

---

# **User Manual and Test Guide**

---

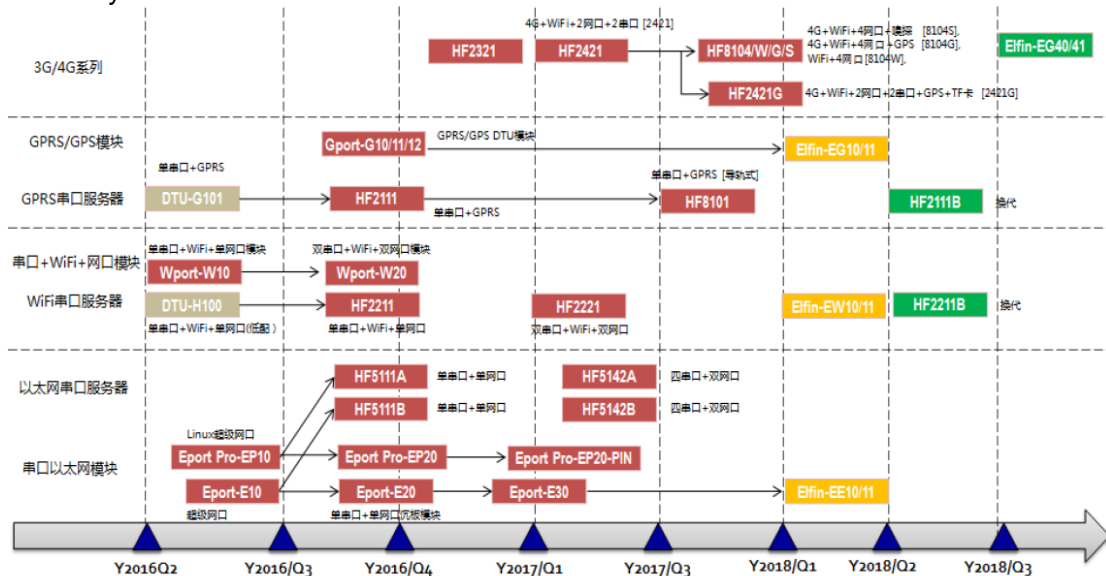
HTTP

## Content

1. DEVELOPMENT KIT INTRODUCTION .....	3
2. HARDWARE REQUIREMENTS .....	3
3. SOFTWARE REQUIREMENTS .....	3
4. HTTP tEST .....	4
4.1. Device Connection .....	4
4.2. HTTP Test .....	7
5. WIRESHARK PACKAGE CAPTURING SOFTWARE .....	10
5.1. Tool Introduction .....	10
Appendix: Contact Information .....	12

# 1. DEVELOPMENT KIT INTRODUCTION

This document is applicable to High Flying's IoT equipment. The specific supported models are as follows. This document introduces HF5111B. Other products are used in the same way.



# 2. HARDWARE REQUIREMENTS

- HF5111B 1 Pcs

## 4. HTTP TEST

### 4.1. Device Connection

Connect PC and HF5111B to router LAN.  
 Sending data in HTTP format to HTTP server (Set product socket to HTTP by IOTService software or webpage). When device socket works in HTTP mode. All received UART data will automatically transform to HTTP format (add HTTP header) and send to HTTP server. For the received HTTP data from HTTP server, it will automatically remove HTTP header and only output the data packet to UART.

