

# User Manual and Test Guide

---

Eport HTTP Protocol Application

Rev: 1.0

## Content

1. DEVELOPMENT KIT INTRODUCTION .....	3
2. HARDWARE REQUIREMENTS .....	3
3. SOFTWARE REQUIREMENTS .....	3
4. HARDWARE CONNECTION .....	4
4.1. Device Connection .....	4
4.2. TCP&UDP Test HTTP Server .....	4
4.3. Send HTTP data under Transparent Mode .....	5
4.4. HTTP Send data .....	6
5. WIRESHARK PACKAGE CAPTURING SOFTWARE .....	8
5.1. Tool Introduction .....	8
Appendix: Contact Information .....	10

### Version List:

V1.0 20160901: First Edition

## 1. DEVELOPMENT KIT INTRODUCTION

Eport-E10 development kit is provided to help customer rapidly get used to how to develop the product. The following figure is to show its appearance. Customers are able to use RS232 UART interface or USB-invert-TTL interface for parameter configure, product management and function Test .etc.

Development Kit List:

- Eport-E10 Product: 1 Pcs
- Eport EVB : 1 Pcs
- Cable : 1 Pcs
- USB: 1 Pcs

			
1pcs Eport EVB	1~5pcs Eport-E10	1pcs USB线	1pcs 网线

## 2. HARDWARE REQUIREMENTS

- Eport series super Ethernet 1 Pcs

## 3. SOFTWARE REQUIREMENTS

- TCP&UDP Test Tool
- Serial Tool

Visit Hi-flying official website to download related softwares.

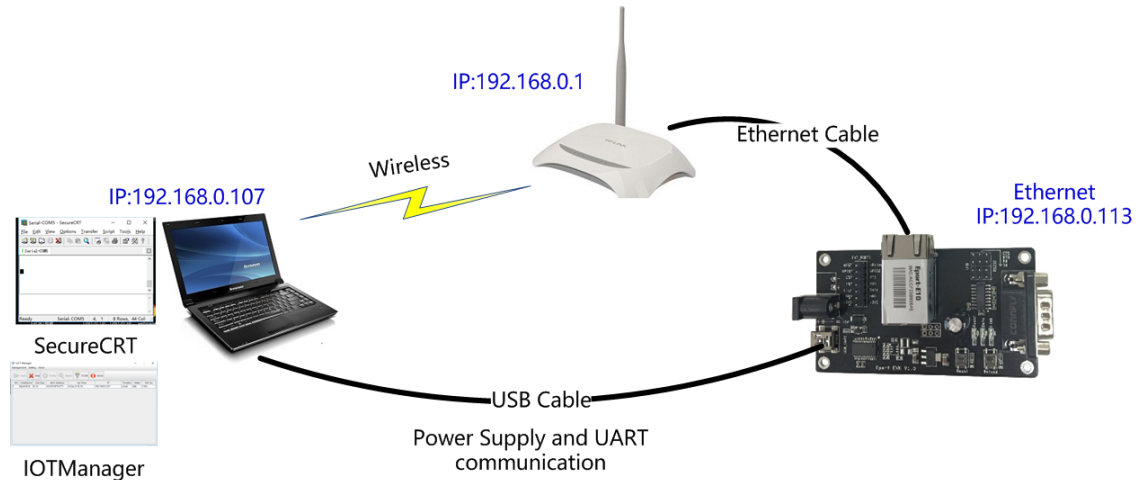
[http://www.hi-flying.com/download\\_list\\_dc/&downloadcategoryid=14&isMode=false&comp\\_stats=comp-FrontDownloadsCategory\\_show01-1376450727769.html](http://www.hi-flying.com/download_list_dc/&downloadcategoryid=14&isMode=false&comp_stats=comp-FrontDownloadsCategory_show01-1376450727769.html)

[http://gb.hi-flying.com/download\\_list\\_dc/&downloadcategoryid=14&isMode=false&comp\\_stats=comp-FrontDownloadsCategory\\_show01-1376450727769.html](http://gb.hi-flying.com/download_list_dc/&downloadcategoryid=14&isMode=false&comp_stats=comp-FrontDownloadsCategory_show01-1376450727769.html)

## 4. HARDWARE CONNECTION

### 4.1. Device Connection

Connect devices as following figure.



