

# OPL1000

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

## Iperf 网络性能测量工具



OPULINKS

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

Copyright © 2018, OpuLinks. All Rights Reserved.

---

OPL1000-Power-Consumption-Measurement-Guide-R01 | Draft Version

V03

Date	Version	Contents Updated
2018/11/14	0.1	<ul style="list-style-type: none"><li>Initial Release</li></ul>
2018/11/28	0.2	<ul style="list-style-type: none"><li>Add Close The Firewall Instruction</li></ul>
2019/02/15	0.3	<ul style="list-style-type: none"><li>Add help command</li></ul>

目录

1. 介绍 2

1.1. 文档应用范围 2

1.2. 缩略语 2

1.3. 参考文献 2

2. IPERF 网络性能测量环境 3

2.1. PC 端环境架设 4

3. IPERF 网络性能测量指令 6

3.1. DevKit Wi-Fi 指令 6

3.2. Iperf 指令 9

3.3. Iperf 指令 – 上行带宽测试实例 11

3.4. Iperf 指令 – 下行带宽测试实例 12

3.5. Iperf 组合指令 12

LIST OF FIGURES

图 1 Iperf 网络性能测量环境示意图 ..... 3

图 2 help 指令 ..... 6

图 3 scan 指令 ..... 8

图 4 connect 指令 ..... 8

图 5 query 指令 ..... 9

图 6 Wi-Fi Tx 固定数据传输率 ..... 9

图 7 Iperf 使用说明 ..... 9

图 8 测试上行频宽环境 ..... 11

图 9 Iperf client ..... 11

图 10 测试下行频宽环境 ..... 12

表 3-1 help 指令说明 ..... 7

表 3-2 Iperf 使用说明列表 ..... 10

表 3-3 Iperf UDP Test ..... 13

表 3-4 Iperf TCP Test ..... 13

# 1. 介绍

## 1.1. 文档应用范围

Iperf 网络性能测量工具是一个 TCP 和 UDP 的性能测量工具，能够提供测试最大 TCP 和 UDP 带宽性能，网络吞吐率信息，以及延迟抖动、丢包率、最大传输单元大小等统计信息，从而能够帮助我们测试网络性能，定位网络瓶颈。

## 1.2. 缩略语

Abbr.	Explanation
Wi-Fi	Wireless Fidelity
DTO	Dynamic Throughput Optimize

## 1.3. 参考文献

[1] OPL1000-DEVKIT-getting-start-guide.pdf

## 2. IPERF 网络性能测量环境

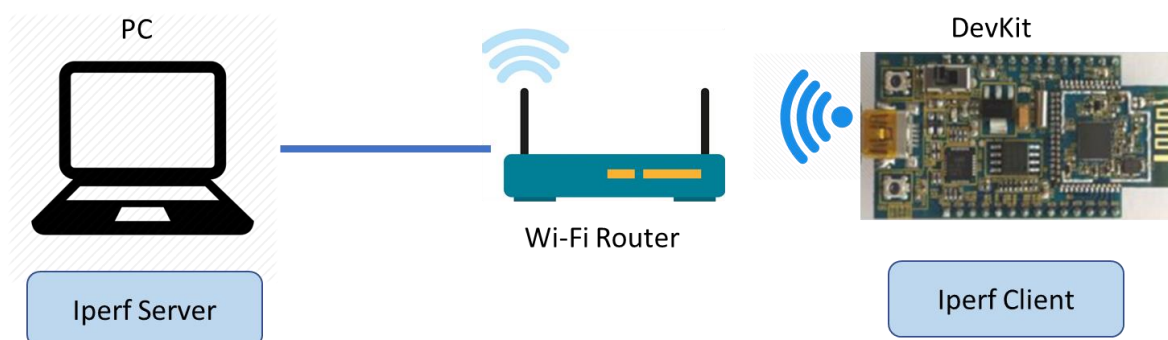


图 1 Iperf 网络性能测量环境示意图

Iperf 在工作时，测试的两端一方作为服务端，另一方为客户端。程序启动的命令相同，通过不同的参数来区别不同的工作方式运行。通常情况下先启动服务端，使 Iperf 监听在某个固定埠。然后在客户端执行相应的命令开始测试。