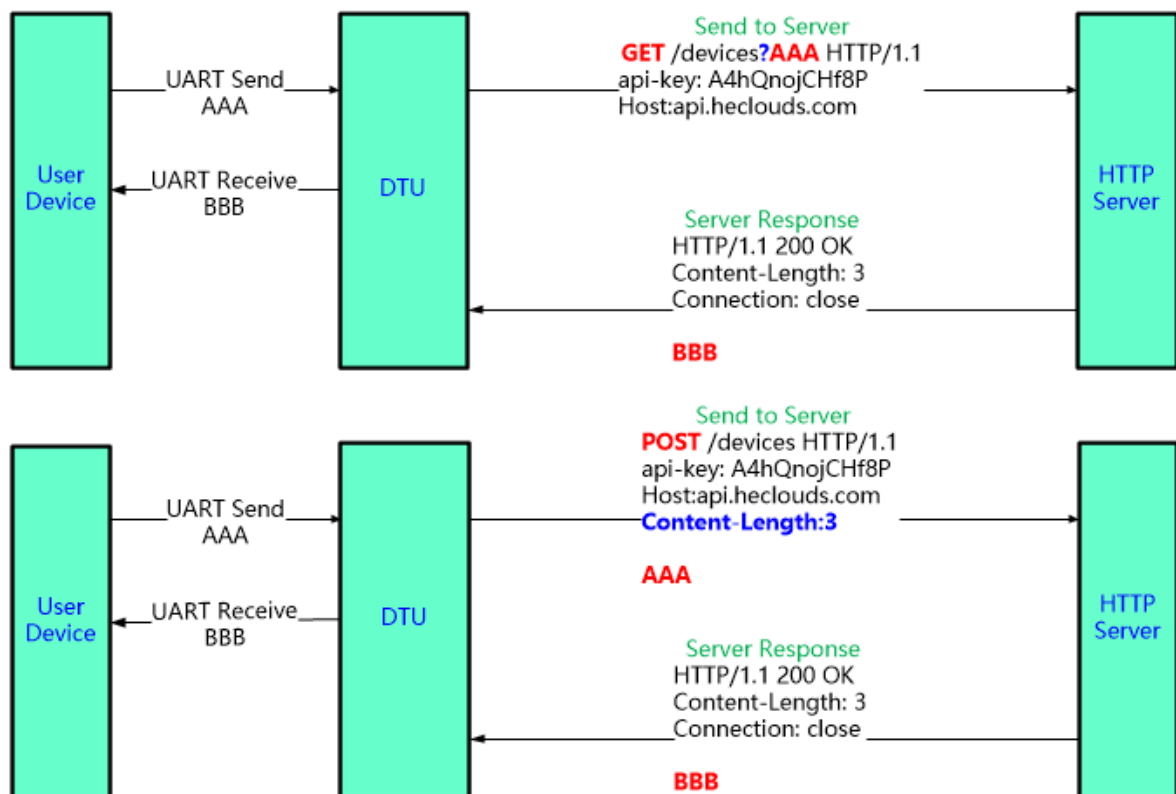


Figure 1. HTTP Request

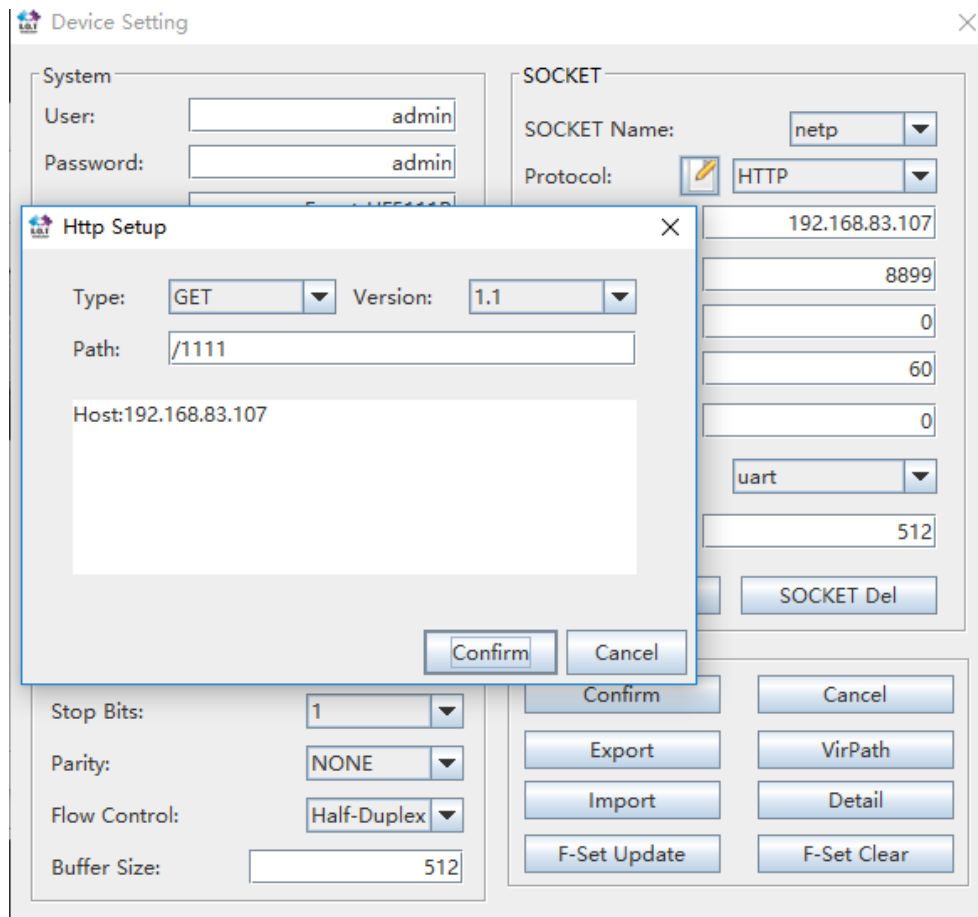


Figure 2. IOTService Configure

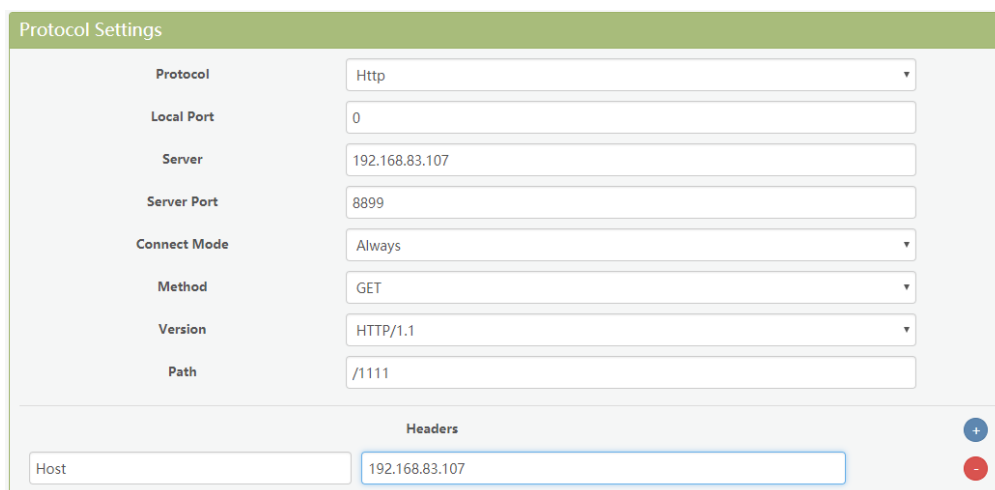


Figure 3. Webpage Configure

For GET request, the received UART packet AAA will put after the HTTP path (auto add “?” between path and parameters), for POST request, packet is put in the content (auto add Content-Length header information).

