STM32 Function Test Guide

Version: 1.0

Date: 2024-3-27

Status: Preliminary

At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

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

Or our local offices. 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 us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

About the Document

Revision History

Version	Date	Author	Description
1.0	2024-03-27	Fiona Fang	Creation of the Document
1.9	2024-05-22	Mandy Wang	Modified Some Test Logs

Table Index

Table 1: Device Definition 8
Table 2: SD Function Definition 17
Table 3: TCP&UDP Function Definition 18
Table 4: FTP(S) Function Definition 18
Table 5: HTTP(S) Function Definition 20
Table 6: PSM Function Definition 21
Table 7: Certificates for the Test 23
Table 8: Related Documents 24

Contents

About the Document **3**

Table Index **4**

Contents **5**

Figure Index **6**

1 Introduction **7**

2 Test Environment **7**

2.1. Preparation 7
2.2. Connection 8
2.3. Port 9
2.4. Download 9
2.5. Function Test 11
2.6. Example 12
2.6.1. TCP-client 12
2.6.2. FTP Uploading 13
2.6.3. HTTPS-POST 14

3 Functions **17**

3.1. Save Log in SD Card 17
3.2. TCP&UDP 17
3.3. FTP(S) 18
3.4. HTTP(S) 20
3.5. PSM 21
3.6. MQTT 22
3.7. Function Help 22

4 Test Files **23**

5 Appendix A References **24**

Figure Index

Figure 1: Reference Circuit of the Devices 7
Figure 2: Reference Circuit of the Top View 8
Figure 3: Reference Circuit of the Bottom View 9
Figure 4: Reference Circuit of Port 9
Figure 5: Memory & File Editing 10
Figure 6: Successful Download 10
Figure 7: Reset Location 11
Figure 8: Succeed to Enable Port 11
Figure 9: Do your own business 12
Figure 10: TCP Command 12
Figure 11: Successful TCP 13
Figure 12: FTP Command 13
Figure 13: Successful Uploading 14
Figure 14: Location of the uploaded file 14
Figure 15: Test Files 14
Figure 16: HTTPS-POST Command 15
Figure 17: POST Successfully 15
Figure 18: Path of the Posted File 15
Figure 19: Post File Successfully 16
Figure 20: Reference Circuit of the Help 23

1 Introduction

The article mainly illustrates how to test basic function (TCPUDP/FTP(s)/HTTP(s)/PSM/MQTT(s)) in the module based on the STM32 MB1136 EVB, Quectel FAE carrier board, and Wireless Cellular Module TE-A board.

2 Test Environment

The devices of test environment must be required.

2.1. Preparation

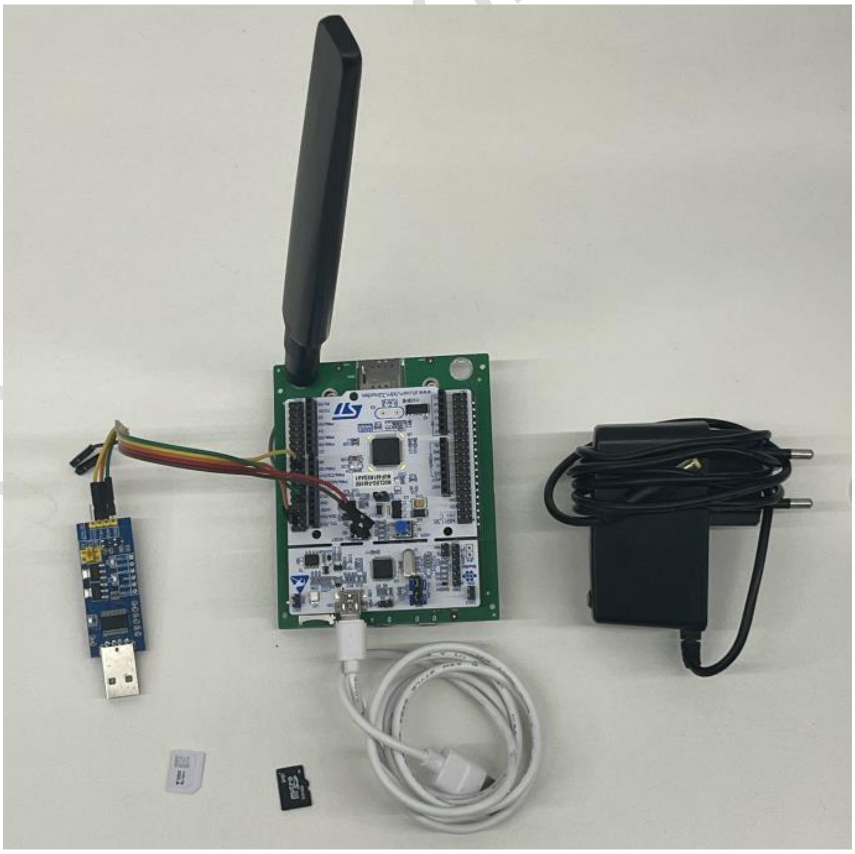


Figure 1: Reference Circuit of the Devices

Table 1: Device Definition

Device	Description
STM32 MB1135	STM32 EVB board
Carrier board	Quectel FAE Carrier board
TE-A	Wireless Cellular Module TE-A board
Antenna	Cellular antenna
SIM card	Select the sim card that corresponding to the module
SD card	Store the certificate and test files
Power	5V power supply
STlink	Download the firmware
TTL	USB TO TTL(3.3V)

2.2. Connection

Please connect Power, SIM card and Antenna.

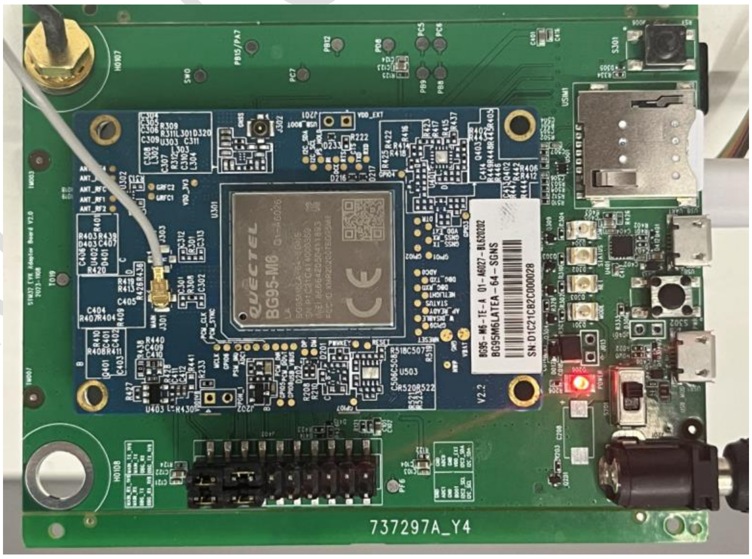


Figure 2: Reference Circuit of the Top View

Please connect STlink, SD card, and the TTL(TX,RX,GND).

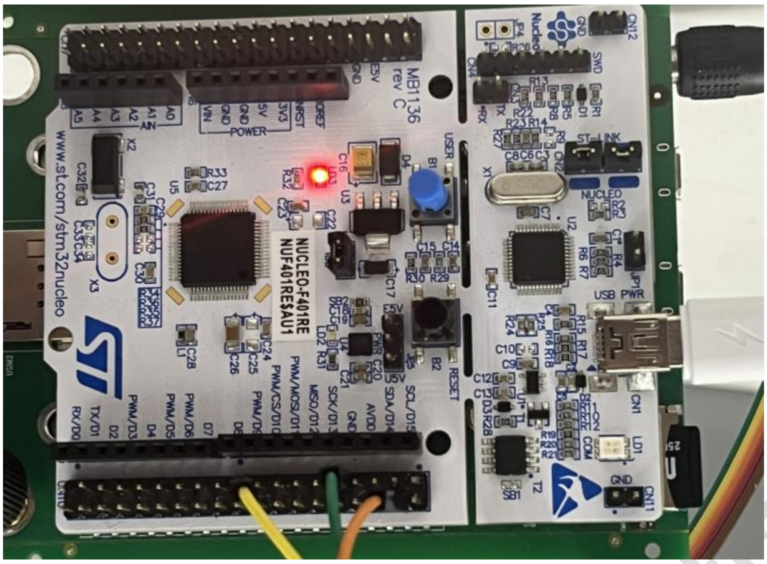
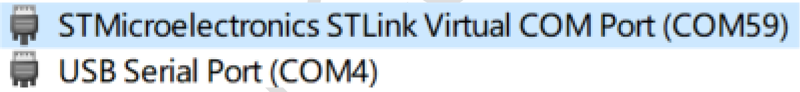


Figure 3: Reference Circuit of the Bottom View

2.3. Port

The STLink Virtual COM Port is used to download the firmware.
The USB serial Port is used to connect Xshell and test the function.