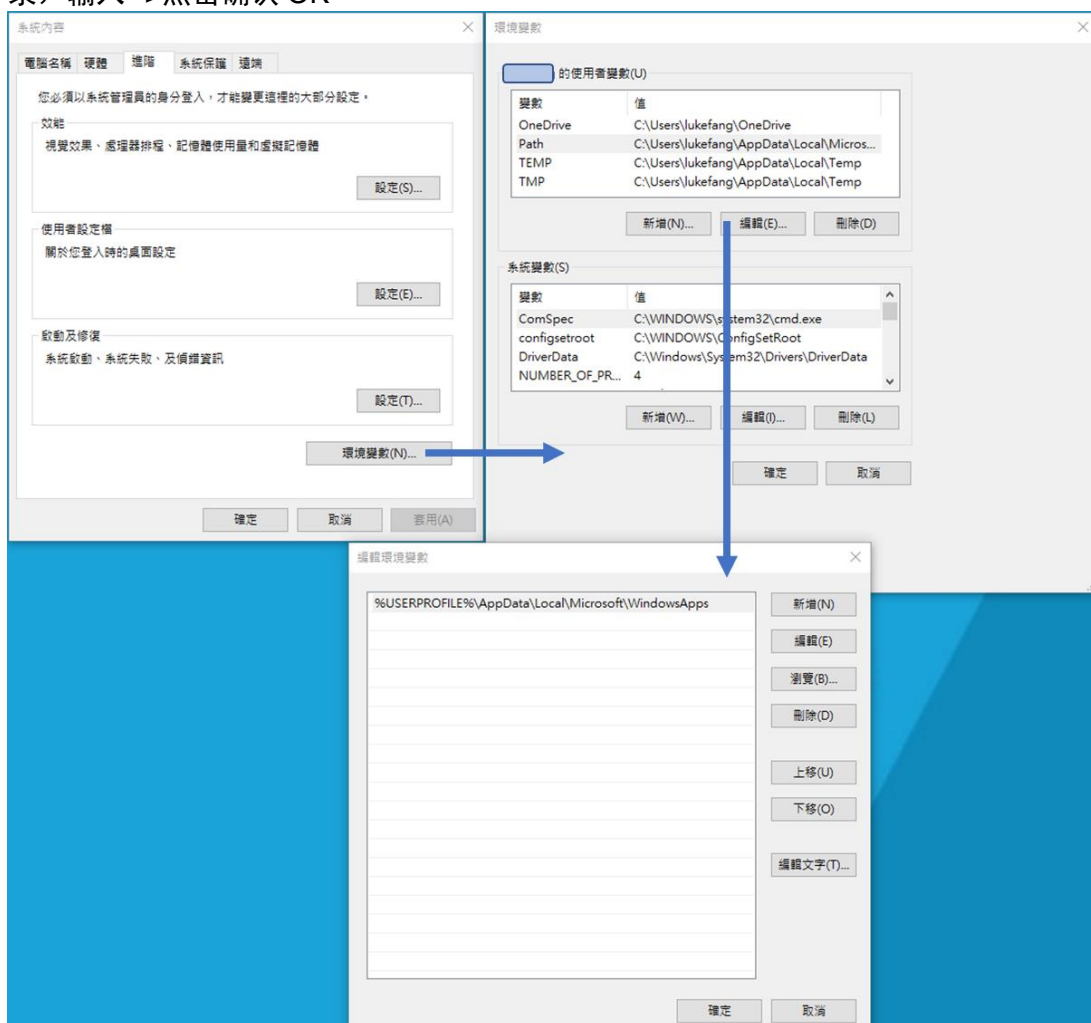
图 1 为 Iperf 测量环境的示意图，其中 PC 可以透过有线或无线的方式连接到 Wi-Fi 路由器，DevKit 可以透过无线的方式连接到 Wi-Fi 路由器。其中最主要的目的在于让 PC 及 DevKit 可以在同一个网段上面，可以互相 ping 的到对方。当彼此都可以互 ping 成功时，表示目前的测试环境已经设定完成，可以接下来的 Iperf 网络性能量测的实验。

