QLog Linux&Android User Guide

UMTS/HSPA(+)/LTE/5G Module Series

Rev. QLog_Linux&Android_User_Guide_V1.1

Date: 2019-07-25

Status: Released

www.quectel.com

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai, China 200233
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2015-06-02	Arno WANG	Initial
1.1	2019-07-25	Carl YIN	1. Added applicable modules. Please refer to Table 1 for details. 2. Supported catching MDM dump. 3. Supported catching log via tty2tcp function.

Contents

Figure Index

1 Introduction

This document mainly introduces how to use the QLog tool to catch log data from Quectel modules in Linux and Android systems.

1.1. Applicable Modules

Table 1: Applicable Modules

Module Series		Models
UMTS/HSPA(+)		● UCxx: UC15/ UC20/ UC200T
LTE	LTE Standard	● EC2x: EC21/ EC25/ EC20 R2.0/ EC20 R2.1 ● EG9x: EG91/ EG95 ● EM05 ● EG25-G
	Automotive	● AGxx: AG35/ AG15/ AG520R/ AG550Q
	LPWA	● BGxx: BG96/ BG95/ BG77 ● BCxx: BC69/ BC600L-M3
	LTE-A	● Ex06: EG06/ EP06/ EM06 ● Ex12: EG12/ EM12 ● EG18 ● EM20
5G		● Rx500Q: RG500Q/ RM550Q ● Rx510Q: RG510Q/ RM510Q

2 Tool Package

QLog tool package includes source codes and filter configuration files.

The files in the tool package are shown as below:

- *main.c asr.c mdm.c sahara.c tty2tcp.c*
- *conf/*.cfg*

conf/.cfg* are filter configuration files.

NOTES

1. Before running the QLog tool, please make sure the USB virtual ports have readable and writable permissions.
2. UC200T does not need filter configuration files.

3 Introduction on Ports

Before using QLog tool in Linux & Android systems, please ensure that USB driver of the module has been installed successfully in host system. After the module has been connected to the host via USB cable, the corresponding USB virtual ports will be displayed, which are listed as below.

- ttyUSB0------DM port
- ttyUSB1------NEMA port
- ttyUSB2------AT port
- ttyUSB3------Modem port

NOTES

1. QLog tool catches log data through the DM port.
2. The descriptors of USB virtual ports listed above are on the condition that host is not connected to other USB virtual port devices, i.e., host is only connected to Quectel module.

The following command can be put in Linux console to query whether the USB virtual ports exists.

```
ls –al /dev/ttyUSB*
```

4 Command Arguments of QLog Tool

The operating arguments in the command lines of QLog program can be specified, and the detailed arguments are illustrated as below.

Table 2: Description of Command Arguments

Number	Argument	Optional/ Non-optional	Description
1	-p <port>	Optional	Port. <port> can be specified as /dev/ttyUSB*, and the default port is /dev/ttyUSB0.
2	-m <size>	Optional	The max size of a single file to be saved. Unit: MB. Default: 128. Range: 2~512.
3	-n <number>	Optional	The max number of log files to be saved. Default: 0. Range: 0~512. “0” means no limit, and other values mean QLog will automatically delete the oldest log file when the number exceeds max number.
4	-s <path>	Optional	The path to save the log data file. If this argument is set as 9000, QLog will automatically work in TCP server mode, and TCP port is 9000. Then Windows tools like QPST [1)], QWinLog [2)] and CatStudio [1)] can be connected to this TCP port to catch log. When modules enter MDM dump state, QLog will automatically catch MDM dump.
5	-f <filename>	Optional	The name of the filter configuration file. If this argument is not set, default configuration embedded in QLog will be used. (UC200T does not need configuration file.)

NOTES

1. [1)] QPST(Qualcomm Product Support Tool) and CatStudio tools can be used without licensing if needed.
2. [2)] QWinLog is Quectel's tool and can be provided for customers if needed.

5 How to Use QLog Tool

This chapter mainly introduces the procedure of using QLog tool in Linux and Android systems.

QLog tool can catch log data from the module in real time and save it into any directory of customer's device. The log data can be used to analyze the abnormality of the module.