STMicroelectronics STLink Virtual COM Port (COM59)

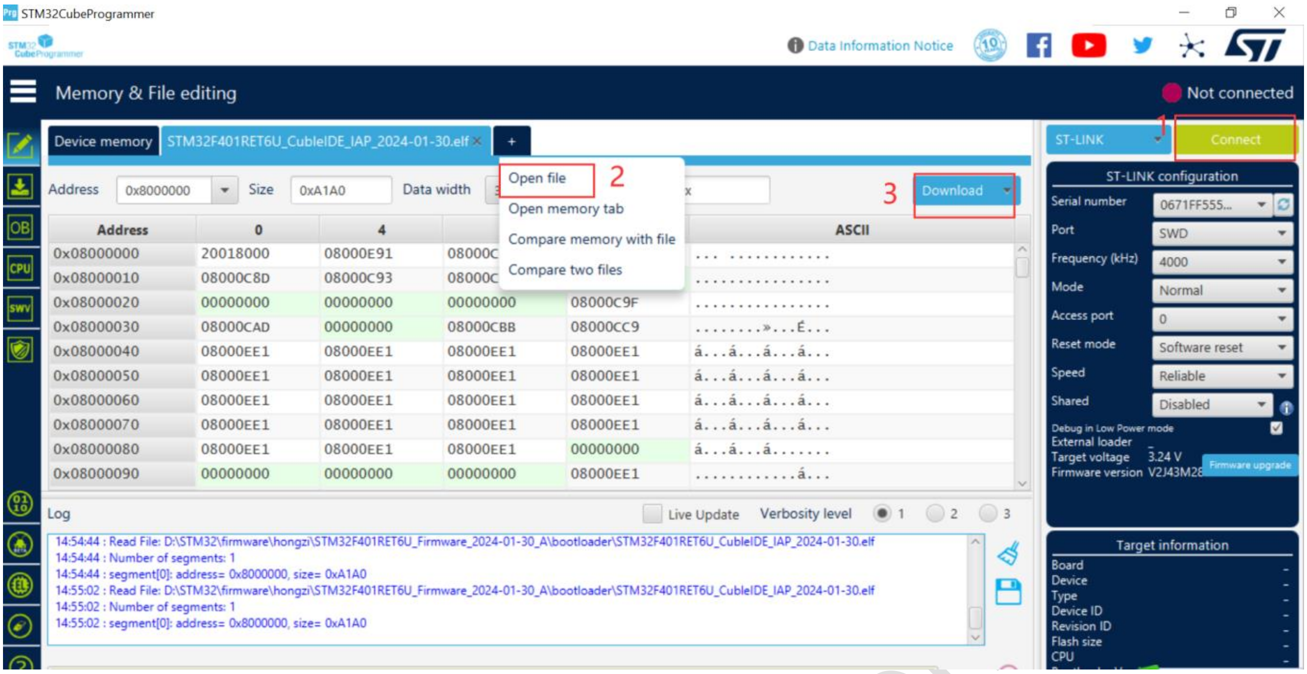
USB Serial Port (COM4)

Figure 4: Reference Circuit of Port

2.4. Download

Open the STM32CubeProgrammer.

1. Click “**Connect**” button to connect the STM32 EVB board.
2. Open file, select the firmware “STM32F401RET6U_CubleIDE_IAP_2024-03-01.elf”, which is the Bootloader firmware.
3. Click “**Download**” button to download the Bootloader firmware. Note: there is no need to update Bootloader firmware each time. If the bootloader has been updated before, do not upgrade it later.
4. Open file, select the firmware “STM32F401RET6U_CubleIDE_FreeRTOS_2024-03-01.elf”, which is the Application firmware.
5. Click “**Download**” button to download the Application firmware.



Device memory | STM32F401RET6U_CubleIDE_IAP_2024-01-30.elf

Address 0x8000000 Size 0xA1A0 Data width

```
Address      0         4         8         C         ASCII
0x08000000   20018000  08000E91  08000C              ... ...........
0x08000010   08000C8D  08000C93  08000C              ...............
0x08000020   00000000  00000000  00000000  08000C9F  ...............
0x08000030   08000CAD  00000000  08000CBB  08000CC9  ........»...É...
0x08000040   08000EE1  08000EE1  08000EE1  08000EE1  á...á...á...á...
0x08000050   08000EE1  08000EE1  08000EE1  08000EE1  á...á...á...á...
0x08000060   08000EE1  08000EE1  08000EE1  08000EE1  á...á...á...á...
0x08000070   08000EE1  08000EE1  08000EE1  08000EE1  á...á...á...á...
0x08000080   08000EE1  08000EE1  08000EE1  00000000  á...á...á.......
0x08000090   00000000  00000000  00000000  08000EE1  ............á...
```

Log □ Live Update Verbosity level ● 1 ○ 2 ○ 3

```
14:54:44 : Read File: D:\STM32\firmware\hongzi\STM32F401RET6U_Firmware_2024-01-30_A\bootloader\STM32F401RET6U_CubleIDE_IAP_2024-01-30.elf
14:54:44 : Number of segments: 1
14:54:44 : segment[0]: address= 0x8000000, size= 0xA1A0
14:55:02 : Read File: D:\STM32\firmware\hongzi\STM32F401RET6U_Firmware_2024-01-30_A\bootloader\STM32F401RET6U_CubleIDE_IAP_2024-01-30.elf
14:55:02 : Number of segments: 1
14:55:02 : segment[0]: address= 0x8000000, size= 0xA1A0
```

ST-LINK

ST-LINK configuration

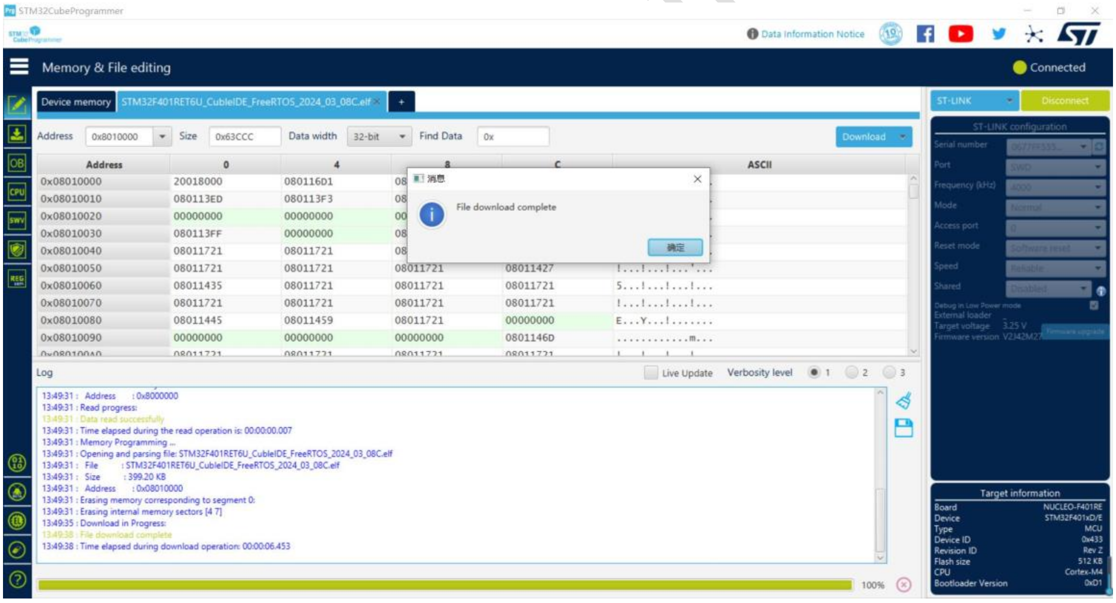
Serial number	0671FF555...
Port	SWD
Frequency (kHz)	4000
Mode	Normal
Access port	0
Reset mode	Software reset
Speed	Reliable
Shared	Disabled
Debug in Low Power mode	☑
External loader	
Target voltage	3.24 V
Firmware version	V2J43M28

Firmware upgrade

Target information

Board	-
Device	-
Type	-
Device ID	-
Revision ID	-
Flash size	-
CPU	-

Figure 5: Memory & File Editing



Device memory | STM32F401RET6U_CubleIDE_FreeRTOS_2024_03_08C.elf

Address 0x8010000 Size 0x63CCC Data width 32-bit Find Data 0x

```
Address      0         4         8         C         ASCII
0x08010000   20018000  080116D1  08
0x08010010   080113ED  080113F3  08
0x08010020   00000000  00000000  00
0x08010030   080113FF  00000000  08
0x08010040   08011721  08011721  08
0x08010050   08011721  08011721  08011721  08011427  !...!...!...'...
0x08010060   08011435  08011721  08011721  08011721  5...!...!...!...
0x08010070   08011721  08011721  08011721  08011721  !...!...!...!...
0x08010080   08011445  08011459  08011721  00000000  E...Y...!.......
0x08010090   00000000  00000000  00000000  0801146D  ............m...
```

File download complete

Log □ Live Update Verbosity level ● 1 ○ 2 ○ 3

```
13:49:31 :   Address      : 0x8000000
13:49:31 : Read progress:
13:49:31 : Data read successfully
13:49:31 : Time elapsed during the read operation is: 00:00:00.007
13:49:31 : Memory Programming ...
13:49:31 : Opening and parsing file: STM32F401RET6U_CubleIDE_FreeRTOS_2024_03_08C.elf
13:49:31 :   File         : STM32F401RET6U_CubleIDE_FreeRTOS_2024_03_08C.elf
13:49:31 :   Size         : 399.20 KB
13:49:31 :   Address      : 0x08010000
13:49:31 : Erasing memory corresponding to segment 0:
13:49:31 : Erasing internal memory sectors [4 7]
13:49:35 : Download in Progress:
13:49:38 : File download complete
13:49:38 : Time elapsed during download operation: 00:00:06.453
```

100%

ST-LINK

ST-LINK configuration

Serial number	0677FF555...
Port	SWD
Frequency (kHz)	4000
Mode	Normal
Access port	0
Reset mode	Software reset
Speed	Reliable
Shared	Disabled
Debug in Low Power mode	☑
External loader	
Target voltage	3.25 V
Firmware version	V2J42M27

Target information

Board	NUCLEO-F401RE
Device	STM32F401xD/E
Type	MCU
Device ID	0x433
Revision ID	Rev Z
Flash size	512 KB
CPU	Cortex-M4
Bootloader Version	0xD1

Figure 6: Successful Download

2.5. Function Test

Open the Serial communication tool Xshell, connect with the Serial port and reset the device.