首先可以先在 PC 端使用 Iperf-2.0.10\_exe 为服务端，在 PC 上进入命令提示字符执行，如 Iperf -s。这样 Iperf 将以服务端方式启动，并且通过 TCP 的 5001 埠号监听，监听端口号也可以透过指令做更改。DevKit 本身当成客户端，详细的指令描述会在下一个章节做完整的描述。

## 2.1. PC 端环境架设

Iperf 工具安装：

- 在 C 盘目录下新建，iperf 文件夹（C:\iperf），将附件的解压包，放到此文件夹下。（注意保证解压后的工具名称为 iperf.exe）。
- 构建环境变量：右键计算机属性 -> 点击高级系统设置 -> 点击环境变量 -> 在系统变量下找到 Path 变量，点击编辑 -> 点击新建，将 iperf 工具目录（根据用户安装目录）输入--> 点击确认 OK

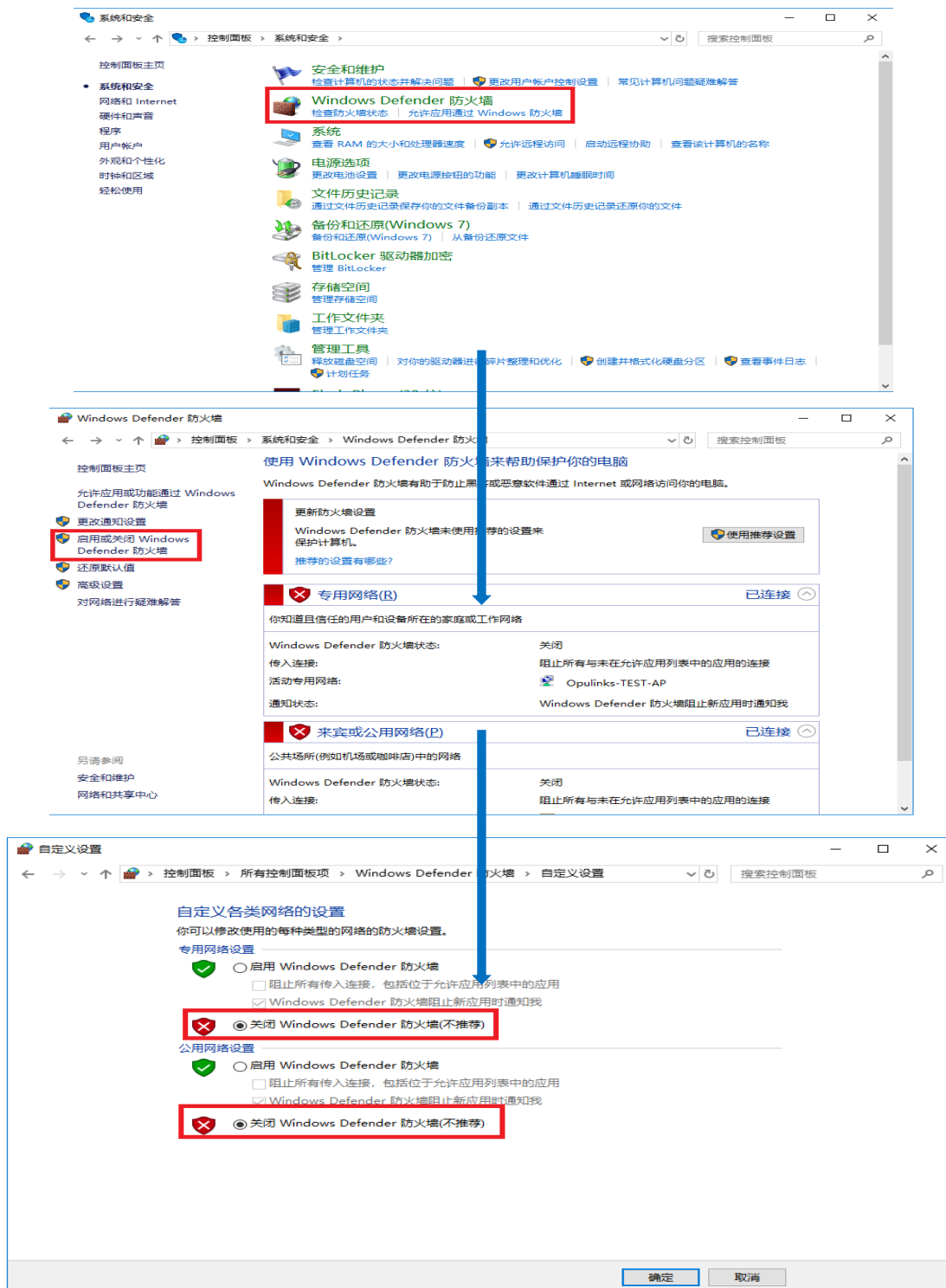


- 环境变量检查：打开 cmd 窗口输入 iperf -v 如果出现非系统命令，则环境变量构建错误，如果出现版本号，请保证当前工具版本为 2.0.10。

关闭 PC 防火墙：

- 打开 PC 上的控制面板，在系统和安全目录下点击 Windows Defender 防火墙

- b) 点击启动或关闭 Windows Defender 防火墙  
c) 勾选关闭选项，点击确认即可。





### 3. IPERF 网络性能测量指令

#### 3.1. DevKit Wi-Fi 指令

DevKit 上在 AT 串口中敲入 help 会出现如图 2 所示的指令，其说明可以参考表 3-1。

```
>help
I cmd_wifi: wifi Part:
I cmd_wifi:   Scan AP      : scan
I cmd_wifi:   Connect AP   : connect SSID Passphrase
I cmd_wifi:   Query States : query
I cmd_wifi:   Set Tx Data rate : wifi_data_rate TYPE
I cmd_wifi: Iperf Part:
I cmd_wifi:   help      : iperf -h
```

图 2 help 指令