### 4.2. TCP&UDP Test HTTP Server

- Send HTTP original data

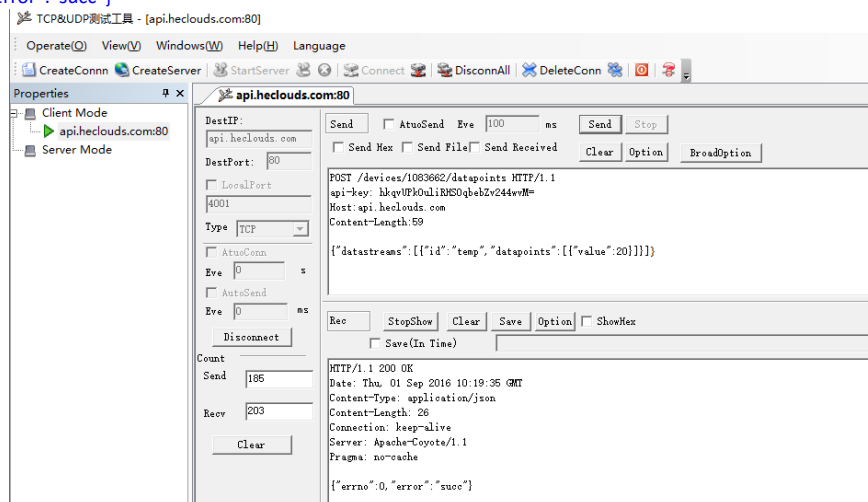
```
POST /devices/1083662/datapoints HTTP/1.1
api-key: hkqvUPkOuliRHS0qbebZv244wM=
Host:api.heclouds.com
Content-Length:59
```

```
{"datastreams":[{"id":"temp","datapoints":[{"value":20}]]}
```

- Server response:

```
HTTP/1.1 200 OK
Date: Thu, 01 Sep 2016 10:23:16 GMT
Content-Type: application/json
Content-Length: 26
Connection: keep-alive
Server: Apache-Coyote/1.1
Pragma: no-cache
```

```
{"errno":0,"error":"succ"}
```



### 4.3. Send HTTP data under Transparent Mode

Step 1: Create TCP client connection

- ◆ Name: Socket name
- ◆ Local Port: Please set to zero when worded as TCP client. It is local and random port.
- ◆ Protocol: Tcp Client
- ◆ Server: Server address, such as api.heclouds.com
- ◆ Server Port: Server port, such as 80

**Basic Settings**

<b>Name</b>	<input type="text" value="Tcp_Client"/>
<b>Local Port</b>	<input type="text" value="0"/>
<b>Buffer Size</b>	<input type="text" value="512"/>
<b>Keep Alive(s)</b>	<input type="text" value="60"/>
<b>Timeout(s)</b>	<input type="text" value="300"/>

**Protocol Settings**

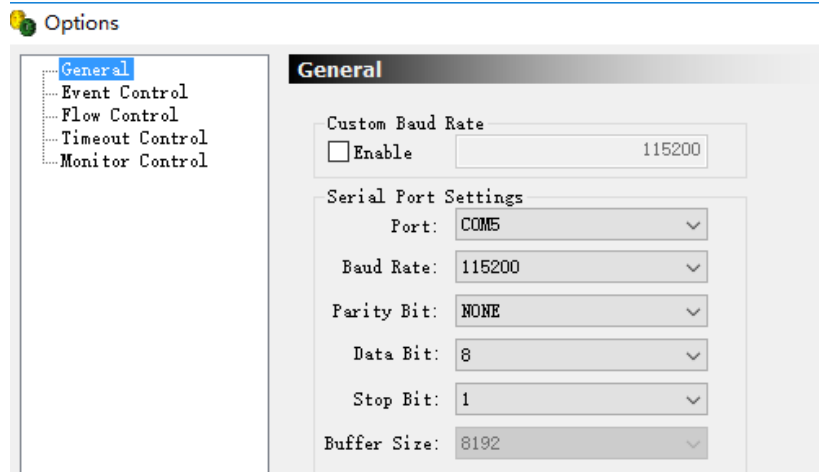
<b>Protocol</b>	<input type="text" value="Tcp Client"/>
<b>Server</b>	<input type="text" value="api.heclouds.com"/>
<b>Server Port</b>	<input type="text" value="80"/>

Step 2: After configuration, confirm TCP client connection status.