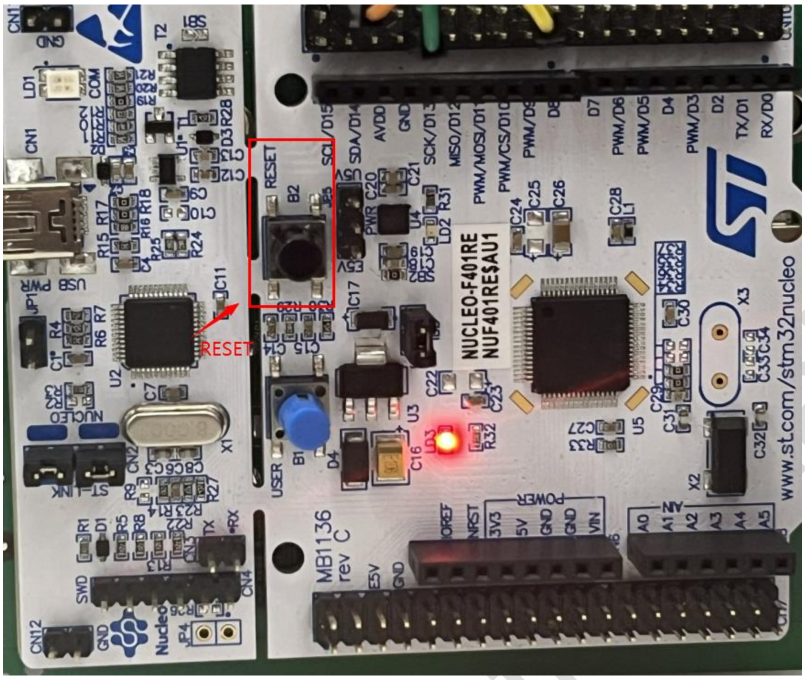
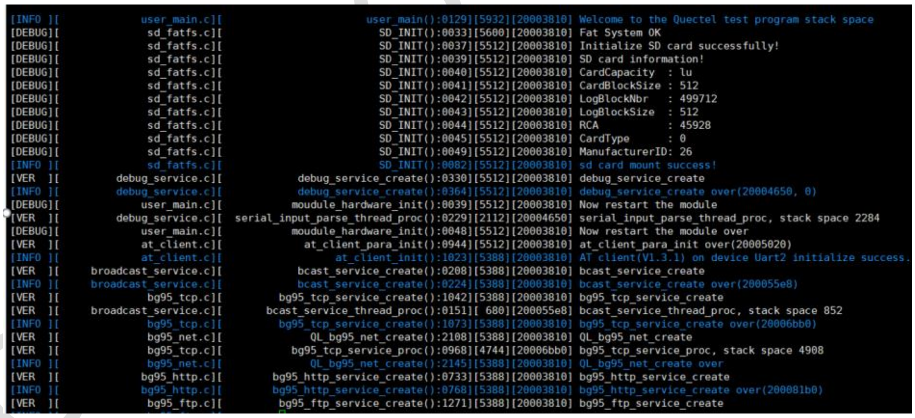


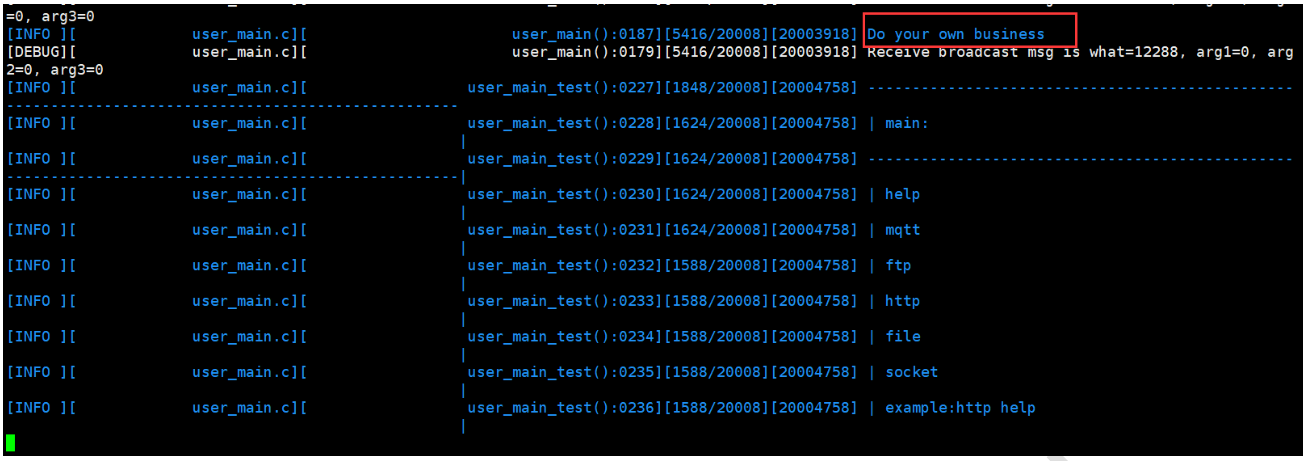
Figure 7: Reset Location



```
[INFO ][           user_main.c][                       user_main():0129][5932][20003810] Welcome to the Quectel test program stack space
[DEBUG][            sd_fatfs.c][                         SD_INIT():0033][5600][20003810] Fat System OK
[DEBUG][            sd_fatfs.c][                         SD_INIT():0037][5512][20003810] Initialize SD card successfully!
[DEBUG][            sd_fatfs.c][                         SD_INIT():0039][5512][20003810] SD card information!
[DEBUG][            sd_fatfs.c][                         SD_INIT():0040][5512][20003810] CardCapacity  : lu
[DEBUG][            sd_fatfs.c][                         SD_INIT():0041][5512][20003810] CardBlockSize : 512
[DEBUG][            sd_fatfs.c][                         SD_INIT():0042][5512][20003810] LogBlockNbr   : 499712
[DEBUG][            sd_fatfs.c][                         SD_INIT():0043][5512][20003810] LogBlockSize  : 512
[DEBUG][            sd_fatfs.c][                         SD_INIT():0044][5512][20003810] RCA           : 45928
[DEBUG][            sd_fatfs.c][                         SD_INIT():0045][5512][20003810] CardType      : 0
[DEBUG][            sd_fatfs.c][                         SD_INIT():0049][5512][20003810] ManufacturerID: 26
[INFO ][            sd_fatfs.c][                         SD_INIT():0082][5512][20003810] sd card mount success!
[VER  ][       debug_service.c][           debug_service_create():0330][5512][20003810] debug_service_create
[INFO ][       debug_service.c][           debug_service_create():0364][5512][20003810] debug_service_create over(20004650, 0)
[DEBUG][           user_main.c][         moudule_hardware_init():0039][5512][20003810] Now restart the module
[VER  ][       debug_service.c][  serial_input_parse_thread_proc():0229][2112][20004650] serial_input_parse_thread_proc, stack space 2284
[DEBUG][           user_main.c][         moudule_hardware_init():0048][5512][20003810] Now restart the module over
[VER  ][           at_client.c][          at_client_para_init():0944][5512][20003810] at_client_para_init over(20005020)
[INFO ][           at_client.c][               at_client_init():1023][5388][20003810] AT client(V1.3.1) on device Uart2 initialize success.
[VER  ][   broadcast_service.c][           bcast_service_create():0208][5388][20003810] bcast_service_create
[INFO ][   broadcast_service.c][           bcast_service_create():0224][5388][20003810] bcast_service_create over(200055e8)
[VER  ][            bg95_tcp.c][        bg95_tcp_service_create():1042][5388][20003810] bg95_tcp_service_create
[VER  ][   broadcast_service.c][      bcast_service_thread_proc():0151][ 680][200055e8] bcast_service_thread_proc, stack space 852
[INFO ][            bg95_tcp.c][        bg95_tcp_service_create():1073][5388][20003810] bg95_tcp_service_create over(20006bb0)
[VER  ][            bg95_net.c][             QL_bg95_net_create():2108][5388][20003810] QL_bg95_net_create
[VER  ][            bg95_tcp.c][          bg95_tcp_service_proc():0968][4744][20006bb0] bg95_tcp_service_proc, stack space 4908
[INFO ][            bg95_net.c][             QL_bg95_net_create():2145][5388][20003810] QL_bg95_net_create over
[VER  ][           bg95_http.c][       bg95_http_service_create():0733][5388][20003810] bg95_http_service_create
[INFO ][           bg95_http.c][       bg95_http_service_create():0768][5388][20003810] bg95_http_service_create over(200081b0)
[VER  ][            bg95_ftp.c][        bg95_ftp_service_create():1271][5388][20003810] bg95_ftp_service_create
```

Figure 8: Succeed to Enable Port

From the Xshell tool, we can check the log is outputted automatically. Module initialization like network -related log will be outputted, and “**do your own business”** message will be displayed till the network registration is done. Then you can execute **main** command test function.