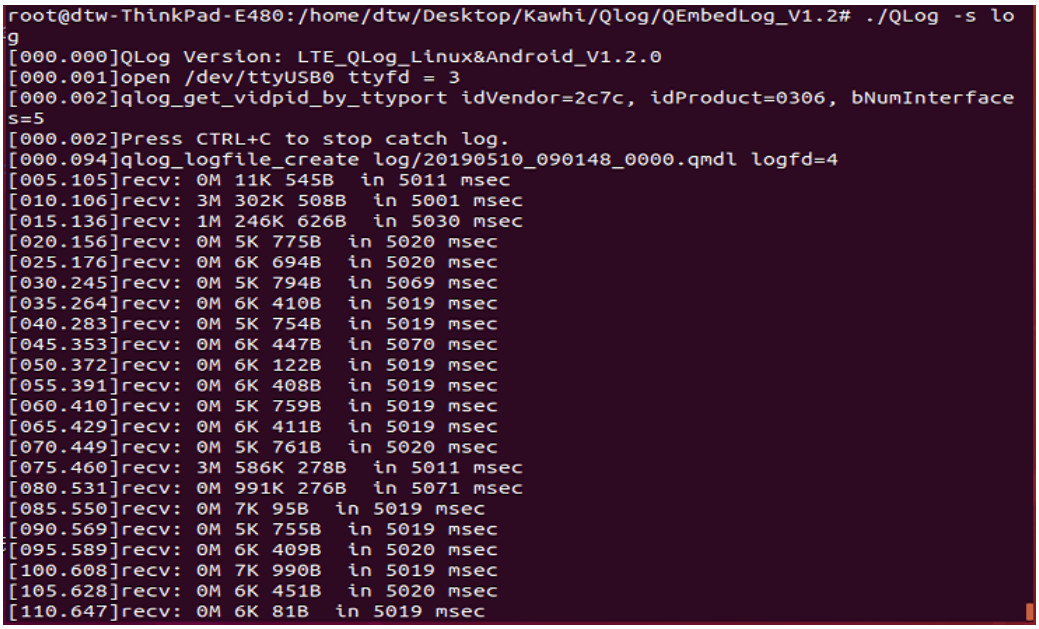
5.1. Linux

In Linux system, the source codes of the tool need to be compiled with the following command:

```
make CROSS_COMPILE=<your platform's cross compiler>
```

After compilation, put the QLog tool and the filter configuration files into a directory of customer's device. Then run the tool to catch log or dump according to their requirements.

5.1.1. Catch Log to Local Disk



```
root@dtw-ThinkPad-E480:/home/dtw/Desktop/Kawhi/Qlog/QEmbedLog_V1.2# ./QLog -s lo
g
[000.000]QLog Version: LTE_QLog_Linux&Android_V1.2.0
[000.001]open /dev/ttyUSB0 ttyfd = 3
[000.002]qlog_get_vidpid_by_ttyport idVendor=2c7c, idProduct=0306, bNumInterface
s=5
[000.002]Press CTRL+C to stop catch log.
[000.094]qlog_logfile_create log/20190510_090148_0000.qmdl logfd=4
[005.105]recv: 0M 11K 545B  in 5011 msec
[010.106]recv: 3M 302K 508B  in 5001 msec
[015.136]recv: 1M 246K 626B  in 5030 msec
[020.156]recv: 0M 5K 775B  in 5020 msec
[025.176]recv: 0M 6K 694B  in 5020 msec
[030.245]recv: 0M 5K 794B  in 5069 msec
[035.264]recv: 0M 6K 410B  in 5019 msec
[040.283]recv: 0M 5K 754B  in 5019 msec
[045.353]recv: 0M 6K 447B  in 5070 msec
[050.372]recv: 0M 6K 122B  in 5019 msec
[055.391]recv: 0M 6K 408B  in 5019 msec
[060.410]recv: 0M 5K 759B  in 5019 msec
[065.429]recv: 0M 6K 411B  in 5019 msec
[070.449]recv: 0M 5K 761B  in 5020 msec
[075.460]recv: 3M 586K 278B  in 5011 msec
[080.531]recv: 0M 991K 276B  in 5071 msec
[085.550]recv: 0M 7K 95B  in 5019 msec
[090.569]recv: 0M 5K 755B  in 5019 msec
[095.589]recv: 0M 6K 409B  in 5020 msec
[100.608]recv: 0M 7K 990B  in 5019 msec
[105.628]recv: 0M 6K 451B  in 5020 msec
[110.647]recv: 0M 6K 81B  in 5019 msec
```