Wi-Fi Part	
Scan AP (扫描 AP 列表)	scan
Connect AP (连上 AP)	connect [SSID] [Passphrase]
Disconnect the link with AP (与 AP 断开连接)	disconnect
Query States (查询目前 Wi-Fi 状态)	query
Set Tx Data rate (设定 Wi-Fi Tx 的数据传轮量， 0:DTO, 1:1M, 2:2M, 3:5.5M, 4:11M)	wifi_data_rate <TYPE>
Get current setting of STA mac address(获取当前的 STA mac 地址)	sta_mac
Set STA mac address. Format likes XX:XX:XX:XX:XX:XX.(设置 station mac 地址，设置完后需要重启)	sta_mac set [MAC_ADDR]
Iperf Part	
Help (Iperf 使用说明指令)	iperf -h
Common Part	

Show current SDK version (查看当前 SDK 版本)	version
Reboot device (重启设备)	reset
Stop ping command (停止 ping 命令)	ping s
Start ping command. Default send 3 ICMP packets to target HOST. 其中: [IP] - The target HOTS IP address (目标主机的 IP 地址) [Pkt counts] - which total of packet numbers to be send. (包的数量) [Pkt len] - which packet size per packet. (每个包的大小) [Recv timeout] - which timeout per packet. (接收延时) [Ping period] - which ping interval time. (ping 的时间间隔)	ping [IP] [Pkt counts] [Pkt len] [Recv timeout] [Ping period]

