

OPL1000

ULTRA-LOW POWER 2.4GHZ WI-FI + BLUETOOTH SMART SOC

TCP Client Demo User Guide



OPULINKS

<http://www.opulinks.com/>

Copyright © 2017-2018, OpuLinks. All Rights Reserved.

OPL1000-Demo-TCP-client-guide-R01 | Version V02

Date	Version	Contents Updated
2018-05-31	0.1	<ul style="list-style-type: none">Initial Release
2018-07-27	0.2	<ul style="list-style-type: none">Add processing: TCP server response ACK message to client
2018-12-14	0.3	<ul style="list-style-type: none">Add skip DTIM, timeout, high power, low power definition

TABLE OF CONTENTS

1. 介绍 1

1.1. 文档应用范围 1

1.2. 缩略语 1

1.3. 参考文献 1

2. TCP_Client 程序设计 2

2.1. 工作原理 2

2.2. API 调用 3

3. 验证 TCP Client 功能 5

3.1. 编译 TCP_client Example 5

3.2. 下载固件 5

3.3. PC 端执行网络调试助手 6

LIST OF FIGURES

Figure 1 : OPL1000 TCP client 示例网络连接图2

Figure 2: 网络调试助手中 TCP Server 参数配置3

Figure 3: OPL1000 作为 TCP Client 通信示例.....6

Figure 4: OPL1000 客户端收到的信息7

LIST OF TABLES

Table 1: 调用 API 说明3

1. 介绍

1.1. 文档应用范围

本文档介绍如何通过使用调用 SDK API 将 OPL1000 配置为 TCP Client，然后与同一网段的 TCP Server 进行连接和数据传输。

1.2. 缩略语

Abbr.	Explanation
AP	Wireless Access Point 无线访问接入点
APP	APPLication 应用程序
APS	Application Sub-system 应用子系统，在本文中亦指 M3 MCU
Blewifi	BLE config WIFI 蓝牙配网应用
DevKit	Development Kit 开发工具板
DTIM	Delivery Traffic Indication Message 传输指示消息
TCP	Transmission Control Protocol 传输控制协议

1.3. 参考文献

[1] DEVKIT 快速使用指南 OPL1000-DEVKIT-getting-start-guide.pdf

[2] Download 工具使用指南 OPL1000-patch-download-tool-user-guide.pdf

[3] SDK 应用程序开发指南 OPL1000-SDK-Development-guide.pdf

[4]电流功耗量测指南 OPL1000-Power-Consumption-Measurement-Guide.pdf

2. TCP_CLIENT 程序设计

2.1. 工作原理

TCP_Client 示例程序目录为

SDK\APS_PATCH\examples\ examples\protocols\tcp_client

它的工作过程为：

- 1 启动 WIFI 任务，将 OPL1000 配置为 Station 模式。
- 2 扫描可用的 AP。
- 3 如果指定连接的 AP SSID 在扫描到的 AP 列表中，则尝试去连接。
- 4 连接成功后，建立 Socket，和指定的 TCP Server 和端口号进行连接。
- 5 如果连接成功，则将一字符串发送给 TCP Server。

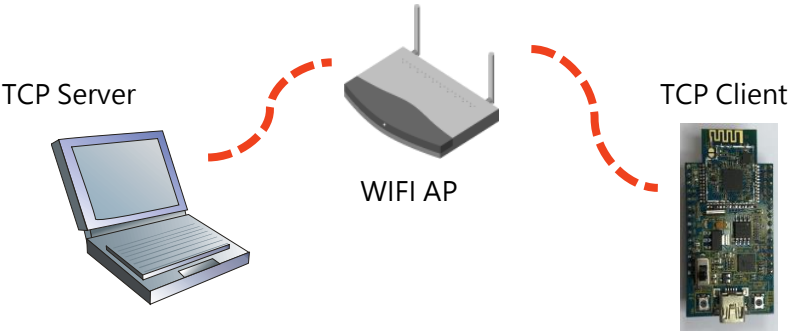
指定连接的 AP SSID 和 TCP Server，端口号，DTIM 等在 tc_client.h 文件中定义。如下所示：

```
#define WIFI_SSID          "Opulinks-TEST-AP"
#define WIFI_PASSWORD      "1234abcd"
#define TCP_SERVER_ADDR    "192.168.43.80"
#define TCP_SERVER_PORT    8181
#define DTIM_SKIP_COUNT   19
#define TCP_RECV_TIMEOUT   180
#define __RF_LP_MODE__
```

TCP server 和 OPL1000 建立 TCP 数据传输的网络拓扑如 Figure 1 所示。

OPL1000 以 Station 的角色连接到 WIFI AP，PC 端也作为 Station 连接到 WIFI AP，这样 PC 和 OPL1000 接入到同一个 AP，处于一个局域网网段。

Figure 1: OPL1000 TCP client 示例网络连接图



PC 端执行网络调试助手程序 NetAssist.exe（路径 Demo\TCP_Client）。在网络设置对话框，选择 TCP Server 作为协议类型, 填入 WIFI AP 分配给 PC 的 IP 地址，在本例中为 192.168.43.80。本地端口号可以任意取一个数，但最好不要使用已知、通用的端口号，例如 8080。在本例中端口号定义为 8181。