Product will send the below data to HTTP Server when UART receive “pppp” data for GET request.

GET /1111?pppp HTTP/1.1

Host: 192.168.83.107

Product will output “DDDDD” when get response from the HTTP server.

HTTP/1.1 200 OK

Server: nginx

DDDDD

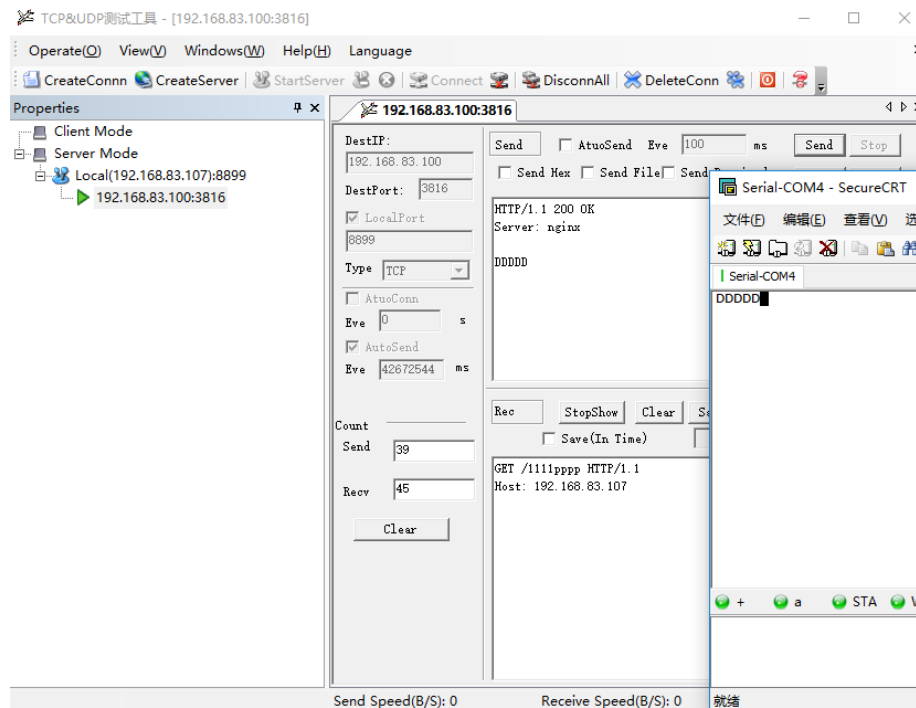


Figure 4. HTTP GET Request Example

Product will send the below data to HTTP Server when UART receive “pppp” data for POST request.