Figure 1: Specify Arguments to Catch Log to Local Disk

5.1.2. Catch Dump to Local Disk

```
root@dtw-ThinkPad-E480:/home/dtw/Desktop/Kawhi/Qlog/QEmbedLog_V1.2# ./QLog -s du
mp
[000.000]QLog Version: LTE_QLog_Linux&Android_V1.2.0
[000.000]open /dev/ttyUSB0 ttyfd = 3
[000.000]qlog_get_vidpid_by_ttyport idVendor=2c7c, idProduct=0125, bNumInterface
s=1
[000.000]Press CTRL+C to stop catch log.
[000.000]STATE <-- SAHARA_WAIT_HELLO
[005.006]select returned error: Success
[005.006]Read 8 bytes, command 1 and packet length 48 bytes
[005.006]RECEIVED <--  SAHARA_HELLO_ID
[005.006]RECEIVED <-- SAHARA_MODE_MEMORY_DEBUG
[005.006]SENDING --> SAHARA_HELLO_RESPONSE
[005.006]STATE <-- SAHARA_WAIT_COMMAND
[005.007]Read 8 bytes, command 9 and packet length 16 bytes
[005.007]RECEIVED <-- SAHARA_MEMORY_DEBUG_ID
[005.007]RECEIVED <-- SAHARA_MEMORY_DEBUG
[005.007]Memory Table Address: 0x87C91B20, Memory Table Length: 0x000001A0
[005.007]SENDING -->  SAHARA_MEMORY_READ, address 0x87C91B20, length 0x000001A0
[005.007]STATE <-- SAHARA_WAIT_MEMORY_TABLE
[005.007]STATE <-- SAHARA_WAIT_MEMORY_TABLE
[005.007]Memory Debug table received
[005.007]Base 0x08600000 Len 0x00004000, 'OCIMEM.BIN', 'OCIMEM'
[005.007]Base 0x00200000 Len 0x00020000, 'CODERAM.BIN', 'RPM Code RAM region'
[005.007]Base 0x00290000 Len 0x00010000, 'DATARAM.BIN', 'RPM Data RAM region'
[005.007]Base 0x00060000 Len 0x00005000, 'MSGRAM.BIN', 'RPM MSG RAM region'
[005.007]Base 0x87C00338 Len 0x00000008, 'PMIC_PON.BIN', 'Pmic PON stat'
[005.007]Base 0x87C00330 Len 0x00000004, 'RST_STAT.BIN', 'Reset Status Region'
[005.007]Base 0x80000000 Len 0x10000000, 'DDRCS0.BIN', ' DDR CS0 Memory'
[005.007]Base 0x87C91320 Len 0x0000025C, 'load.cmm', 'CMM Script'
```

Figure 2: Specify Arguments to Catch Dump to Local Disk

5.1.3. Catch Log to PC via tty2tcp with QLog and QWinLog

```
[142.208]recv: 0M 341K 986B  in 5008 msec
[147.319]recv: 0M 366K 591B  in 5111 msec
[152.481]recv: 0M 370K 682B  in 5162 msec
[157.589]recv: 0M 352K 589B  in 5108 msec
[162.651]recv: 0M 373K 656B  in 5062 msec
[167.713]recv: 0M 374K 771B  in 5062 msec
[172.409]ttyfd recv -1 Bytes. break
root@dtw-ThinkPad-E480:/home/dtw/Desktop/Kawhi/Qlog/QEmbedLog_V1.2# ./QLog -s 90
00
[000.000]QLog Version: LTE_QLog_Linux&Android_V1.2.0
[000.001]open /dev/ttyUSB0 ttyfd = 3
[000.002]qlog_get_vidpid_by_ttyport idVendor=2c7c, idProduct=0306, bNumInterface
s=5
[000.002]Press CTRL+C to stop catch log.
[000.002]Starting the TCP server(9000)...
[000.002]bind OK!
[000.002]listen OK!
Waiting the TCP Client...
[010.332]TCP Client 172.18.112.59:32218 connect
[015.120]recv: 4M 3K 343B  in 4783 msec
[019.568]recv: 4M 1K 14B  in 4448 msec
[023.942]recv: 4M 0K 534B  in 4374 msec
[028.358]recv: 4M 0K 731B  in 4416 msec
[032.652]recv: 4M 2K 764B  in 4294 msec
```