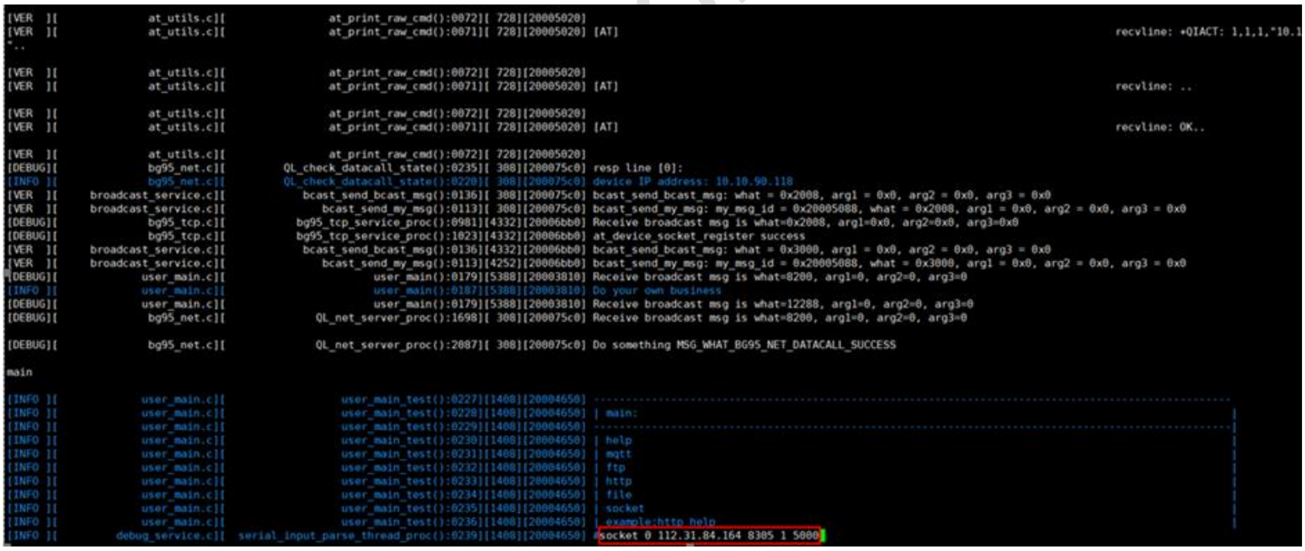
```
=0, arg3=0
[INFO ][            user_main.c][                    user_main():0187][5416/20008][20003918] Do your own business
[DEBUG][            user_main.c][                    user_main():0179][5416/20008][20003918] Receive broadcast msg is what=12288, arg1=0, arg
2=0, arg3=0
[INFO ][            user_main.c][               user_main_test():0227][1848/20008][20004758] ------------------------------------------------
---------------------------------------------------
[INFO ][            user_main.c][               user_main_test():0228][1624/20008][20004758] | main:
                                                   |
[INFO ][            user_main.c][               user_main_test():0229][1624/20008][20004758] ------------------------------------------------
---------------------------------------------------|
[INFO ][            user_main.c][               user_main_test():0230][1624/20008][20004758] | help
                                                   |
[INFO ][            user_main.c][               user_main_test():0231][1624/20008][20004758] | mqtt
                                                   |
[INFO ][            user_main.c][               user_main_test():0232][1588/20008][20004758] | ftp
                                                   |
[INFO ][            user_main.c][               user_main_test():0233][1588/20008][20004758] | http
                                                   |
[INFO ][            user_main.c][               user_main_test():0234][1588/20008][20004758] | file
                                                   |
[INFO ][            user_main.c][               user_main_test():0235][1588/20008][20004758] | socket
                                                   |
[INFO ][            user_main.c][               user_main_test():0236][1588/20008][20004758] | example:http help
                                                   |
```

Figure 9: Do your own business

2.6. Example

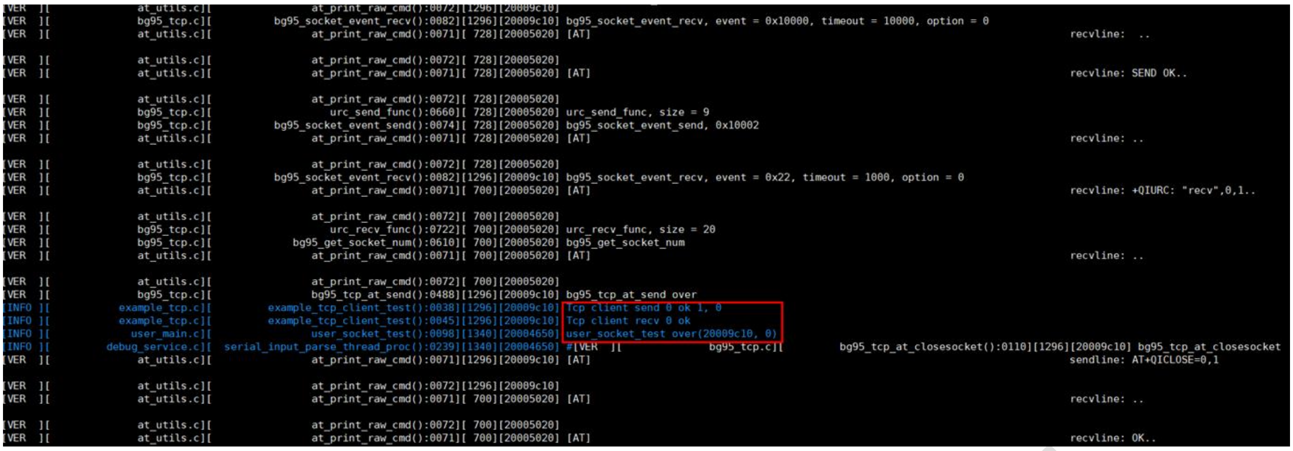
2.6.1. TCP-client

1. After resetting board, it will occur “**do your down business**”
2. Execute the command “**main**”
3. Execute the command “**socket 0 112.31.84.164 8305 1 5000**”



```
[VER  ][          at_utils.c][                at_print_raw_cmd():0072][ 728][20005020]
[VER  ][          at_utils.c][                at_print_raw_cmd():0071][ 728][20005020] [AT]                                        recvline: +QIACT: 1,1,1,"10.1
"..

[VER  ][          at_utils.c][                at_print_raw_cmd():0072][ 728][20005020]
[VER  ][          at_utils.c][                at_print_raw_cmd():0071][ 728][20005020] [AT]                                        recvline: ..

[VER  ][          at_utils.c][                at_print_raw_cmd():0072][ 728][20005020]
[VER  ][          at_utils.c][                at_print_raw_cmd():0071][ 728][20005020] [AT]                                        recvline: OK..

[VER  ][          at_utils.c][                at_print_raw_cmd():0072][ 728][20005020]
[DEBUG][          bg95_net.c][         QL_check_datacall_state():0235][ 308][200075c0] resp line [0]:
[INFO ][          bg95_net.c][         QL_check_datacall_state():0220][ 308][200075c0] device IP address: 10.10.90.118
[VER  ][ broadcast_service.c][           bcast_send_bcast_msg():0136][ 308][200075c0] bcast_send_bcast_msg: what = 0x2008, arg1 = 0x0, arg2 = 0x0, arg3 = 0x0
[VER  ][ broadcast_service.c][              bcast_send_my_msg():0113][ 308][200075c0] bcast_send_my_msg: my_msg_id = 0x20005088, what = 0x2008, arg1 = 0x0, arg2 = 0x0, arg3 = 0x0
[DEBUG][          bg95_tcp.c][          bg95_tcp_service_proc():0981][4332][20006bb0] Receive broadcast msg is what=0x2008, arg1=0x0, arg2=0x0, arg3=0x0
[DEBUG][          bg95_tcp.c][          bg95_tcp_service_proc():1023][4332][20006bb0] at_device_socket_register success
[VER  ][ broadcast_service.c][           bcast_send_bcast_msg():0136][4332][20006bb0] bcast_send_bcast_msg: what = 0x3000, arg1 = 0x0, arg2 = 0x0, arg3 = 0x0
[VER  ][ broadcast_service.c][              bcast_send_my_msg():0113][4252][20006bb0] bcast_send_my_msg: my_msg_id = 0x20005088, what = 0x3000, arg1 = 0x0, arg2 = 0x0, arg3 = 0x0
[DEBUG][         user_main.c][                      user_main():0179][5388][20003810] Receive broadcast msg is what=8200, arg1=0, arg2=0, arg3=0
[INFO ][         user_main.c][                      user_main():0187][5388][20003810] Do your own business
[DEBUG][         user_main.c][                      user_main():0179][5388][20003810] Receive broadcast msg is what=12288, arg1=0, arg2=0, arg3=0
[DEBUG][          bg95_net.c][            QL_net_server_proc():1698][ 308][200075c0] Receive broadcast msg is what=8200, arg1=0, arg2=0, arg3=0

[DEBUG][          bg95_net.c][            QL_net_server_proc():2087][ 308][200075c0] Do something MSG_WHAT_BG95_NET_DATACALL_SUCCESS

main

[INFO ][         user_main.c][                 user_main_test():0227][1408][20004650] ---------------------------------------------------------------------------
[INFO ][         user_main.c][                 user_main_test():0228][1408][20004650] | main:                                                                    |
[INFO ][         user_main.c][                 user_main_test():0229][1408][20004650] ---------------------------------------------------------------------------|
[INFO ][         user_main.c][                 user_main_test():0230][1408][20004650] | help                                                                     |
[INFO ][         user_main.c][                 user_main_test():0231][1408][20004650] | mqtt                                                                     |
[INFO ][         user_main.c][                 user_main_test():0232][1408][20004650] | ftp                                                                      |
[INFO ][         user_main.c][                 user_main_test():0233][1408][20004650] | http                                                                     |
[INFO ][         user_main.c][                 user_main_test():0234][1408][20004650] | file                                                                     |
[INFO ][         user_main.c][                 user_main_test():0235][1408][20004650] | socket                                                                   |
[INFO ][         user_main.c][                 user_main_test():0236][1408][20004650] | example:http help                                                        |
[INFO ][     debug_service.c][  serial_input_parse_thread_proc():0239][1408][20004650] #socket 0 112.31.84.164 8305 1 5000
```