POST /1111 HTTP/1.1

Host: 192.168.83.107

Content-Length:4

pppp

Product will output “DDDD” when get response from the HTTP server.

HTTP/1.1 200 OK

Content-Length: 4

Connection: close

DDDD

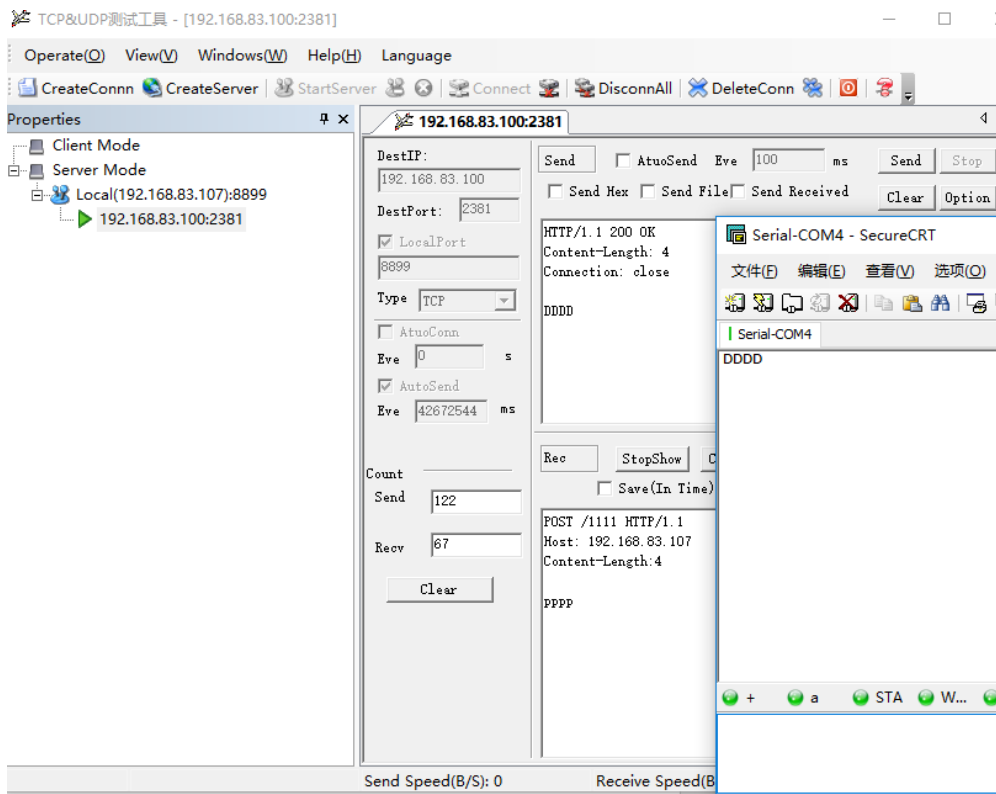


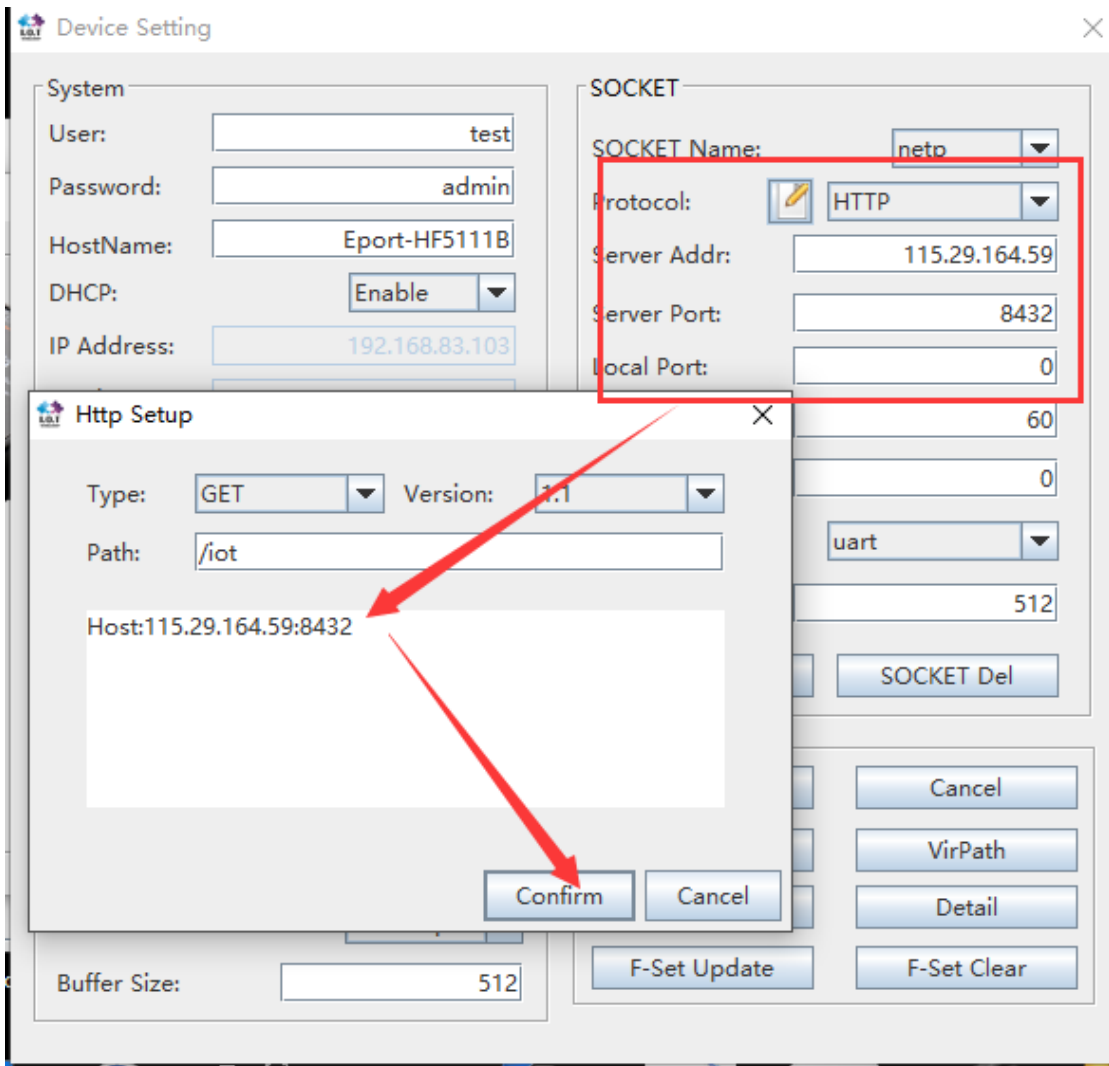
Figure 5. HTTP POST Request Example