表 3-1 help 指令说明

DevKit 与 AP 联机的指令。首先敲入如图 3 所示之指令 scan，之后会出现 AP 列表。AP 列表当中会显示出 AP 的 SSID、当前此 AP 所使用的 channel number，以及当前此 AP 的信号强度对于 DevKit 而言。

```
>scan
I cmd_wifi: sta scanning
Wi-Fi Scan Done
I iperf: SSID=       Oplinks-ESP32, Channel= 1, RSSI=-66
I iperf: SSID=       D-Link_DIR-612, Channel= 1, RSSI=-64
I iperf: SSID=Sapido_RB-1602G3_df1ff, Channel= 6, RSSI=-21
I iperf: SSID=       TP-LINK_ccctv1, Channel= 3, RSSI=-69
I iperf: SSID=       TP-LINK_8688, Channel= 4, RSSI=-67
I iperf: SSID=       D-Link_DIR-816, Channel= 6, RSSI=-67
I iperf: SSID=       dlink-03AA, Channel= 6, RSSI=-74
I iperf: SSID=       CCTV-B, Channel= 4, RSSI=-76
I iperf: SSID=       netis_3B10DA, Channel= 6, RSSI=-67
I iperf: SSID=       ASUS_18, Channel=11, RSSI=-70
```

图 3 scan 指令

开始与 AP 进行联机的动作。首先选定一个 AP，然后敲入 connect 指令，如图 4 所示。connect 其后接 AP 的 SSID 与 AP 所设定的密码。当 DevKit 有与 AP 连上之后，会看到 Wi-Fi connected 的反馈字符串呈现在 AT 串口之上。随后 DevKit 拿到 IP 之后，会看到 Wi-Fi Got IP 的反馈字符串呈现在 AT 串口之上，并呈现 DevKit 所拿到的 IP 地址。

```
>connect Sapido_RB-1602G3_df1ff 12345678
I cmd_wifi: connecting to Sapido_RB-1602G3_df1ff
wifi connected
Wi-Fi Got IP

interface: st1
ip mode : DHCP
ip       192.168.1.100
netmask  255.255.255.0
gateway  192.168.1.1
```

图 4 connect 指令

DevKit 拿到 IP 之后，此时在去 query DevKit 目前 Wi-Fi 联机的状况时，才会出现如图 5 的讯息，呈现在 AT 串口之上。DevKit 拿没有到 IP 的时候，此时在去 query DevKit 目前 Wi-Fi 联机的状况时，会呈现 Disconnected 的讯息呈现在串口之上。

```
>query
I cmd_wifi: query wifi states
I cmd_wifi: Wifi Connected, AP SSID=Sapido_RB-1602G3_df1ffe
```

图 5 query 指令

用户也可以设定 Wi-Fi Tx 的数据传输率，让 DevKit 用固定的数据传输率来测试网络的最大 TCP 和 UDP 带宽性能。理论上用户使用越高的数据传输率，得到的带宽性能数据越好。

```
>wifi_data_rate 4
I cmd_wifi: set wifi Tx fix data rate
I cmd_wifi: current setting : 4
I cmd_wifi: 0:DT0, 1:1M, 2:2M, 3:5.5M, 4:11M
```

图 6 Wi-Fi Tx 固定数据传输率

3.2. Iperf 指令

Iperf 使用说明列表如图 7 所示的指令，其说明可以参考表 3-2。

```
>iperf -h
I iperf_cli:

Client/Server:
  -a, --abort          abort iperf program
  -i, --interval #    seconds between periodic bandwidth reports (default 3 secs)
  -p, --port #        server port to listen on/connect to (default port 5001)
  -u, --udp            use UDP rather than TCP
  -s, --server        run in server mode

Client specific:
  -c, --client <host> run in client mode, connecting to <host>
  -d, --dualtest      Do a bidirectional test simultaneously
  -n, --num #[kmgKMG] number of bytes to transmit (instead of -t)
  -t, --time #        time in seconds to transmit for (default 30 secs)
```