Figure 10: TCP Command

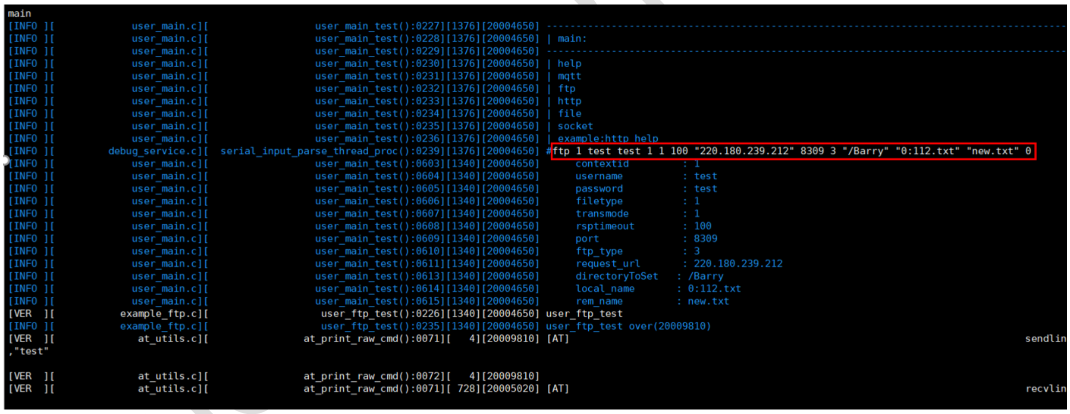


```
[VER  ][          at_utils.c][              at_print_raw_cmd():0072][1296][20009c10]
[VER  ][          bg95_tcp.c][        bg95_socket_event_recv():0082][1296][20009c10] bg95_socket_event_recv, event = 0x10000, timeout = 10000, option = 0
[VER  ][          at_utils.c][              at_print_raw_cmd():0071][ 728][20005020] [AT]                                              recvline:  ..

[VER  ][          at_utils.c][              at_print_raw_cmd():0072][ 728][20005020]
[VER  ][          at_utils.c][              at_print_raw_cmd():0071][ 728][20005020] [AT]                                              recvline: SEND OK..

[VER  ][          at_utils.c][              at_print_raw_cmd():0072][ 728][20005020]
[VER  ][          bg95_tcp.c][                urc_send_func():0660][ 728][20005020] urc_send_func, size = 9
[VER  ][          bg95_tcp.c][        bg95_socket_event_send():0074][ 728][20005020] bg95_socket_event_send, 0x10002
[VER  ][          at_utils.c][              at_print_raw_cmd():0071][ 728][20005020] [AT]                                              recvline: ..

[VER  ][          at_utils.c][              at_print_raw_cmd():0072][ 728][20005020]
[VER  ][          bg95_tcp.c][        bg95_socket_event_recv():0082][1296][20009c10] bg95_socket_event_recv, event = 0x22, timeout = 1000, option = 0
[VER  ][          at_utils.c][              at_print_raw_cmd():0071][ 700][20005020] [AT]                                              recvline: +QIURC: "recv",0,1..

[VER  ][          at_utils.c][              at_print_raw_cmd():0072][ 700][20005020]
[VER  ][          bg95_tcp.c][                urc_recv_func():0722][ 700][20005020] urc_recv_func, size = 20
[VER  ][          bg95_tcp.c][           bg95_get_socket_num():0610][ 700][20005020] bg95_get_socket_num
[VER  ][          at_utils.c][              at_print_raw_cmd():0071][ 700][20005020] [AT]                                              recvline: ..

[VER  ][          at_utils.c][              at_print_raw_cmd():0072][ 700][20005020]
[VER  ][          bg95_tcp.c][              bg95_tcp_at_send():0488][1296][20009c10] bg95_tcp_at_send over
[INFO ][       example_tcp.c][       example_tcp_client_test():0038][1296][20009c10] Tcp client send 0 ok 1, 0
[INFO ][       example_tcp.c][       example_tcp_client_test():0045][1296][20009c10] Tcp client recv 0 ok
[INFO ][         user_main.c][              user_socket_test():0098][1340][20004650] user_socket_test over(20009c10, 0)
[INFO ][     debug_service.c][  serial_input_parse_thread_proc():0239][1340][20004650] #[VER  ][          bg95_tcp.c][        bg95_tcp_at_closesocket():0110][1296][20009c10] bg95_tcp_at_closesocket
[VER  ][          at_utils.c][              at_print_raw_cmd():0071][1296][20009c10] [AT]                                              sendline: AT+QICLOSE=0,1

[VER  ][          at_utils.c][              at_print_raw_cmd():0072][1296][20009c10]
[VER  ][          at_utils.c][              at_print_raw_cmd():0071][ 700][20005020] [AT]                                              recvline: ..

[VER  ][          at_utils.c][              at_print_raw_cmd():0072][ 700][20005020]
[VER  ][          at_utils.c][              at_print_raw_cmd():0071][ 700][20005020] [AT]                                              recvline: OK..
```

Figure 11: Successful TCP

2.6.2. FTP Uploading

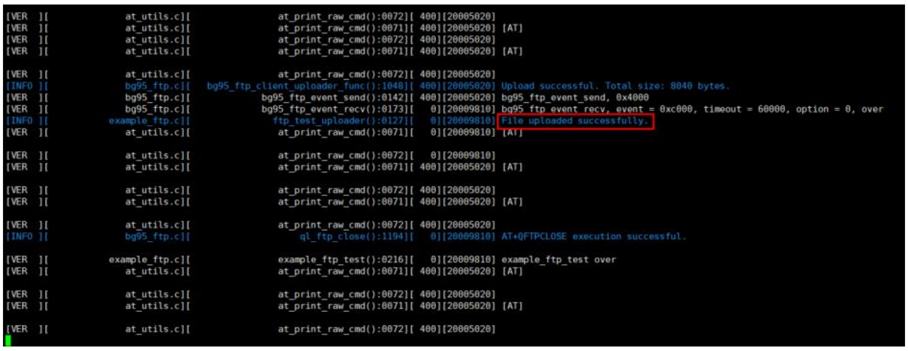
1. After resetting board, it will display “ **do your down business**”
2. Execute the command “**main**”
3. Execute the command “**ftp 1 test test 1 1 100 "220.180.239.212" 8309 3 "/Barry" "0:112.txt" "new.txt" 0**”



```
main
[INFO ][          user_main.c][                 user_main_test():0227][1376][20004650] -----------------------------------------------------------------------------------------
[INFO ][          user_main.c][                 user_main_test():0228][1376][20004650] | main:
[INFO ][          user_main.c][                 user_main_test():0229][1376][20004650] -----------------------------------------------------------------------------------------
[INFO ][          user_main.c][                 user_main_test():0230][1376][20004650] | help
[INFO ][          user_main.c][                 user_main_test():0231][1376][20004650] | mqtt
[INFO ][          user_main.c][                 user_main_test():0232][1376][20004650] | ftp
[INFO ][          user_main.c][                 user_main_test():0233][1376][20004650] | http
[INFO ][          user_main.c][                 user_main_test():0234][1376][20004650] | file
[INFO ][          user_main.c][                 user_main_test():0235][1376][20004650] | socket
[INFO ][          user_main.c][                 user_main_test():0236][1376][20004650] | example:http help
[INFO ][      debug_service.c][  serial_input_parse_thread_proc():0239][1376][20004650] #ftp 1 test test 1 1 100 "220.180.239.212" 8309 3 "/Barry" "0:112.txt" "new.txt" 0
[INFO ][          user_main.c][                 user_main_test():0603][1340][20004650]     contextid          : 1
[INFO ][          user_main.c][                 user_main_test():0604][1340][20004650]     username           : test
[INFO ][          user_main.c][                 user_main_test():0605][1340][20004650]     password           : test
[INFO ][          user_main.c][                 user_main_test():0606][1340][20004650]     filetype           : 1
[INFO ][          user_main.c][                 user_main_test():0607][1340][20004650]     transmode          : 1
[INFO ][          user_main.c][                 user_main_test():0608][1340][20004650]     rsptimeout         : 100
[INFO ][          user_main.c][                 user_main_test():0609][1340][20004650]     port               : 8309
[INFO ][          user_main.c][                 user_main_test():0610][1340][20004650]     ftp_type           : 3
[INFO ][          user_main.c][                 user_main_test():0611][1340][20004650]     request_url        : 220.180.239.212
[INFO ][          user_main.c][                 user_main_test():0613][1340][20004650]     directoryToSet   : /Barry
[INFO ][          user_main.c][                 user_main_test():0614][1340][20004650]     local_name       : 0:112.txt
[INFO ][          user_main.c][                 user_main_test():0615][1340][20004650]     rem_name         : new.txt
[VER  ][        example_ftp.c][                  user_ftp_test():0226][1340][20004650] user_ftp_test
[INFO ][        example_ftp.c][                  user_ftp_test():0235][1340][20004650] user_ftp_test over(20009810)
[VER  ][           at_utils.c][               at_print_raw_cmd():0071][   4][20009810] [AT]                                            sendlin
,"test"

[VER  ][           at_utils.c][               at_print_raw_cmd():0072][   4][20009810]
[VER  ][           at_utils.c][               at_print_raw_cmd():0071][ 728][20005020] [AT]                                            recvlin
```

Figure 12: FTP Command

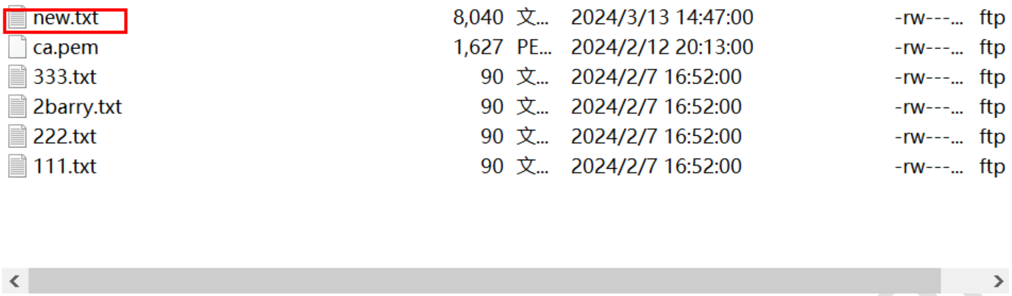
The file is uploaded successfully.



```
[VER  ][           at_utils.c][                at_print_raw_cmd():0072][ 400][20005020]
[VER  ][           at_utils.c][                at_print_raw_cmd():0071][ 400][20005020] [AT]
[VER  ][           at_utils.c][                at_print_raw_cmd():0072][ 400][20005020]
[VER  ][           at_utils.c][                at_print_raw_cmd():0071][ 400][20005020] [AT]

[VER  ][           at_utils.c][                at_print_raw_cmd():0072][ 400][20005020]
[INFO ][           bg95_ftp.c][   bg95_ftp_client_uploader_func():1048][ 400][20005020] Upload successful. Total size: 8040 bytes.
[VER  ][           bg95_ftp.c][             bg95_ftp_event_send():0142][ 400][20005020] bg95_ftp_event_send, 0x4000
[VER  ][           bg95_ftp.c][             bg95_ftp_event_recv():0173][   0][20009810] bg95_ftp_event_recv, event = 0xc000, timeout = 60000, option = 0, over
[INFO ][        example_ftp.c][              ftp_test_uploader():0127][   0][20009810] File uploaded successfully.
[VER  ][           at_utils.c][                at_print_raw_cmd():0071][   0][20009810] [AT]

[VER  ][           at_utils.c][                at_print_raw_cmd():0072][   0][20009810]
[VER  ][           at_utils.c][                at_print_raw_cmd():0071][ 400][20005020] [AT]

[VER  ][           at_utils.c][                at_print_raw_cmd():0072][ 400][20005020]
[VER  ][           at_utils.c][                at_print_raw_cmd():0071][ 400][20005020] [AT]

[VER  ][           at_utils.c][                at_print_raw_cmd():0072][ 400][20005020]
[INFO ][           bg95_ftp.c][                   ql_ftp_close():1194][   0][20009810] AT+QFTPCLOSE execution successful.

[VER  ][        example_ftp.c][               example_ftp_test():0216][   0][20009810] example_ftp_test over
[VER  ][           at_utils.c][                at_print_raw_cmd():0071][ 400][20005020] [AT]

[VER  ][           at_utils.c][                at_print_raw_cmd():0072][ 400][20005020]
[VER  ][           at_utils.c][                at_print_raw_cmd():0071][ 400][20005020] [AT]

[VER  ][           at_utils.c][                at_print_raw_cmd():0072][ 400][20005020]
```