## 4.2. HTTP Test

Step 1: Browser open <http://115.29.164.59:8432/iot?msg=123456788>, got the response as following:



Step 2: Input the HTTP parameters as the following steps.



Protocol: HTTP

Server Addr: Server address, IP or domain name.

Server Port: Server port.

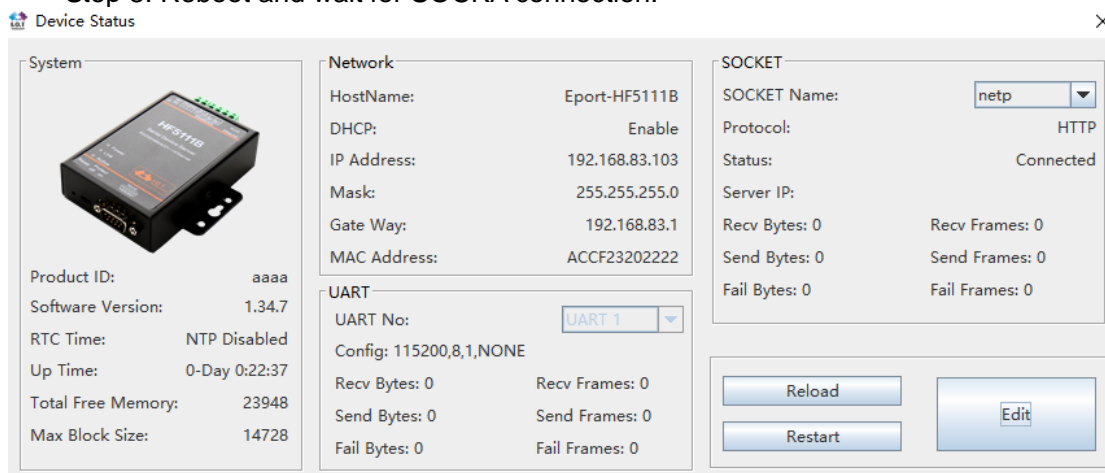
Type: HTTP Type, GET or POST.