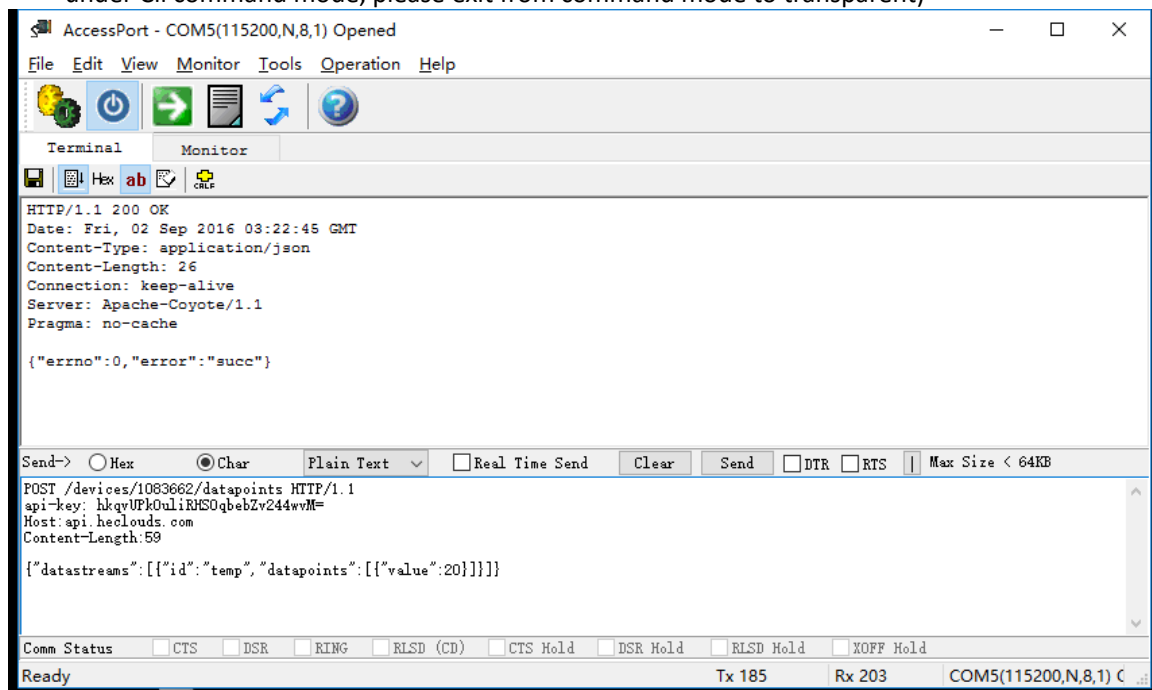
**Communication State - 'Tcp\_Client'**

Received Bytes 0	Received Frames 0
Sent Bytes 0	Sent Frames 0
Failed Bytes 0	Failed Frames 0
Protocol TCP-CLIENT	State Connected
Client Ip 183.230.40.33	

Step 3: Open serial port to set work parameter.



Step 4: Send transparent data (make sure product works under transparent mode. If work under Cli command mode, please exit from command mode to transparent)



#### 4.4. HTTP Send data

Step 1: Create HTTP client connection

- ◆ Name: Socket name
- ◆ Local Port: Please set to zero when worded in HTTP aplication. It is local and random port.
- ◆ Protocol: HTTP
- ◆ Server: Server address, such as api.heclouds.com
- ◆ Server Port: Server port, such as 80
- ◆ Method: According to the original data of HTTP request, type the related data, normally POST and GET.
- ◆ Path: HTTP request path
- ◆ Headers: HTTP header

**Basic Settings**

Name	Tcp_Client
Local Port	0
Buffer Size	512
Keep Alive(s)	60
Timeout(s)	300

**Protocol Settings**

Protocol	Http
Server	api.heclouds.com
Server Port	80
Connect Mode	Always
Method	POST
Version	HTTP/1.1
Path	/devices/1083662/datapoints