TCP server 的 IP 地址和端口参数在 TCP_Client 例程中赋值给宏定义 TCP_SERVER_ADDR 和 TCP_SERVER_PORT。

Figure 2: 网络调试助手中 TCP Server 参数配置



2.2. API 调用

TCP_Client 例程中使用到的 API 说明如表 Table 1 所示。

Table 1: 调用 API 说明

API 接口	API 说明
wifi_register_event_handler	注册内部 WIFI 事件句柄

API 接口	API 说明
wifi_event_loop_init	初始化事件处理回调函数，本例中定义为 wifi_event_handler_cb
wifi_init	初始化 WIFI 任务所用堆栈以及 wifi 初始化完成事件句柄
wifi_set_config	设置 OPL1000 WIFI 工作模式
wifi_start	启动 WIFI 任务
osThreadCreate	创建用户应用进程，进程入口为 user_wifi_app_entry
wifi_event_handler_cb	定义收到 WIFI 相关事件消息 ID 时对应的处理操作
wifi_do_scan	扫描可用的无线接入点
wifi_connection	如果指定的 AP 在扫描到的 AP 列表中，则连接它
lwip_net_start	启动 lwip 网路协议栈
lwip_network_init	Tcpip 协议栈和网络接口初始化操作
lwip_net_ready	等待连接并从 AP 获取动态分配 IP 地址
socket	创建 Socket 句柄
connect	连接指定的 TCP Server 和端口
write	向 TCP Server 发送数据

3. 验证 TCP CLIENT 功能

3.1. 编译 TCP_client Example

编译 TCP client 示例工程包括三个步骤：

Step1: 首先确定用户需要连接的 WIFI AP，将其 SSID 和访问密码填写到 tcp_client.h 文件的宏定义 WIFI_SSID、WIFI_PASSWORD 中。如果 AP 是开放接入没有密码，则将 WIFI_PASSWORD 定义为空字符。

```
#define WIFI_SSID        "Opulinks-TEST-AP"
#define WIFI_PASSWORD    "1234abcd"
```

Step2: 将 PC 连接 WIFI AP，可以使用 ipconfig 命令查看 WIFI AP 分配给 PC 的 IP 地址。将 PC 的 IP 信息填入到宏定义 TCP_SERVER_ADDR 和 TCP_SERVER_PORT 中。

```
#define TCP_SERVER_ADDR  "192.168.43.80"
#define TCP_SERVER_PORT  8181
```

Step3: 添加 DTIM 参数，由 AP 通过设置 DTIM 的间隔，根据这个间隔发送组播流量；19 (唤醒时间为 $20 \times 0.2\text{ms} = 3\text{s}$)，19 代表 OPL1000 跳过的值，第 20dtim 才会被接收，其中 0.2ms 代表 AP 的 beacon interval time：

```
#define DTIM_SKIP_COUNT 19
```

Step4: 添加 timeout 参数：

```
#define TCP_RECV_TIMEOUT 180
```

Step5: 定义此时 RF 的状态，如果是 High power 状态，则不需要定义，默认状态 High Power，如果定义 Low Power，则需要添加如下参数：

```
#define __RF_LP_MODE__
```

Step6: 使用 Keil C 编译 TCP client 工程。工程文件路径：

SDK\APS_PATCH\examples\protocols\tcp_client\opl1000_app_m3.uvprojx

Keil C 工具设置以及编译过程可以参考文献[3] [SDK 应用程序开发指南](#)。

3.2. 下载固件

编译成功后在 SDK\APS_PATCH\examples\protocols\tcp_client\Output\Objects 目录下产生 opl1000_app_m3.bin 文件。将其拷贝到 FW_binary 目录下，然后使用 download 工具将其和 m0 bin 文件合并，下载到 DEVKIT 中。下载工具的使用可参考文献[2] [Download 工具使用指南](#)。DEVKIT 板的使用可参考文献[1] [DEVKIT 快速使用指南](#)。