Version: HTTP Version, 1.1.

Path: HTTP path

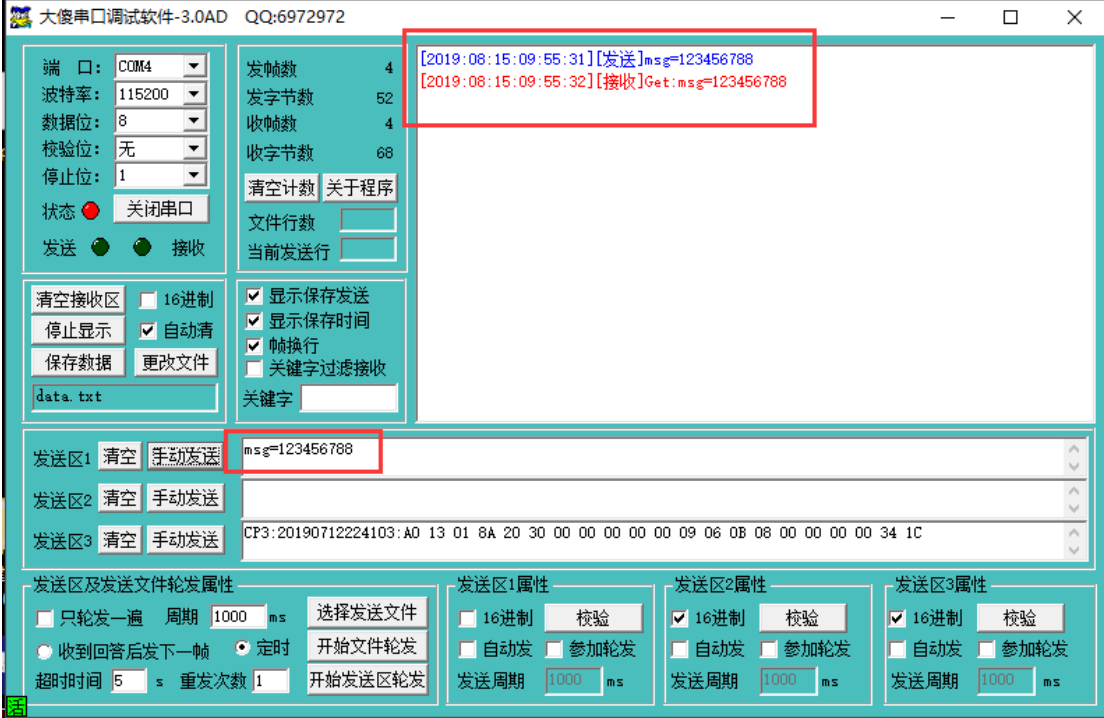
HTTP header input: Input HTTP header. Usually is Host information.

Step 3: Reboot and wait for SOCKA connection.





Step 4: UART send data id=1, and got response of the server.