**Headers** +

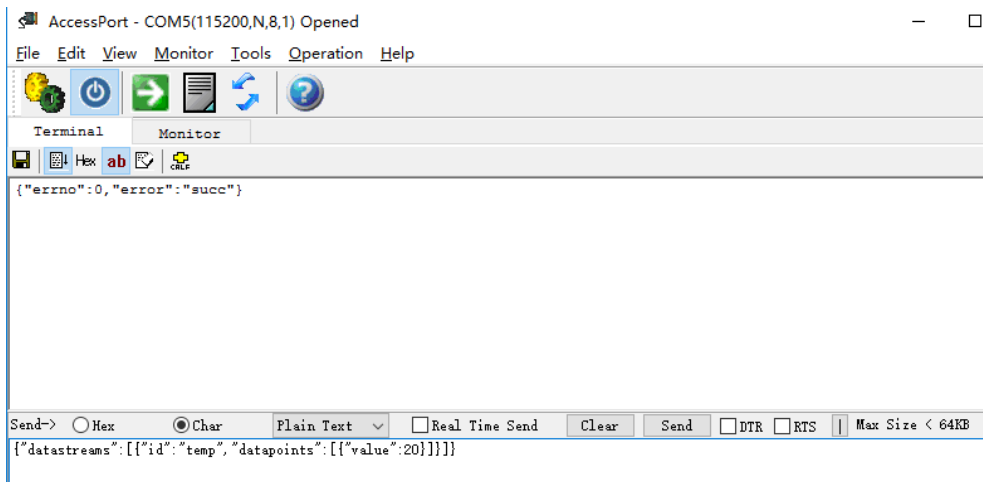
api-key	hkqvUPkOulIRHS0qbebZv244wvM=	-
Host	api.heclouds.com	-
Content-Length	59	-

Step 2: After configuration, confirm TCP client connection status.

**Communication State - 'Tcp\_Client'**

Received Bytes 0	Received Frames 0
Sent Bytes 0	Sent Frames 0
Failed Bytes 0	Failed Frames 0
Protocol TCP-CLIENT	State Connected
Client Ip 183.230.40.33	

Step 3: Serial port sends HTTP data. When device receives the response from server, it will output related HTTP data.

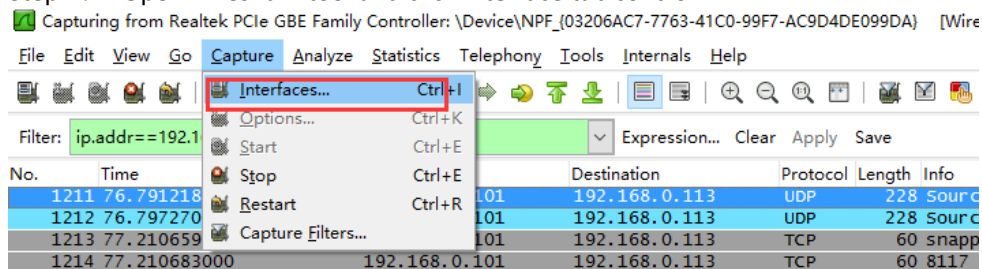


## 5. WIRESHARK PACKAGE CAPTURING SOFTWARE

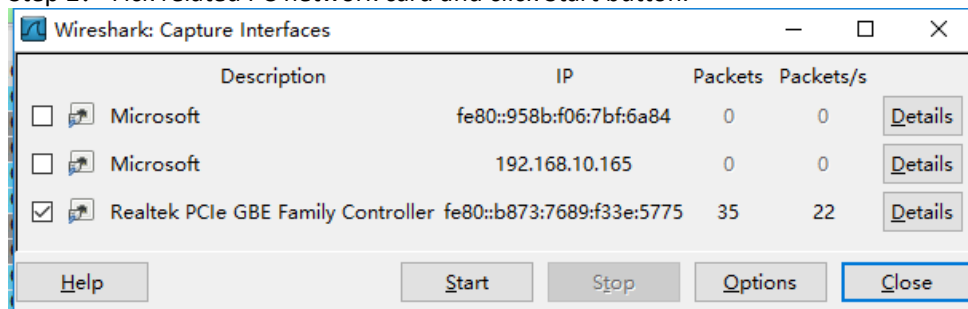
### 5.1. Tool Introduction

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.