图 7 Iperf 使用说明

Client / Server		
-a, --abort	有关 Iperf 程序说明	
-i, --interval	<number>	回报带宽报告周期(单位:秒)

-p, --port	<number>	指定连结的埠号
-u, --udp	使用 udp 测试，没有指定默认值为 tcp	
-s, --server	跑在伺服器端模式	
Client specific		
-c, --client	<host>	跑在客户端模式，链接到主机地址 <host>
-d, --dualtest	同时做双向测试	
-n, --num	<number>[kmgKMG] 'k'=Kbits/sec 'm'=Mbits/sec 'g' = Gbits/sec 'K'=KBytes/sec 'M'=MBytes/sec 'G'=GBytes/sec	传输固定的 bytes 数，取代 -t
-t, --time	<number>	传输多久，单位时间为秒

表 3-2 Iperf 使用说明列表

3.3. Iperf 指令 – 上行带宽测试实例

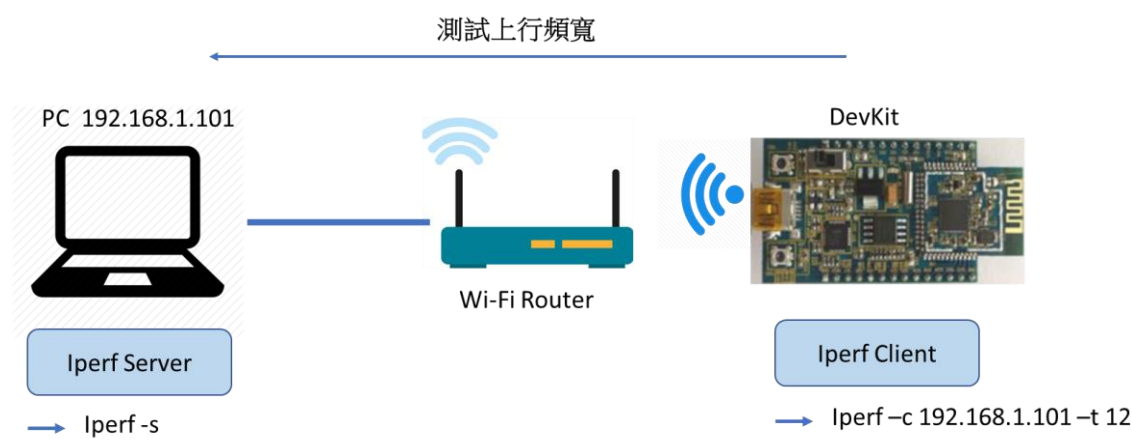


图 8 测试上行频宽环境

用户要开始测试 DevKit 的上行带宽时，可以开始使用 DevKit 做为 client 端。首先敲入如图 9 所示之指令，其指令意思可以参考表 3-2，如 iperf -c 192.168.1.101 -t 12。其中 -c 是指要跑在客户端模式，链接到的伺服器端为 192.168.1.101。-t 是指要量测多久的时间，单位时间为秒。此范例量测的时间维持 12 秒。回报带宽报告周期的默认值是每 3 秒，会回报量测数据在串口之上。

```
>iperf -c 192.168.1.101 -t 12
I iperf_cli: sip=192.168.1.100:5001 interval=3 time=12 amount=0
I iperf: iperf_reportTask create successful
I iperf: iperf_traf_client Task create successful

Interval      Transfer      Bandwidth
client connecting to 192.168.1.101, 5001
0- 3 sec      0.315 MBytes  0.882 Mbits/sec
3- 6 sec      0.403 MBytes  1.127 Mbits/sec
6- 9 sec      0.312 MBytes  0.873 Mbits/sec
9- 12 sec     0.372 MBytes  1.040 Mbits/sec
0- 12 sec     1.403 MBytes  0.980 Mbits/sec
I iperf: iperf report exit
I iperf: iperf client exit
```