大傻串口调试软件-3.0AD QQ:6972972

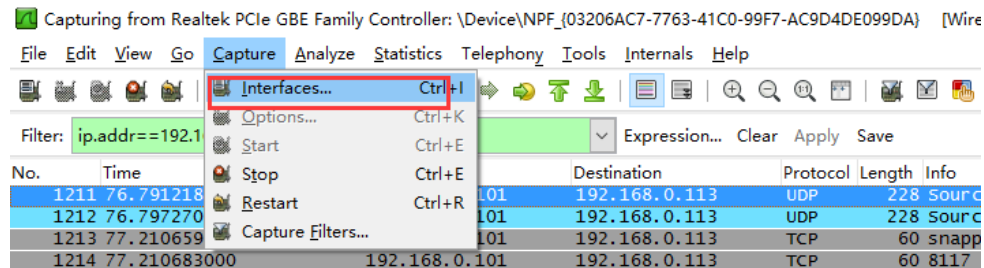
端口: COM4 发帧数: 4 [2019:08:15:09:55:31][发送]msg=123456788  
波特率: 115200 发字节数: 52 [2019:08:15:09:55:32][接收]Get:msg=123456788  
数据位: 8 收帧数: 4  
校验位: 无 收字节数: 68  
停止位: 1 清空计数 关于程序  
状态: 关闭串口 文件行数:   
发送: 接收 当前发送行:   
清空接收区  16进制  显示保存发送  
停止显示  自动清  显示保存时间  
保存数据 更改文件  帧换行  
data.txt 关键字过滤接收  关键字:   
发送区1 清空 手动发送 msg=123456788  
发送区2 清空 手动发送  
发送区3 清空 手动发送 CP3:20190712224103:A0 13 01 8A 20 30 00 00 00 00 00 09 06 0B 08 00 00 00 34 1C  
发送区及发送文件轮发属性 发送区1属性 发送区2属性 发送区3属性  
 只轮发一遍 周期 1000 ms 选择发送文件  16进制 校验  16进制 校验  
 收到应答后发下一帧  定时 开始文件轮发  自动发  参加轮发  自动发  参加轮发  
超时时间 5 s 重发次数 1 开始发送区轮发 发送周期 1000 ms 发送周期 1000 ms 发送周期 1000 ms

## 5. WIRESHARK PACKAGE CAPTURING SOFTWARE

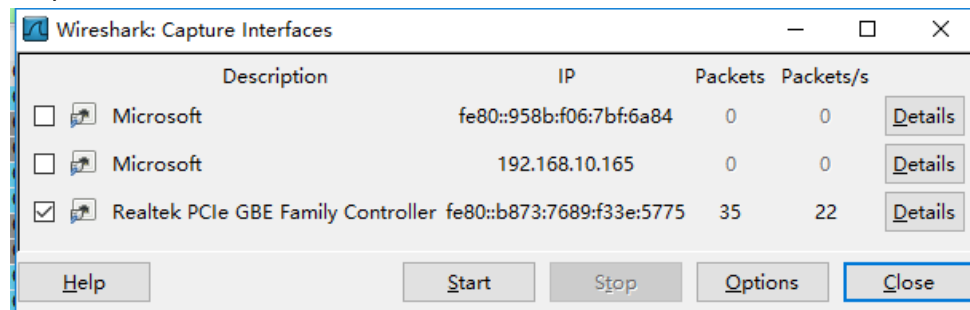
### 5.1. Tool Introduction

Wireshark can be used to analyze network packages about sending and receiving data. Please download and install this software from searching tools.

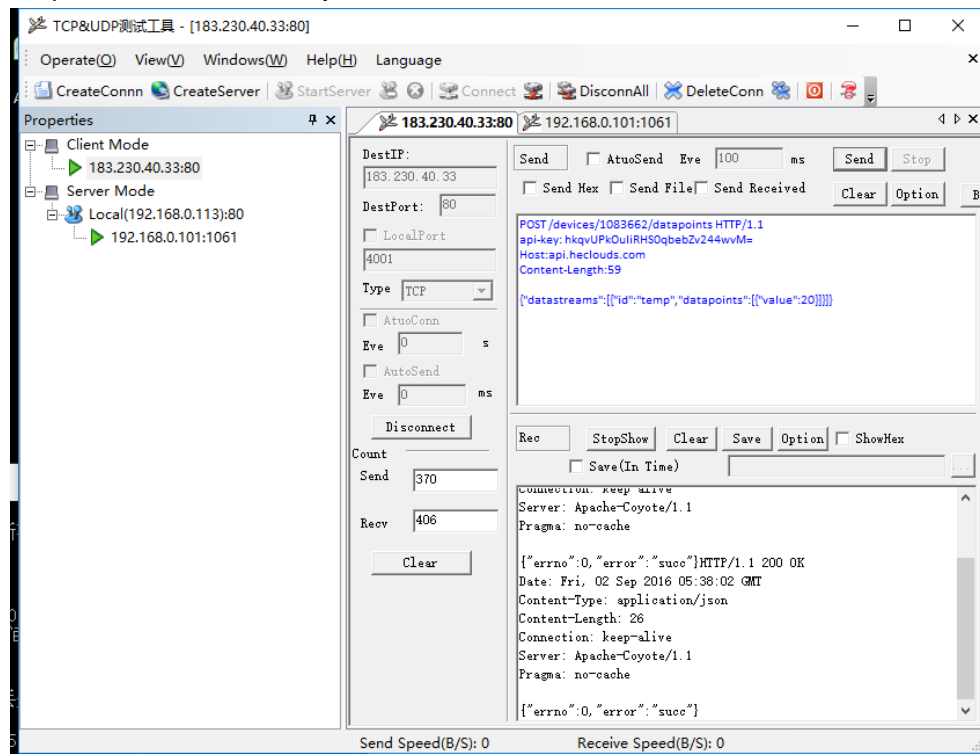
Step 1: Open wireshark tool and click interface tab control.