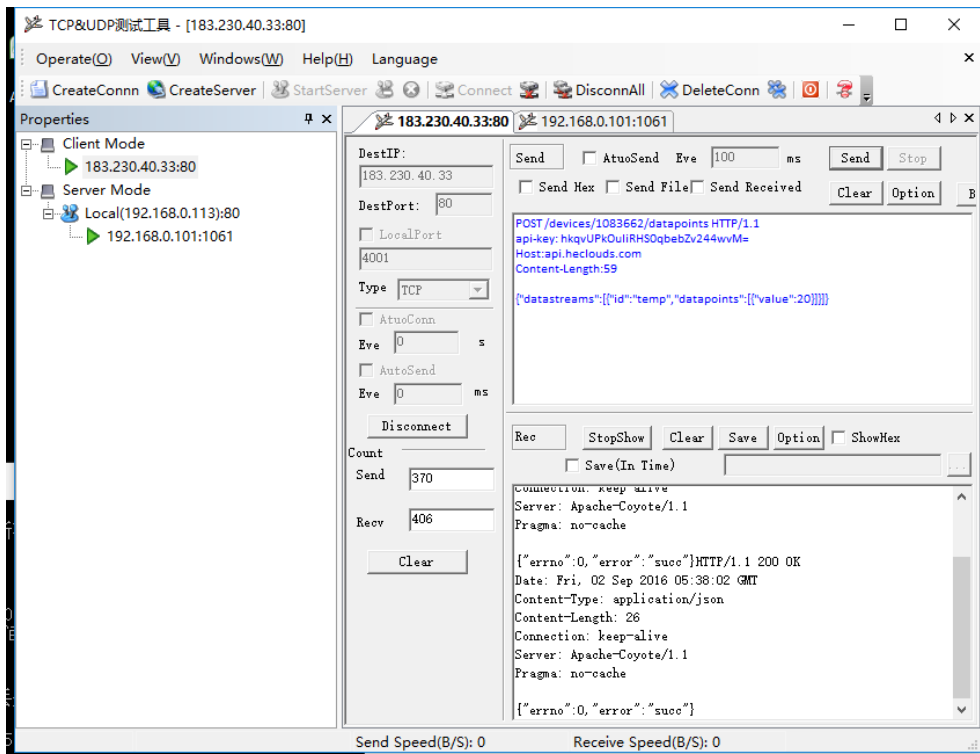


Step 2: Tick related PC network card and click Start button.



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\_{03206AC7-7763-41CD-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=66304 Len=0
129	28.929734000	192.168.0.113	183.230.40.33	HTTP	239	POST /devices/1083662/datapoints HTTP/1.1
130	28.980916000	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	54.351809000	192.168.0.113	183.230.40.33	TCP	54	56482 > http [FIN, ACK] seq=186 Ack=204 win=66304 Len=0
236	54.604429000	183.230.40.33	192.168.0.113	TCP	60	http > 56482 [FIN, ACK] seq=204 Ack=187 win=15744 Len=0
237	54.604520000	192.168.0.113	183.230.40.33	TCP	54	56482 > http [ACK] seq=186 Ack=204 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
257	61.415433000	192.168.0.113	183.230.40.33	TCP	54	56490 > http [ACK] seq=1 Ack=1 win=66304 Len=0
274	67.178769000	192.168.0.113	183.230.40.33	HTTP	239	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	67.293938000	183.230.40.33	192.168.0.113	TCP	60	56490 > http [FIN, ACK] seq=186 Ack=204 win=66304 Len=0
279	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=204 win=66304 Len=0
581	132.237577000	192.168.0.113	183.230.40.33	TCP	54	56490 > http [FIN, ACK] seq=186 Ack=204 win=66304 Len=0
582	132.290350000	183.230.40.33	192.168.0.113	TCP	60	http > 56490 [ACK] seq=205 Ack=187 win=15744 Len=0

## APPENDIX: CONTACT INFORMATION

---

**Address:** Room 1002, Building 1, No.3000, Longdong Avenue, Pudong New Area, Shanghai, China

**Postcode:** 201203

**Web Site:** <http://www.iotworkshop.com/> or [www.hi-flying.com](http://www.hi-flying.com)

**Business Contact:** [business@iotworkshop.com](mailto:business@iotworkshop.com)

**Technical Contact:** [support@iotworkshop.com](mailto:support@iotworkshop.com)

**After Sale Contact:** [service@iotworkshop.com](mailto:service@iotworkshop.com)

---

More information about product, please visit the webpage: [www.iotworkshop.com](http://www.iotworkshop.com)