Figure 3: Specify Arguments in QLog

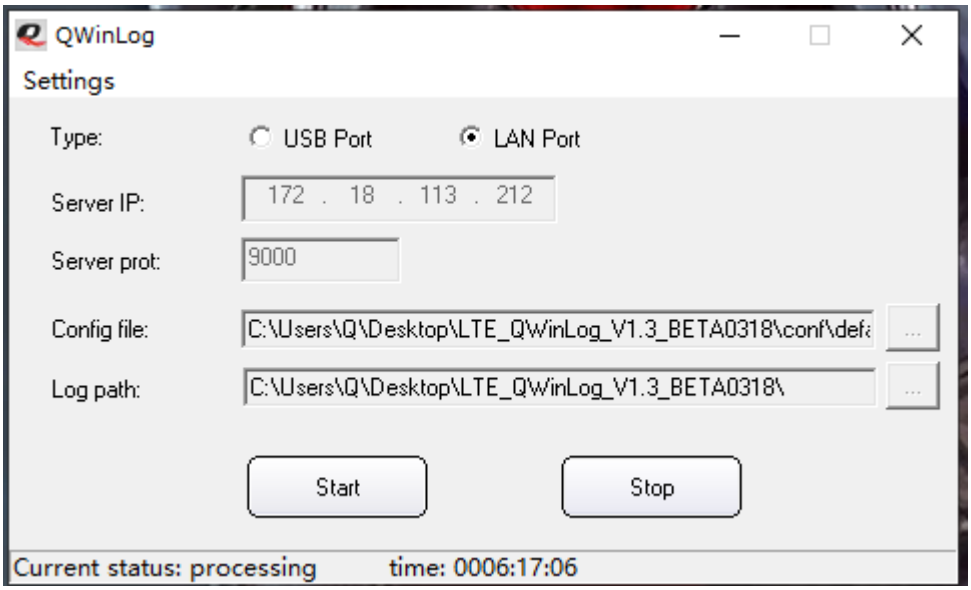
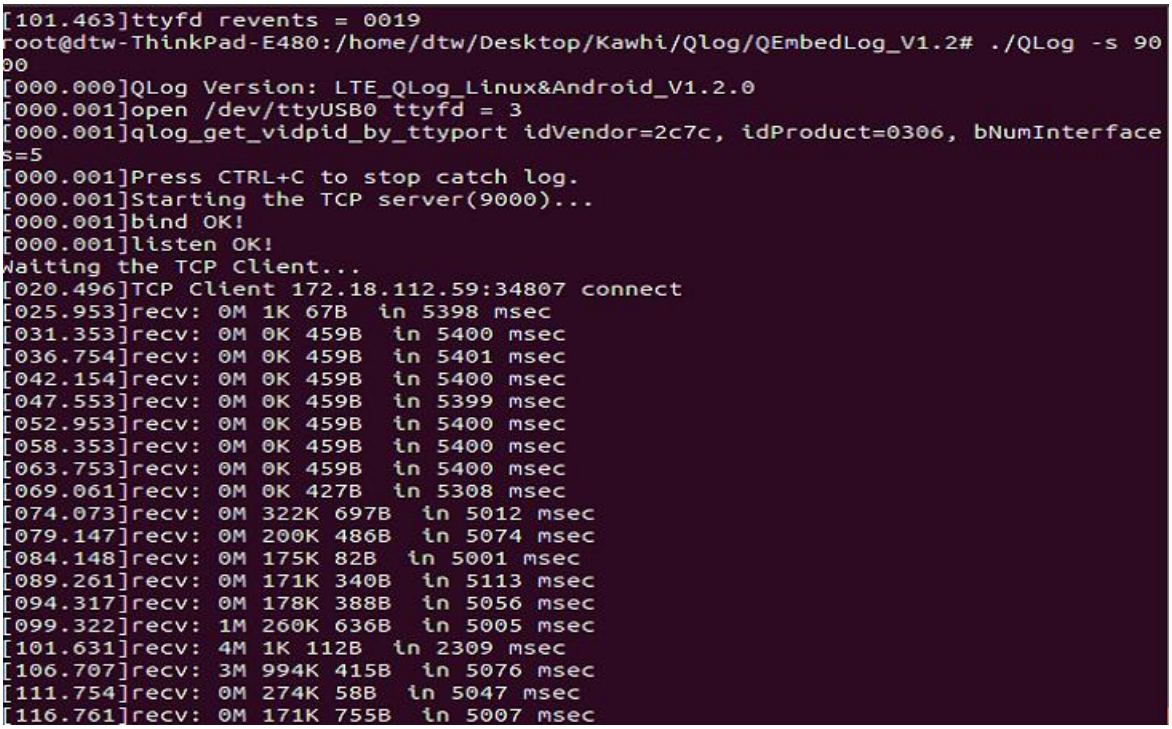


Figure 4: QWinLog Configuration for Catching Log via tty2tcp

5.1.4. Catch Log to PC via tty2tcp with QLog and QXDM



```
[101.463]ttyfd revents = 0019
root@dtw-ThinkPad-E480:/home/dtw/Desktop/Kawhi/Qlog/QEmbedLog_V1.2# ./QLog -s 90
00
[000.000]QLog Version: LTE_QLog_Linux&Android_V1.2.0
[000.001]open /dev/ttyUSB0 ttyfd = 3
[000.001]qlog_get_vidpid_by_ttyport idVendor=2c7c, idProduct=0306, bNumInterface
s=5
[000.001]Press CTRL+C to stop catch log.
[000.001]Starting the TCP server(9000)...
[000.001]bind OK!
[000.001]listen OK!
Waiting the TCP Client...
[020.496]TCP Client 172.18.112.59:34807 connect
[025.953]recv: 0M 1K 67B  in 5398 msec
[031.353]recv: 0M 0K 459B  in 5400 msec
[036.754]recv: 0M 0K 459B  in 5401 msec
[042.154]recv: 0M 0K 459B  in 5400 msec
[047.553]recv: 0M 0K 459B  in 5399 msec
[052.953]recv: 0M 0K 459B  in 5400 msec
[058.353]recv: 0M 0K 459B  in 5400 msec
[063.753]recv: 0M 0K 459B  in 5400 msec
[069.061]recv: 0M 0K 427B  in 5308 msec
[074.073]recv: 0M 322K 697B  in 5012 msec
[079.147]recv: 0M 200K 486B  in 5074 msec
[084.148]recv: 0M 175K 82B  in 5001 msec
[089.261]recv: 0M 171K 340B  in 5113 msec
[094.317]recv: 0M 178K 388B  in 5056 msec
[099.322]recv: 1M 260K 636B  in 5005 msec
[101.631]recv: 4M 1K 112B  in 2309 msec
[106.707]recv: 3M 994K 415B  in 5076 msec
[111.754]recv: 0M 274K 58B  in 5047 msec
[116.761]recv: 0M 171K 755B  in 5007 msec
```