Figure 13: Successful Uploading

By opening the FTP Server folder, it is vivid the **112.txt** from SD card is uploaded to the folder of Barry/new.txt.

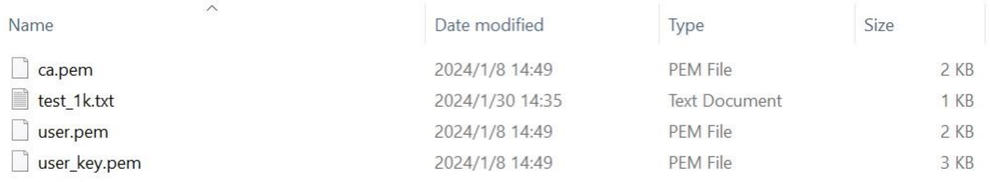


new.txt	8,040	文...	2024/3/13 14:47:00	-rw---...	ftp
ca.pem	1,627	PE...	2024/2/12 20:13:00	-rw---...	ftp
333.txt	90	文...	2024/2/7 16:52:00	-rw---...	ftp
2barry.txt	90	文...	2024/2/7 16:52:00	-rw---...	ftp
222.txt	90	文...	2024/2/7 16:52:00	-rw---...	ftp
111.txt	90	文...	2024/2/7 16:52:00	-rw---...	ftp

Figure 14: Location of the uploaded file

2.6.3. HTTPS-POST

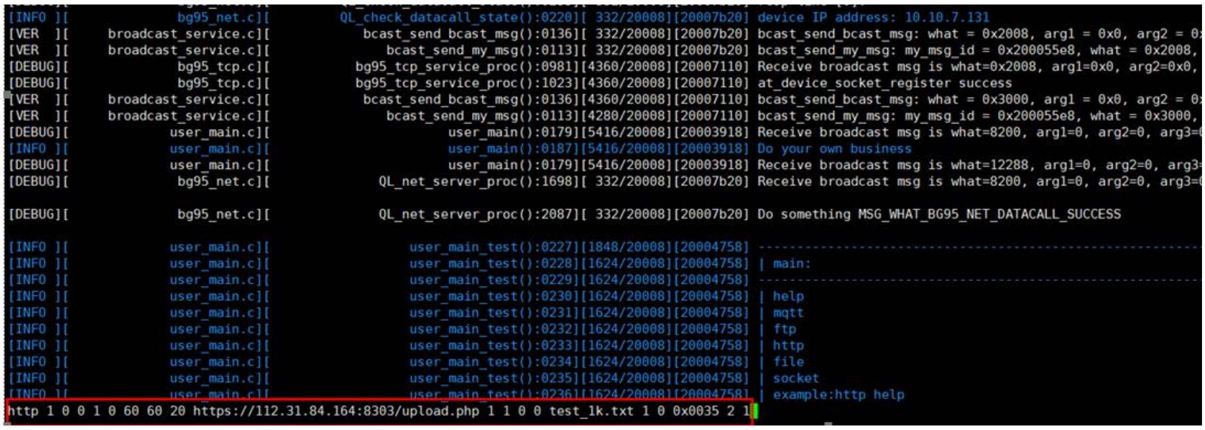
In order to test HTTPS POST (two-way authentication), please make sure that the certificates (**ca.pem**, **user.pem**, **user_key.pem**) and the post file (**test_1k.txt**) are stored in the SD card. Please refer to **Chapter 4** to get test files.



Name	Date modified	Type	Size
ca.pem	2024/1/8 14:49	PEM File	2 KB
test_1k.txt	2024/1/30 14:35	Text Document	1 KB
user.pem	2024/1/8 14:49	PEM File	2 KB
user_key.pem	2024/1/8 14:49	PEM File	3 KB

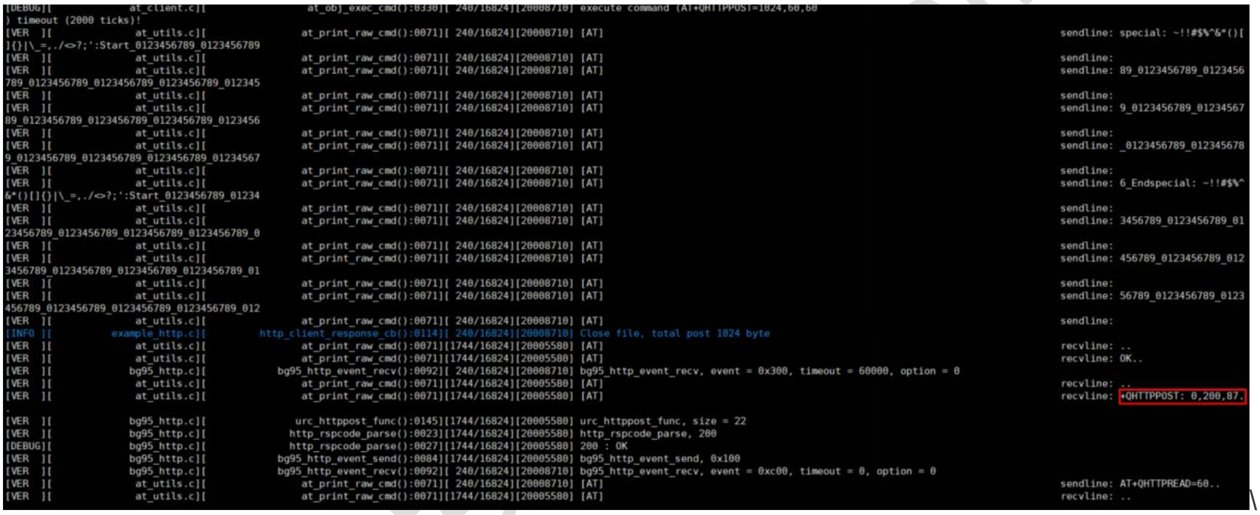
Figure 15: Test Files

1. After resetting board, it will show “**do your down business**”;
2. Execute the command “**main**”;
3. Execute the command “**http 1 0 0 1 0 60 60 20 https://112.31.84.164:8303/upload.php 1 1 0 0 test_1k.txt 1 0 0x0035 2 1**”;
4. Check the path and name of the posted file;
5. Open the HTTP server and reach the file of **7feac070-ef32-f56a-3996-20fe1ed8cad4**;