图 9 Iperf client

3.4. Iperf 指令 – 下行带宽测试实例

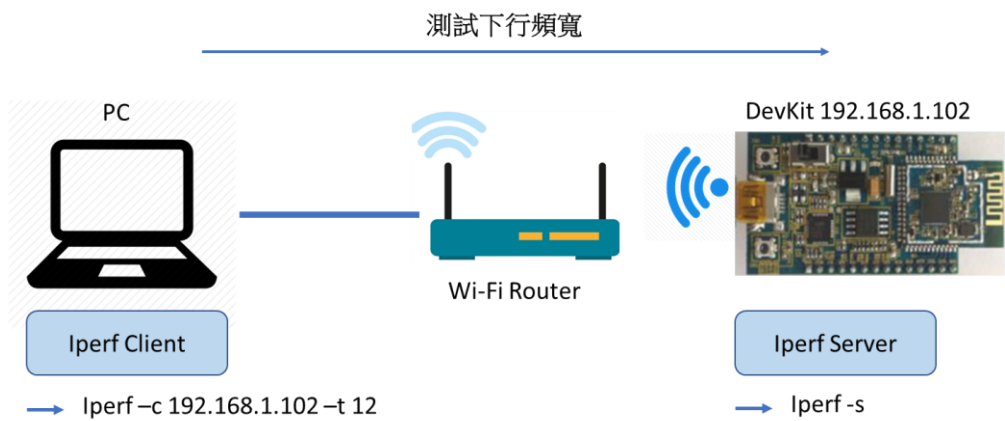


图 10 测试下行频宽环境

用户要开始测试 DevKit 的下行带宽时，可以开始使用 DevKit 做为 Server 端。首先敲入如图 10 所示之指令，其指令意思可以参考表 3-2，如 iperf -s，其中 -s 是指要跑在伺服器端模式。PC 端的指令可以参考图 10 所示。更多的测试组合可以参考 错误! 未找到引用源。节中，表 3-3 和 表 3-4 提供 Iperf 量测组合指令。

3.5. Iperf 组合指令

Iperf 的测试有很多种组合，表 3-3 和 表 3-4 提供 Iperf 量测组合指令。用户可以根据表中描述，清楚且快速的，把 DevKit 设置成 UDP 或 TCP 的 Iperf 测试。在此之前，建议使用着先参考**注意事项**，能使得 Iperf 在量测当中减少干扰，进而可以使得量测数据更贴近用户的量测环境。

注意事项：

1. 对于空口测试，需要保证测试环境的干扰尽可能小。一般 AP 较常使用的通道为 Channel 1、Channel 6 以及 Channel 11。用户可以尽量避开这几个通道。所以建议可将 AP 的通道先设置成 Channel 2 进行测试。
2. 对于 PC 上有多网卡的情形，建议先关闭其他不用之网卡。

	DevKit be UDP Server	DevKit be UDP Client
DevKit	iperf -s -u -t 180	iperf -c <PC IP Address> -u -p 6007 -t 180
PC	iperf -c <DevKit IP Address> -u -p 5001 -t 180 -i 5 -b 10M	iperf -s -u -p 6007 -i 3

表 3-3 Iperf UDP Test

	DevKit be TCP Server	DevKit be TCP Client
DevKit	iperf -s -t 180	iperf -c <PC IP Address> -t 180
PC	iperf -c <DevKit IP Address> -p 5001 -t 180 -i 5	iperf -s -i 3

表 3-4 Iperf TCP Test



## CONTACT

[sales@Opulinks.com](mailto:sales@Opulinks.com)