Figure 5: Specify Arguments in QLog

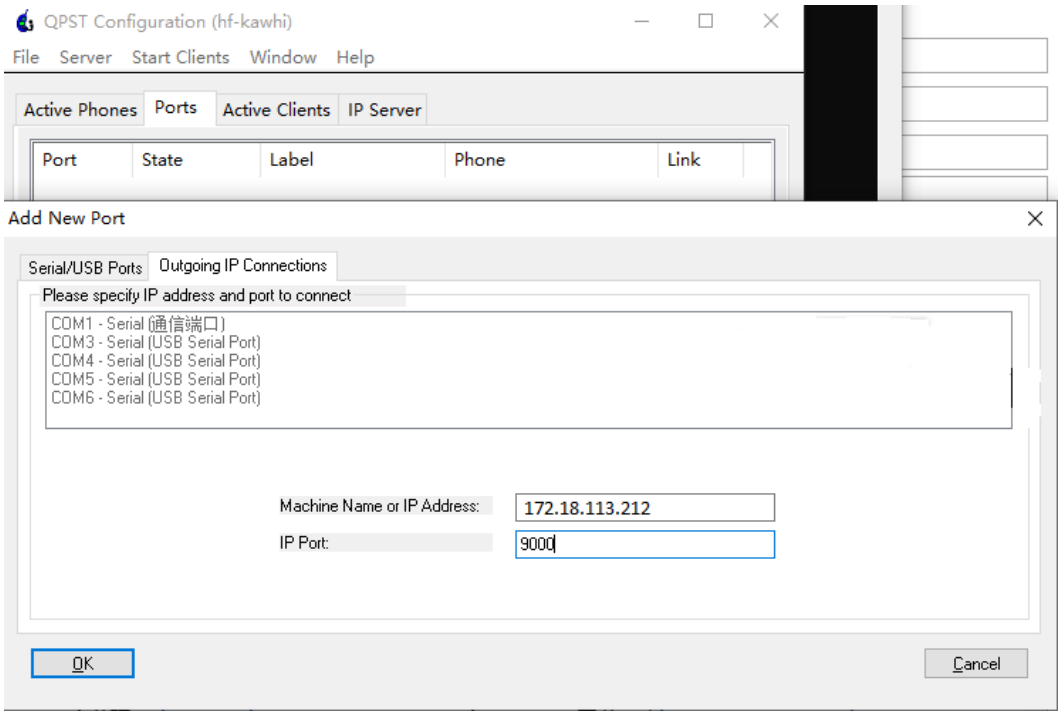


Figure 6: IP Address and Port Configuration of QPST

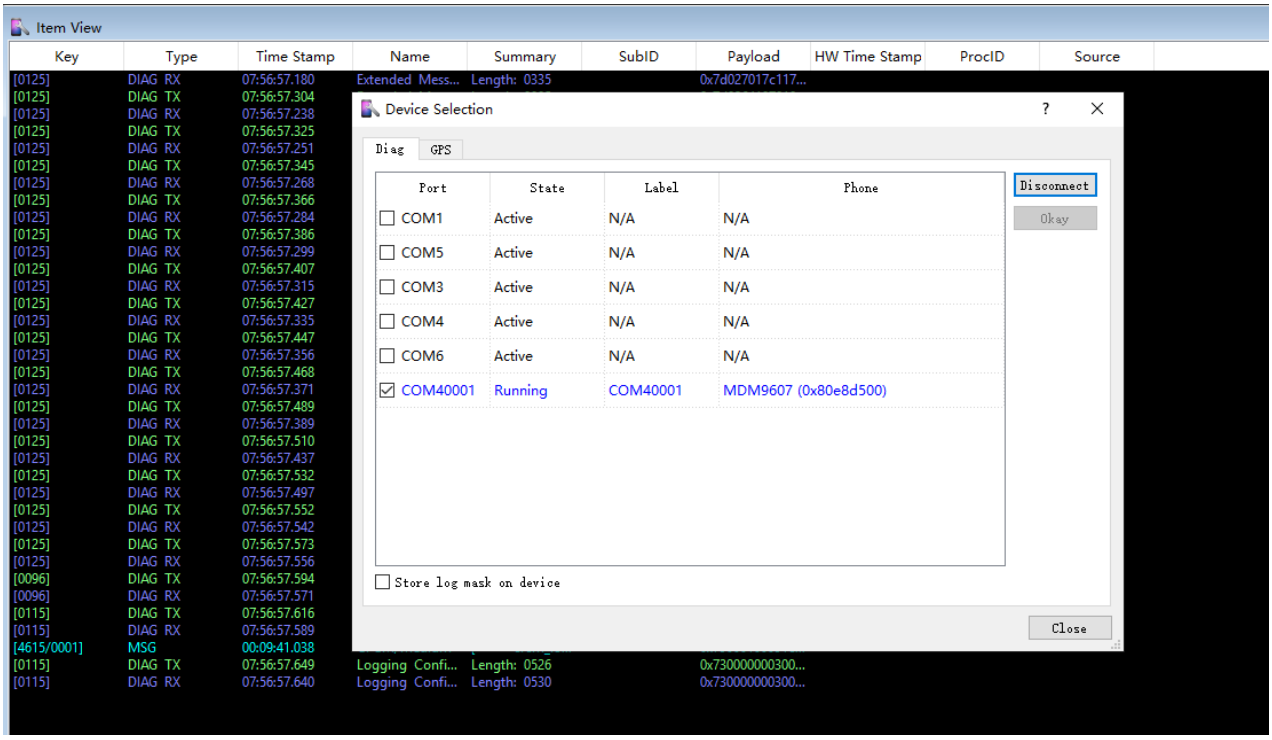


Figure 7: Device Selection for Catching Log via tty2tcp with QXDM

NOTE

QXDM(Qualcomm Extensible Diagnostic Monitor) is Qualcomm's tool and needs to be licensed before use.

5.2. Android

In Android system, please use the pre-built binary QLog tool, which is under directory *android*.

5.2.1. Catch Log to Local Disk

```
255|nanopc-t4:/data/Qlog_test # ./QAndroidLog -s log
[000.000]QLog Version: LTE_QLog_Linux&Android_V1.2.1
[000.003]open /dev/ttyUSB0 ttyfd = 3
[000.003]qlog_get_vidpid_by_ttyport idVendor=2c7c, idProduct=0435, bNumInterfaces=7
[000.003]Press CTRL+C to stop catch log.
[000.281]qlog_logfile_create log/20190509_074750_0000.qmdl logfd=4
[005.418]recv: 0M 460K 516B  in 5137 msec
[010.576]recv: 0M 389K 691B  in 5158 msec
[015.683]recv: 0M 437K 646B  in 5107 msec
[020.791]recv: 0M 426K 928B  in 5108 msec
[025.982]recv: 0M 494K 770B  in 5191 msec
```

Figure 8: Specify Arguments to Catch Log to Local Disk

5.2.2. Catch Dump to Local Disk

```
nanopc-t4:/data/Qlog_test # ./QAndroidLog -s dump
[000.000]QLog Version: LTE_QLog_Linux&Android_V1.2.1
[000.000]open /dev/ttyUSB0 ttyfd = 3
[000.001]qlog_get_vidpid_by_ttyport idVendor=2c7c, idProduct=0125, bNumInterfaces=1
[000.001]Press CTRL+C to stop catch log.
[000.001]STATE <-- SAHARA_WAIT_HELLO
[002.482]Read 8 bytes, command 1 and packet length 48 bytes