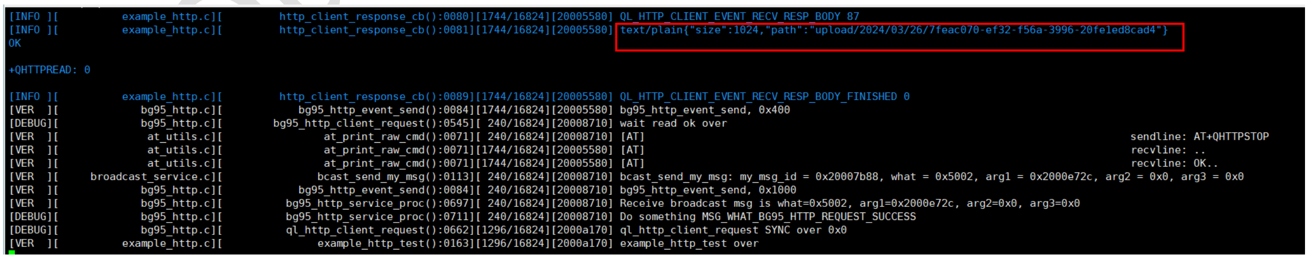
```
[INFO ][           bg95_net.c][        QL_check_datacall_state():0220][ 332/20008][20007b20] device IP address: 10.10.7.131
[VER  ][    broadcast_service.c][          bcast_send_bcast_msg():0136][ 332/20008][20007b20] bcast_send_bcast_msg: what = 0x2008, arg1 = 0x0, arg2 = 0
[VER  ][    broadcast_service.c][             bcast_send_my_msg():0113][ 332/20008][20007b20] bcast_send_my_msg: my_msg_id = 0x200055e8, what = 0x2008,
[DEBUG][           bg95_tcp.c][        bg95_tcp_service_proc():0981][4360/20008][20007110] Receive broadcast msg is what=0x2008, arg1=0x0, arg2=0x0,
[DEBUG][           bg95_tcp.c][        bg95_tcp_service_proc():1023][4360/20008][20007110] at_device_socket_register success
[VER  ][    broadcast_service.c][          bcast_send_bcast_msg():0136][4360/20008][20007110] bcast_send_bcast_msg: what = 0x3000, arg1 = 0x0, arg2 = 0
[VER  ][    broadcast_service.c][             bcast_send_my_msg():0113][4280/20008][20007110] bcast_send_my_msg: my_msg_id = 0x200055e8, what = 0x3000,
[DEBUG][          user_main.c][                     user_main():0179][5416/20008][20003918] Receive broadcast msg is what=8200, arg1=0, arg2=0, arg3=
[INFO ][          user_main.c][                     user_main():0187][5416/20008][20003918] Do your own business
[DEBUG][          user_main.c][                     user_main():0179][5416/20008][20003918] Receive broadcast msg is what=12288, arg1=0, arg2=0, arg3
[DEBUG][           bg95_net.c][           QL_net_server_proc():1698][ 332/20008][20007b20] Receive broadcast msg is what=8200, arg1=0, arg2=0, arg3=

[DEBUG][           bg95_net.c][           QL_net_server_proc():2087][ 332/20008][20007b20] Do something MSG_WHAT_BG95_NET_DATACALL_SUCCESS

[INFO ][          user_main.c][                user_main_test():0227][1848/20008][20004758] ---------------------------------------------------------
[INFO ][          user_main.c][                user_main_test():0228][1624/20008][20004758] | main:
[INFO ][          user_main.c][                user_main_test():0229][1624/20008][20004758] ---------------------------------------------------------
[INFO ][          user_main.c][                user_main_test():0230][1624/20008][20004758] | help
[INFO ][          user_main.c][                user_main_test():0231][1624/20008][20004758] | mqtt
[INFO ][          user_main.c][                user_main_test():0232][1624/20008][20004758] | ftp
[INFO ][          user_main.c][                user_main_test():0233][1624/20008][20004758] | http
[INFO ][          user_main.c][                user_main_test():0234][1624/20008][20004758] | file
[INFO ][          user_main.c][                user_main_test():0235][1624/20008][20004758] | socket
[INFO ][          user_main.c][                user_main_test():0236][1624/20008][20004758] | example:http help
http 1 0 0 1 0 60 60 20 https://112.31.84.164:8303/upload.php 1 1 0 0 test_1k.txt 1 0 0x0035 2 1
```

Figure 16: HTTPS-POST Command



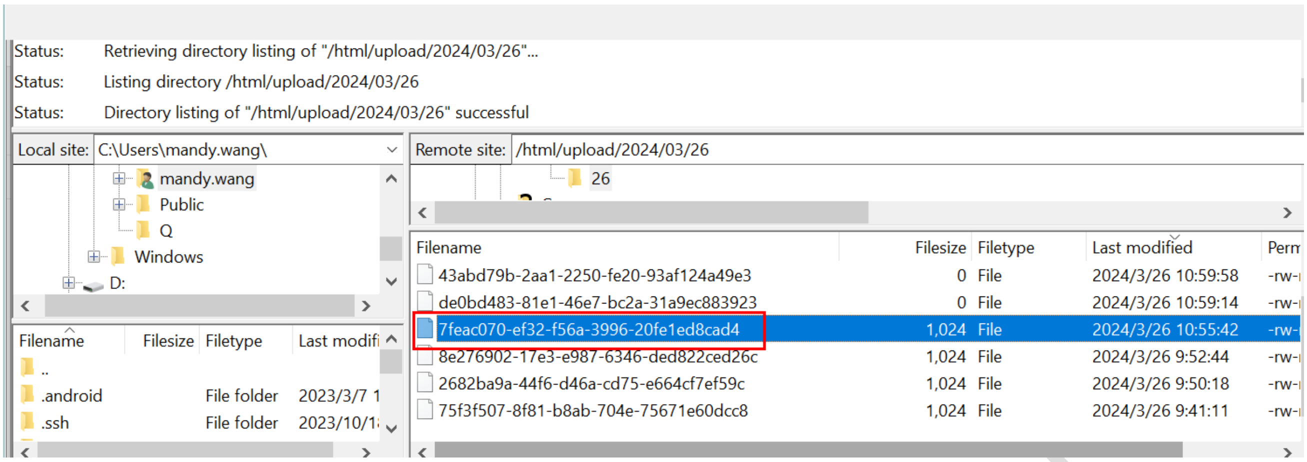
```
[DEBUG][          at_client.c][            at_obj_exec_cmd():0330][ 240/16824][20008710] execute command (AT+QHTTPPOST=1024,60,60
) timeout (2000 ticks)!
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline: special: ~!!#$%^&*()[
]{}|\_=,./<>?;':Start_0123456789_0123456789
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline:
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline: 89_0123456789_0123456
789_0123456789_0123456789_0123456789_012345
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline:
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline: 9_0123456789_01234567
89_0123456789_0123456789_0123456789_0123456
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline:
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline: _0123456789_012345678
9_0123456789_0123456789_0123456789_01234567
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline:
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline: 6_Endspecial: ~!!#$%^
&*()[]{}|\_=,./<>?;':Start_0123456789_01234
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline:
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline: 3456789_0123456789_01
23456789_0123456789_0123456789_0123456789_0
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline:
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline: 456789_0123456789_012
3456789_0123456789_0123456789_0123456789_01
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline:
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline: 56789_0123456789_0123
456789_0123456789_0123456789_0123456789_012
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline:
[INFO ][       example_http.c][   http_client_response_cb():0114][ 240/16824][20008710] Close file, total post 1024 byte
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][1744/16824][20005580] [AT]                                   recvline: ..
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][1744/16824][20005580] [AT]                                   recvline: OK..
[VER  ][          bg95_http.c][      bg95_http_event_recv():0092][ 240/16824][20008710] bg95_http_event_recv, event = 0x300, timeout = 60000, option = 0
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][1744/16824][20005580] [AT]                                   recvline: ..
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][1744/16824][20005580] [AT]                                   recvline: +QHTTPPOST: 0,200,87.
.
[VER  ][          bg95_http.c][        urc_httppost_func():0145][1744/16824][20005580] urc_httppost_func, size = 22
[VER  ][          bg95_http.c][       http_rspcode_parse():0023][1744/16824][20005580] http_rspcode_parse, 200
[DEBUG][          bg95_http.c][       http_rspcode_parse():0027][1744/16824][20005580] 200 : OK
[VER  ][          bg95_http.c][      bg95_http_event_send():0084][1744/16824][20005580] bg95_http_event_send, 0x100
[VER  ][          bg95_http.c][      bg95_http_event_recv():0092][ 240/16824][20008710] bg95_http_event_recv, event = 0xc00, timeout = 0, option = 0
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                   sendline: AT+QHTTPREAD=60..
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][1744/16824][20005580] [AT]                                   recvline: ..
```

Figure 17: POST Successfully



```
[INFO ][       example_http.c][    http_client_response_cb():0080][1744/16824][20005580] QL_HTTP_CLIENT_EVENT_RECV_RESP_BODY 87
[INFO ][       example_http.c][    http_client_response_cb():0081][1744/16824][20005580] text/plain{"size":1024,"path":"upload/2024/03/26/7feac070-ef32-f56a-3996-20fe1ed8cad4"}
OK

+QHTTPREAD: 0

[INFO ][       example_http.c][    http_client_response_cb():0089][1744/16824][20005580] QL_HTTP_CLIENT_EVENT_RECV_RESP_BODY_FINISHED 0
[VER  ][          bg95_http.c][       bg95_http_event_send():0084][1744/16824][20005580] bg95_http_event_send, 0x400
[DEBUG][          bg95_http.c][  bg95_http_client_request():0545][ 240/16824][20008710] wait read ok over
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][ 240/16824][20008710] [AT]                                       sendline: AT+QHTTPSTOP
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][1744/16824][20005580] [AT]                                       recvline: ..
[VER  ][           at_utils.c][           at_print_raw_cmd():0071][1744/16824][20005580] [AT]                                       recvline: OK..
[VER  ][  broadcast_service.c][          bcast_send_my_msg():0113][ 240/16824][20008710] bcast_send_my_msg: my_msg_id = 0x20007b88, what = 0x5002, arg1 = 0x2000e72c, arg2 = 0x0, arg3 = 0x0
[VER  ][          bg95_http.c][       bg95_http_event_send():0084][ 240/16824][20008710] bg95_http_event_send, 0x1000
[VER  ][          bg95_http.c][     bg95_http_service_proc():0697][ 240/16824][20008710] Receive broadcast msg is what=0x5002, arg1=0x2000e72c, arg2=0x0, arg3=0x0
[DEBUG][          bg95_http.c][     bg95_http_service_proc():0711][ 240/16824][20008710] Do something MSG_WHAT_BG95_HTTP_REQUEST_SUCCESS
[DEBUG][          bg95_http.c][     ql_http_client_request():0662][1296/16824][2000a170] ql_http_client_request SYNC over 0x0
[VER  ][       example_http.c][         example_http_test():0163][1296/16824][2000a170] example_http_test over
```

Figure 18: Path of the Posted File



```
Status:    Retrieving directory listing of "/html/upload/2024/03/26"...
Status:    Listing directory /html/upload/2024/03/26
Status:    Directory listing of "/html/upload/2024/03/26" successful
```

Local site: C:\Users\mandy.wang\

Filename	Filesize	Filetype	Last modifi
..			
.android		File folder	2023/3/7 1
.ssh		File folder	2023/10/1

Remote site: /html/upload/2024/03/26

Filename	Filesize	Filetype	Last modified	Perm
43abd79b-2aa1-2250-fe20-93af124a49e3	0	File	2024/3/26 10:59:58	-rw-
de0bd483-81e1-46e7-bc2a-31a9ec883923	0	File	2024/3/26 10:59:14	-rw-
7feac070-ef32-f56a-3996-20fe1ed8cad4	1,024	File	2024/3/26 10:55:42	-rw-
8e276902-17e3-e987-6346-ded822ced26c	1,024	File	2024/3/26 9:52:44	-rw-
2682ba9a-44f6-d46a-cd75-e664cf7ef59c	1,024	File	2024/3/26 9:50:18	-rw-
75f3f507-8f81-b8ab-704e-75671e60dcc8	1,024	File	2024/3/26 9:41:11	-rw-

Figure 19: Post File Successfully

3 Functions

3.1. Save Log in SD Card

Table 2: SD Function Definition

Function	Commands
Log level configuration	level 0
Log level configuration	level 1
Log level configuration	level 2
Log level configuration	level 3
Log level configuration	level 4
SD-storage Start	Save 1
SD-storage Stop	Save 0

NOTE

save (on:1, off:0)
level (v:0, D:1, I:2, W:3, E:4)
level v: Level 0, Print Version
level D: Level 1, Print Debug
level I: Level 2, Print Information
level W: Level 3, Print Warning
level E: Level 4, Print ERROR
Save 1: Start to store the log into SD card
Save 0: Stop storing the log into SD card.