3.3. PC 端执行网络调试助手

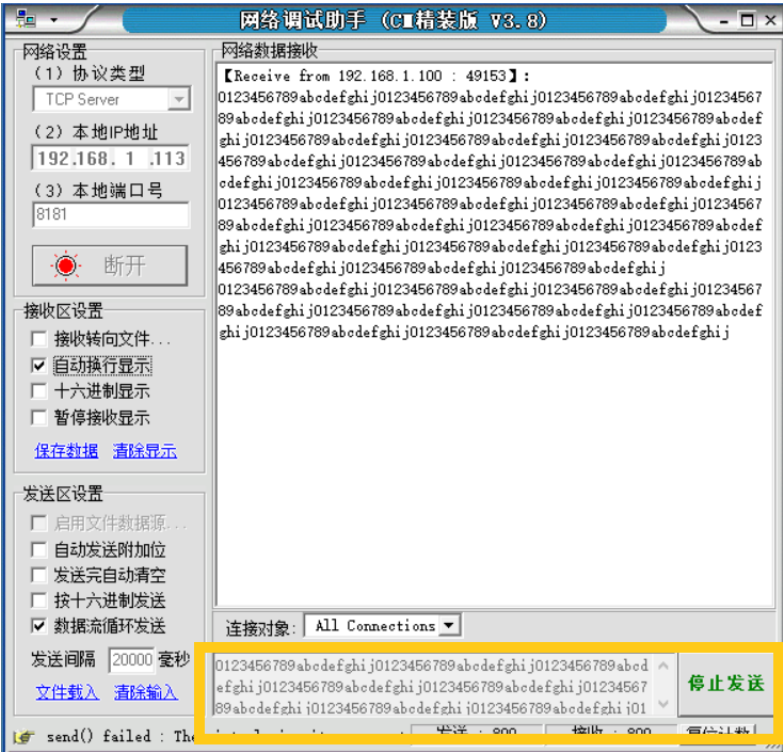
PC 端启动网络调试助手程序。选择 TCP_server 协议，填写 IP 地址和端口号。注意一定要和 tcp_client.h 文件中定义的 TCP_SERVER_ADDR，TCP_SERVER_PORT 一致。点击“连接”按钮，启动 TCP Server 服务。确保 WIFI AP 正常工作，将 OPL1000 DEVKIT 复位，它将自动尝试连接 WIFI AP，成功后，尝试和 TCP Server 进行数据通信。

根据以上环境，搭建 TCP Server，连接无线路由器之后，wifi 模块作为 TCP Client，Server 每两分钟发送一包数据至 Client，其中一个包大小为 200 字节。Client 串口打印数据回传至 TCP Server,测试 10 个小时的平均电流。功率计测电流的方法参见参考文献[4]电流功耗量测指南。DTIM 间隔设置为 4s。

其中，两百个字节的载入方式为，点击界面上的“文件载入”按钮，载入文件，文件中为 200 字节的测试数据。即设置每 120S 循环发送 200 字节的数据。

Figure 3 展示了通信成功的执行结果。在数据接收窗口会不断打印字符串，下图中圈出部分即为 OPL1000 发出的信息。

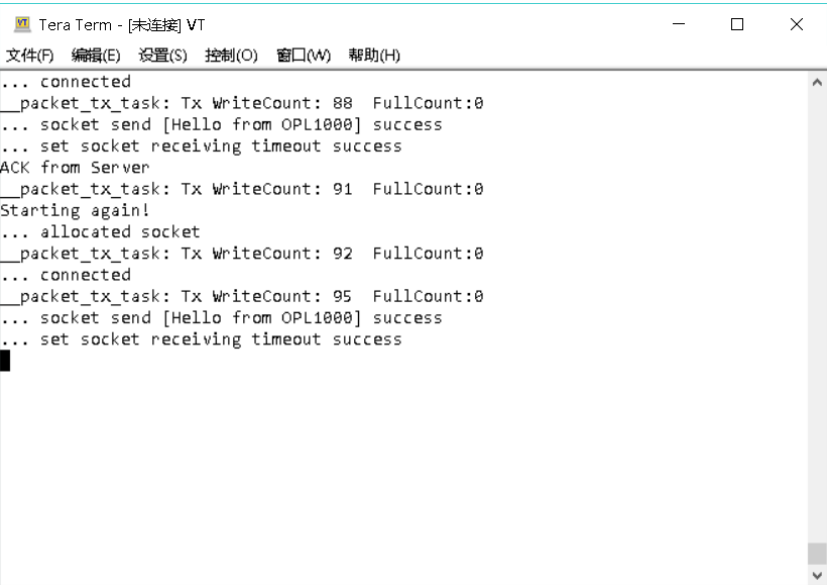
Figure 3: OPL1000 作为 TCP Client 通信示例



当收到来自 OPL1000 TCP 客户端的 200 个字节的数据，在网络调试助手软件的发送窗口中填入 “ACK from Server” 点击 “发送” 按钮，在 Client 端就会收到对应的字符串。

Figure 4 展示了 OPL1000 打印串口收到的消息。可见它收到了来自 Server 端的 “ACK from Server” 响应信息。

Figure 4: OPL1000 客户端收到的信息



为保证执行正确，在运行 “网络调试助手程序” 之前可以在 PC 端先用 Ping 命令验证连接 OPL1000 是否正常。如果成功，表明它们在同一网段。

```
C:\>ping 192.168.43.150

正在 Ping 192.168.43.150 具有 32 字节的数据:
来自 192.168.43.150 的回复: 字节=32 时间=133ms TTL=255
来自 192.168.43.150 的回复: 字节=32 时间=44ms TTL=255
来自 192.168.43.150 的回复: 字节=32 时间=162ms TTL=255
来自 192.168.43.150 的回复: 字节=32 时间=174ms TTL=255

192.168.43.150 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
    往返行程的估计时间(以毫秒为单位):
        最短 = 44ms，最长 = 174ms，平均 = 128ms
```

CONTACT

sales@Opulinks.com