[002.482]RECEIVED <--  SAHARA_HELLO_ID
[002.482]RECEIVED <-- SAHARA_MODE_MEMORY_DEBUG
[002.482]SENDING --> SAHARA_HELLO_RESPONSE
[002.482]STATE <-- SAHARA_WAIT_COMMAND
[004.491]Read 8 bytes, command 4 and packet length 16 bytes
[004.491]RECEIVED <-- SAHARA_END_IMAGE_TX_ID
[004.491]Received an unknown command: 4
[004.491]SENDING --> SAHARA_RESET
[004.491]STATE <-- SAHARA_WAIT_RESET_RESP
[004.491]STATE <-- SAHARA_WAIT_RESET_RESP
[004.491]Read 8 bytes, command 8 and packet length 8 bytes
[004.492]RECEIVED <-- SAHARA_RESET_RESP_ID
[004.495]Read/Write File descriptor returned error: Success, error code 0
[004.495]Get reset response code 8
[004.495]Sahara protocol completed
[004.495]Catch DUMP using Sahara protocol successful


nanopc-t4:/data/Qlog_test #
```

Figure 9: Specify Arguments to Catch Dump to Local Disk

5.2.3. Catch Log to PC via tty2tcp with QLog and QWinLog

```
nanopc-t4:/data/Qlog_test # ./QAndroidLog -s 9000
[000.000]QLog Version: LTE_QLog_Linux&Android_V1.2.1
[000.002]open /dev/ttyUSB0 ttyfd = 3
[000.002]qlog_get_vidpid_by_ttyport idVendor=2c7c, idProduct=0435, bNumInterfaces=7
[000.003]Press CTRL+C to stop catch log.
[000.003]Starting the TCP server(9000)...
[000.003]bind OK!
[000.003]listen OK!
Waiting the TCP Client...
```

Figure 10: Specify Arguments in QLog

Specify the ports in adb(Android Debug Bridge) tool, and the ports must be the same as the ones specified in QLog.

```
C:\Users\Q>adb forward tcp:9000 tcp:9000
```