Step 2: Tick related PC network card and click Start button.



Step 3: Send test data by TCP&UDP tool.



Step 4: Input filter option. The following color marked packets are captured by this tool, which is from device uploading and server reposing.

Capturing from Realtek PCIe GBE Family Controller: \Device\NPF\_{03206AC7-7763-41C0-99F7-AC9D4DE099DA} [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help



Filter: ip.addr==183.230.40.33 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
122	26.764794000	192.168.0.113	183.230.40.33	TCP	66	56482 > http [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
123	26.823045000	183.230.40.33	192.168.0.113	TCP	66	http > 56482 [SYN, ACK] Seq=0 Ack=1 win=14600 Len=0 MSS=1332 SACK_PERM=1 WS=128
124	26.823206000	192.168.0.113	183.230.40.33	TCP	54	56482 > http [ACK] Seq=1 Ack=1 win=66560 Len=0
129	28.929724000	192.168.0.113	183.230.40.33	HTTP	230	POST /devices/1083662/datapoints HTTP/1.1
130	28.950916000	183.230.40.33	192.168.0.113	TCP	60	http > 56482 [ACK] Seq=1 Ack=186 win=15744 Len=0
131	28.984426000	183.230.40.33	192.168.0.113	HTTP	257	HTTP/1.1 200 OK (application/json)
134	29.013417000	192.168.0.113	183.230.40.33	TCP	54	56482 > http [ACK] Seq=186 Ack=204 win=66304 Len=0
215	49.860471000	192.168.0.113	183.230.40.33	ICMP	74	Echo (ping) request id=0x0001, seq=1/256, ttl=64
216	49.914650000	183.230.40.33	192.168.0.113	ICMP	74	Echo (ping) reply id=0x0001, seq=1/256, ttl=64
218	50.865150000	192.168.0.113	183.230.40.33	ICMP	74	Echo (ping) request id=0x0001, seq=2/512, ttl=64
219	50.918652000	183.230.40.33	192.168.0.113	ICMP	74	Echo (ping) reply id=0x0001, seq=2/512, ttl=64
235	34.551809000	192.168.0.113	183.230.40.33	TCP	54	56482 > http [FIN, ACK] Seq=186 Ack=204 win=66304 Len=0
236	34.604429000	183.230.40.33	192.168.0.113	TCP	60	http > 56482 [FIN, ACK] Seq=204 Ack=187 win=15744 Len=0
237	34.604520000	192.168.0.113	183.230.40.33	TCP	54	56482 > http [ACK] Seq=187 Ack=205 win=66304 Len=0
255	61.359199000	192.168.0.113	183.230.40.33	TCP	66	56490 > http [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
256	61.415183000	183.230.40.33	192.168.0.113	TCP	66	http > 56490 [SYN, ACK] Seq=0 Ack=1 win=14600 Len=0 MSS=1332 SACK_PERM=1 WS=128
274	67.178769000	192.168.0.113	183.230.40.33	HTTP	230	POST /devices/1083662/datapoints HTTP/1.1
275	67.270394000	183.230.40.33	192.168.0.113	HTTP	257	HTTP/1.1 200 OK (application/json)
276	67.299863000	192.168.0.113	183.230.40.33	TCP	54	56490 > http [ACK] Seq=186 Ack=204 win=66304 Len=0
579	132.237364000	183.230.40.33	192.168.0.113	TCP	60	http > 56490 [FIN, ACK] Seq=204 Ack=186 win=15744 Len=0
580	132.237443000	192.168.0.113	183.230.40.33	TCP	54	56490 > http [ACK] Seq=186 Ack=205 win=66304 Len=0
581	132.237577000	192.168.0.113	183.230.40.33	TCP	54	56490 > http [FIN, ACK] Seq=186 Ack=205 win=66304 Len=0
582	132.239350000	183.230.40.33	192.168.0.113	TCP	60	http > 56490 [ACK] Seq=205 Ack=187 win=15744 Len=0

