    sAPI_Debug("TASK_TEST: uart");

    sAPP_HelloWorldDemo(); 在入口函数里创建task
}

```

图 5 simcom\_application.c

#### NOTE

请不要在 Application 函数里面做诸如 while(1)死循环等阻塞或耗时操作。

## 3 Demo 介绍及使用

A76xx 系列的二次开发 demo 采用了菜单(UI 界面)的形式,用户可以通过向串口输入参数来运行对应的 demo 程序。这种方式能够将 demo 的功能更直观的展现给用户,使用户快速了解 API 的使用方法。

### 3.1 DEMO UI 界面介绍

步骤 1: 打开串口的 Enhanced 口

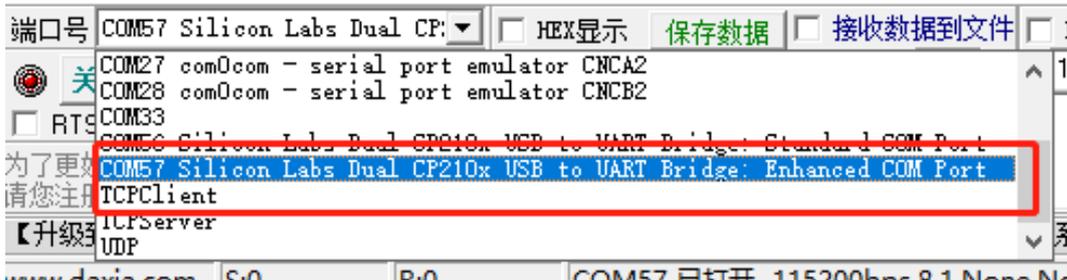


图 6 打开 Enhanced 口

步骤 2: 模块上电开机，即会出现 UI 界面

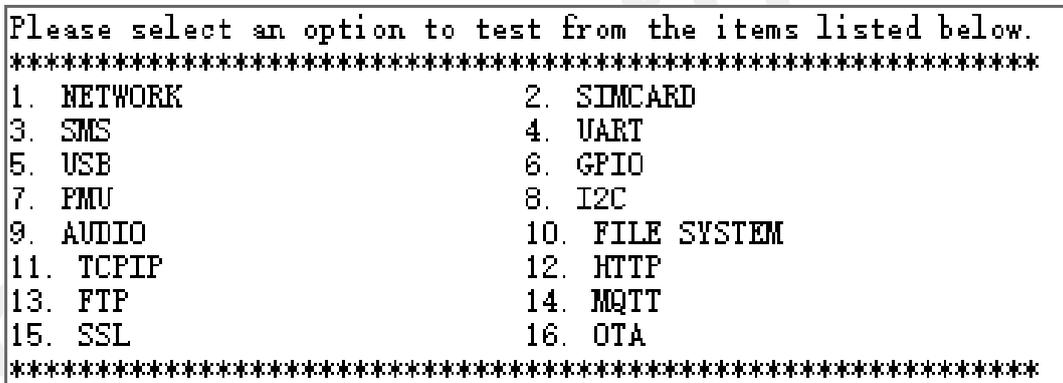


图 7 demo 界面

用户可以根据不同的选项来运行不同的 demo 程序。下面以 FTP demo 为例，讲解如果运行具体的 demo 应用程序

#### NOTE

当开启 Demo UI 展示后，Enhanced 口输入的所有信息都会被认为是对 demo UI 的操作。

### 3.2 DEMO 操作方法

从界面可以看到，FTP 的 demo 是第 13 个选项，我们在 Enhanced 中输入 13 后，就切换到 FTP demo 的界面。

```

Please select an option to test from the items listed below.
*****
1. Init                2. Login
3. Logout             4. Print work directory
5. List               6. Change work directory
7. Mkdir              8. Delete file
9. Delete folder      10. Get file size
11. download file     12. upload file
13. DeInit            99. back
*****

```

图 8 FTP demo 界面

目前，FTP 主要提供了以上 13 个功能的 demo 展示（最后一个选项为返回上一界面）。用户可以在初始化和登陆 FTP 服务器后，任意执行自己想要的功能。这里以选项 6（切换工作目录）为例，在该界面输入 6 后，会出现如下图所示的提示：

提示用户输入想要切换的目录名，当用户输入目录后，会提示切换目录的结果。

```

Please select an option to test from the items listed below.
*****
1. Init                2. Login
3. Logout             4. Print work directory
5. List               6. Change work directory
7. Mkdir              8. Delete file
9. Delete folder      10. Get file size
11. download file     12. upload file
13. DeInit            99. back
*****

Please input directory.

```

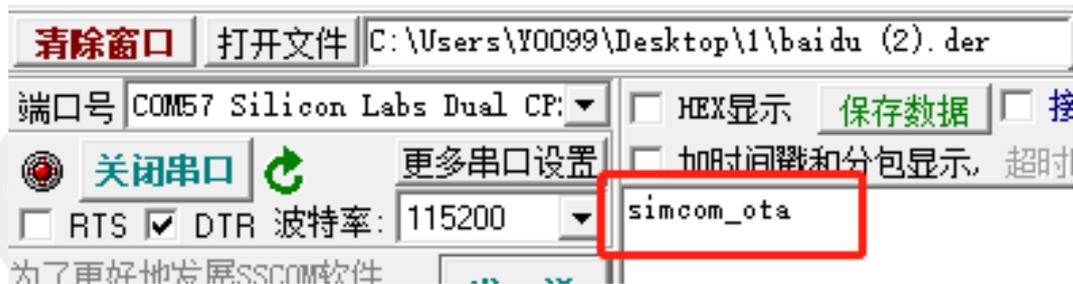


图 8 输入需要切换的目录名

这时候可以结合选项 4（打印当前工作目录）来查看结果。

```
current path = /simcom_ota
Please select an option to test from the items listed below.
*****
1. Init                2. Login
3. Logout              4. Print work directory
5. List                6. Change work directory
7. Mkdir               8. Delete file
9. Delete folder      10. Get file size
11. download file     12. upload file
13. DeInit            99. back
*****
```

图 9 查询当前目录的结果

SIMCom  
Confidential