Figure 11: Specify Ports in adb Tool

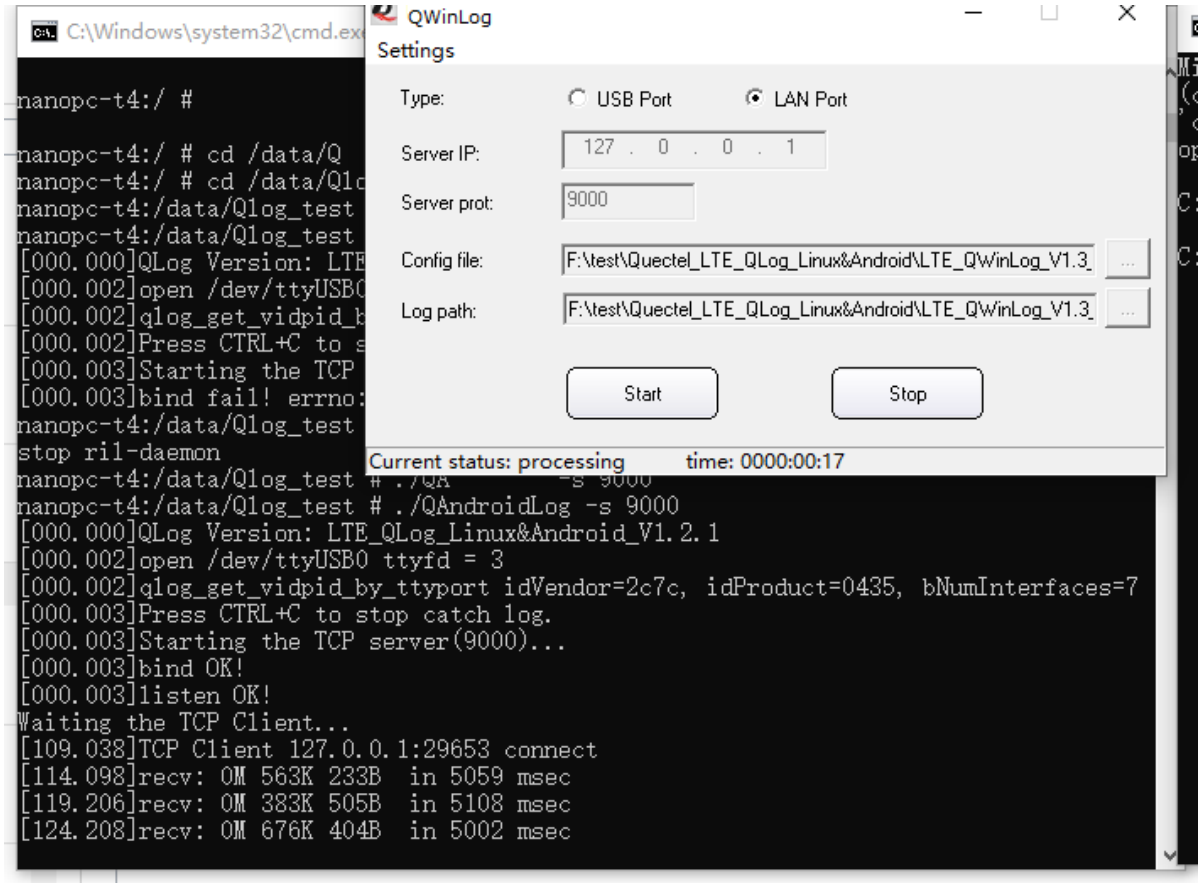


Figure 12: QWinLog Configuration for Catching Log via tty2tcp

NOTE

Please note that the server IP address must be configured as “127.0.0.1”.

5.2.4. Catch Log to PC via tty2tcp with QLog and QXDM

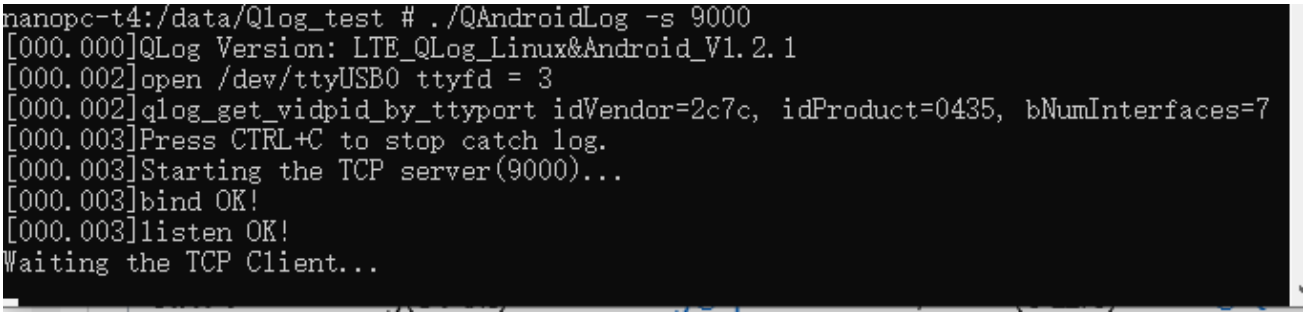


Figure 13: Specify Arguments in QLog

Specify the ports in adb(Android Debug Bridge) tool, and the ports must be the same as the ones specified in QLog.

```
C:\Users\Q>adb forward tcp:9000 tcp:9000
```

Figure 14: Specify Ports in adb Tool

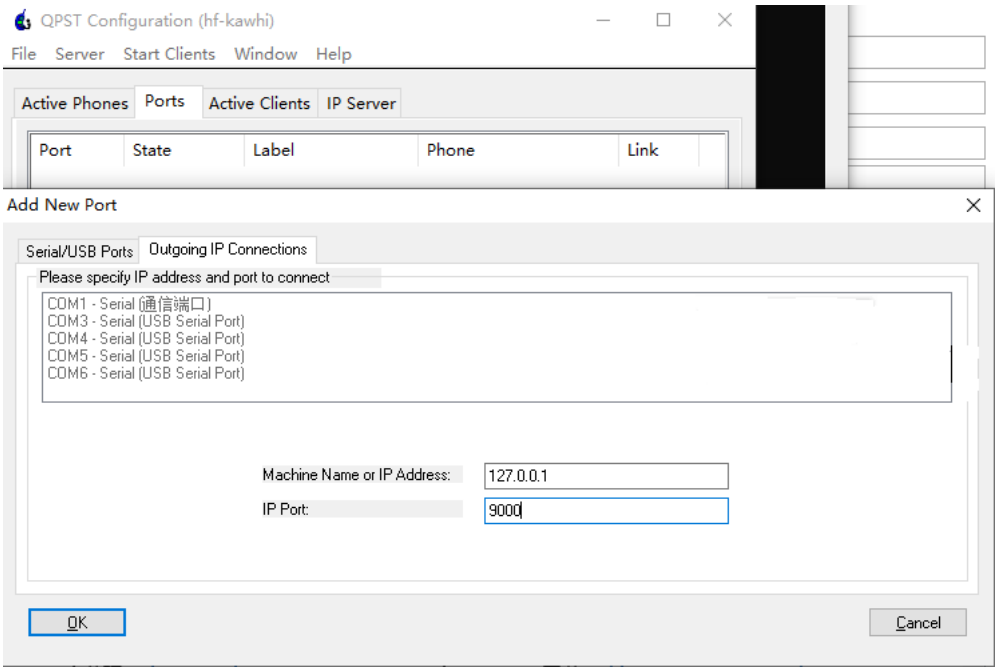


Figure 15: IP Address and Port Configuration of QPST

NOTE

Please note that the IP address must be configured as “127.0.0.1”.

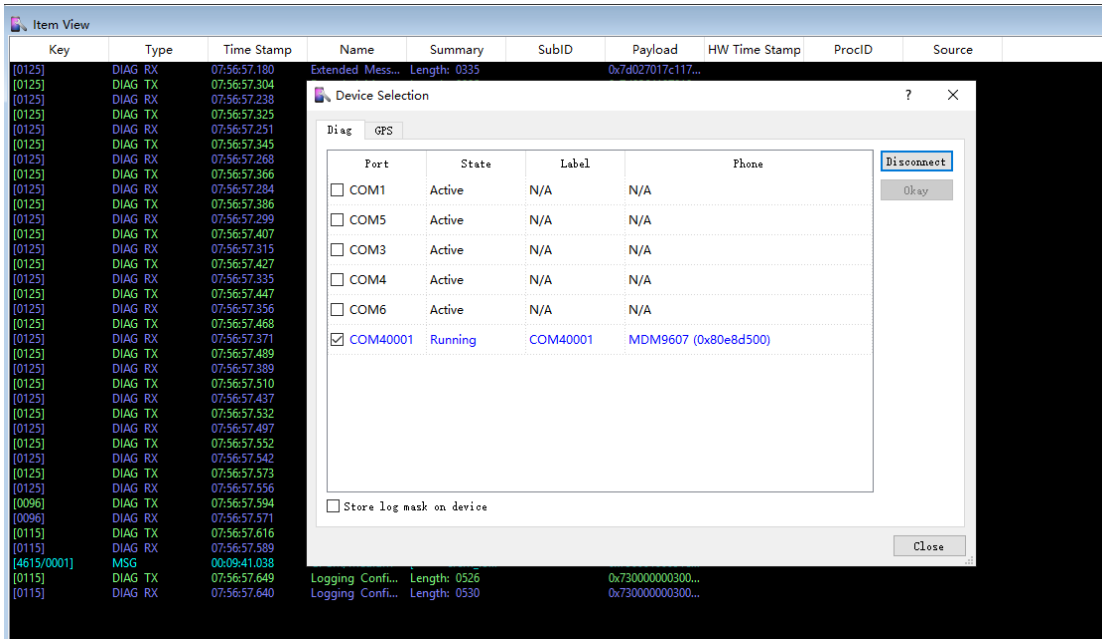


Figure 16: Device Selection for Catching Log via tty2tcp with QXDM

NOTE

QXDM(Qualcomm Extensible Diagnostic Monitor) tool is Qualcomm's tool and needs to be licensed before use.