3.2. TCP&UDP

Table 3: TCP&UDP Function Definition

Function	Commands	Log
TCP-Client	socket 0 112.31.84.164 8305 1 5000	tcp-client.txt
UDP-Client	socket 1 112.31.84.164 8305 1 5000	udp-server.txt
TCP-Server	socket 2 127.0.0.1 2020 100 1000 5	tcp-server.txt
UDP-Server	socket 3 127.0.0.1 2020 1 5000	udp-server.txt

NOTE

Instruction format : socket socket_type ip port count interval_ms max_connect_num

socket_type:

0: TCP

1: UDP

2: TCP SERVER

3: UDP SERVER

Ip : ip address

Port : port

interval_ms : Interval for sending data

max_connect_num : Maximum connection request (only tcp server needs to set)

3.3. FTP(S)

Table 4: FTP(S) Function Definition

Function	Commands	Log
FTP-list	ftp list "220.180.239.212" 8309 "/Barry" "0:2323.txt" "22344.txt" 0	ftp-list.txt
FTP-download	ftp 1 test test 1 1 100 "220.180.239.212" 8309 2 "/Barry" "0:111.txt" "22344.txt" 0	ftp-download.txt
FTP-upload	ftp 1 test test 1 1 100 "220.180.239.212" 8309 3 "/Barry" "0:111.txt" "22344_new.txt" 0	ftp-upload.txt
FTPS-list	ftp list "112.31.84.164" 8311 "/Barry" "0:2323.txt" "22344.txt" 1 1 0xfff 1 4	ftps-list.txt
FTPS-download	ftp 1 test test 1 1 100 "112.31.84.164" 8311 2 "/Barry" "0:111.txt"	ftps-download.txt

	"test5.txt" 1 1 0xffff 1 4	
FTPS-upload	ftp 1 test test 1 1 100 "112.31.84.164" 8311 3 "/Barry" "0:111.txt" "test5.txt" 1 1 0xffff 1 4	ftps-upload.txt

NOTE

Instruction format: ftp contextid username password filetype transmode rsptimeout hostname port ftp_type directoryToSet local_name rem_name sslenble sslctxid ciphersuite seclevel sslversion
contextid: PDP context ID
username: Username for logging in to the Ftp(S) server
password: Password for logging in to the Ftp(S) server
file_type: The type of transferred data
0: Binary
1: ASCII
transmode: Whether the FTP(S) server or client listens on a port for data connection
0: Active mode, the module will listen on a port for data connection
1: Passive mode, the FTP(S) server will listen on a port for data connection
rsptimeout:

Range: 20-180.

Default value: 90.
Unit: second.
hostname: FTP(S) server URL
port: FTP(S) server port
ftp_type: FTP fun mode
1: file list
2: file get
3: file uploader
directoryToSet: The directory of the server
local_name: Data path in SD card
rem_name: The file name of the server
sslenble : Whether ssl is enabled
0: Disable SSL
1: Enable SSL
sslctxid: SSL context ID used for HTTP(S). Range: 0-5
ciphersuite: Numeric type in HEX format. SSL cipher suites
seclevel: Authentication mode
0: No authentication
1: Perform server authentication
2: Perform server and client authentication if requested by the remote server
sslversion: SSL Version
0: SSL3.0
1: TLS1.0

3: TLS1.2
4: ALL

3.4. HTTP(S)

Table 5: HTTP(S) Function Definition

Function	Example commands	Log
HTTP-POST	http 1 0 0 1 0 60 60 20 http://112.31.84.164:8300/upload.php 1 1 0 0 test_1k.txt 0	http-post.txt
HTTP-GET	http 1 0 0 1 0 60 60 2 http://112.31.84.164:8300/upload/2024/01/30/dee1f27e-87b8-e231-fd1c-2c3fbd707da6 0 1 0 0 get_1k.txt 0	http-get.txt
HTTPS-POST	http 1 0 0 1 0 60 60 20 https://112.31.84.164:8303/upload.php 1 1 0 0 test_1k.txt 1 0 0x0035 2 1	https-post.txt
HTTPS-GET	http 1 0 0 1 0 60 60 2 http://112.31.84.164:8303/1024.txt 0 1 0 0 get_1k.txt 1 0 0x0035 2 1	https-get.txt

NOTE

Instruction format: http contextid requestheader responseheader contenttype custom_header timeout rsptime wait_time request_url method request_mode username password sd_card_path sslenble sslctxid ciphersuite seclevel sslversion
contextid: PDP context ID, Range: 1-16
requestheader: Disable or enable customization of HTTP(S) request header
0: Disable
1: Enable
responseheader: Disable or enable the outputting of HTTP(S) response header
0: Disable
1: Enable
contenttype: Data type of HTTP(S) body
0: application/x-www-form-urlencoded
1: text/plain
2: application/octet-stream
3: multipart/form-data
4: application/json
5: image/jpeg
custom_header: User-defined HTTP(S) request header
timeout: The maximum time for inputting URL.

Range: 1-2038.
Unit: second
rsptime: Timeout for the HTTP(S) GET response
Range: 1-65535.
Unit: second
wait_time: Maximum time between receiving two packets of data.
Range: 1-65535.
Unit: second
request_url: HTTP(S) server URL
method: Request type
0: Get
1: Post
request_mode: Request mode
0: Async
1: Sync
username: Username for logging in the HTTP(S) server
password: Password for logging in the HTTP(S) server
sd_card_path: Data path in SD card
sslenble: Whether ssl is enabled
0: Disable SSL
1: Enable SSL
sslctxid: SSL context ID used for HTTP(S)，Range: 0-5
ciphersuite: Numeric type in HEX format. SSL cipher suites
seclevel: Authentication mode
0: No authentication
1: Perform server authentication
2: Perform server and client authentication if requested by the remote server
sslversion: SSL Version
0: SSL3.0
1: TLS1.0
3: TLS1.2
4: ALL

3.5. PSM

Table 6: PSM Function Definition

Function	Commands	Log

PSM-enable	psm enable	psm-enable.txt
PSM-setting	psm setting 00000100 0000111	psm-setting.txt
PSM-disable	psm disable	psm-disable.txt
PSM-modem-optimization	psm modem 2 2 120 5 120 3	psm-modem.txt
PSM-stat	psm stat	psm-stat.txt
PSM-threshold	psm threshold 100	psm-threshold.txt

NOTE

Instruction format: psm enable/disable

psm settings - TAU/active time (ex setting 00000100 00001111)

0: Requested Periodic TAU

1: Requested Active Time

psm threshold - sets the minimum threshold value to enter PSM(ex threshold 100)

psm modem Optimization - sets the Modem Optimization (ex modem 2 2 120 5 120 3)

0: PSM opt mask

1: PSM max oos full scans

2: PSM duration due to oos

3: PSM randomization window

4: PSM max oos time

5: PSM early wakeup time

psm stat - show all psm setting

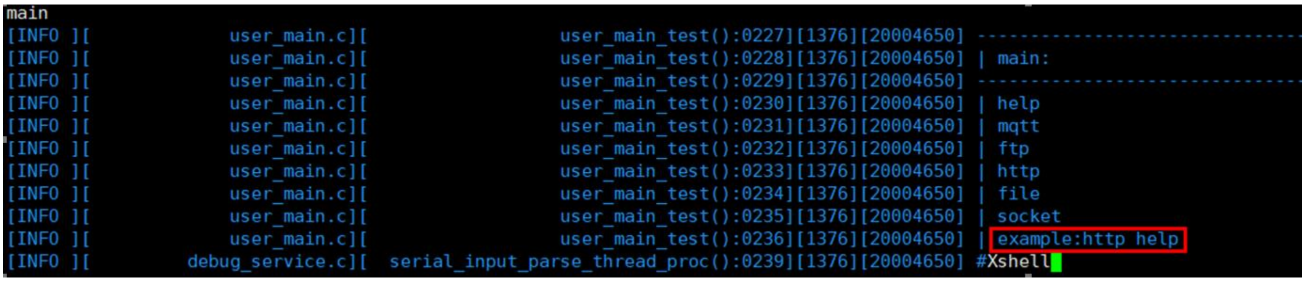
3.6. MQTT

NOTE

…

3.7. Function Help

Execute the command “**Function Help**” and you can get the information the command.



```
main
[INFO ][          user_main.c][                 user_main_test():0227][1376][20004650] --------------------------------
[INFO ][          user_main.c][                 user_main_test():0228][1376][20004650] | main:
[INFO ][          user_main.c][                 user_main_test():0229][1376][20004650] --------------------------------
[INFO ][          user_main.c][                 user_main_test():0230][1376][20004650] | help
[INFO ][          user_main.c][                 user_main_test():0231][1376][20004650] | mqtt
[INFO ][          user_main.c][                 user_main_test():0232][1376][20004650] | ftp
[INFO ][          user_main.c][                 user_main_test():0233][1376][20004650] | http
[INFO ][          user_main.c][                 user_main_test():0234][1376][20004650] | file
[INFO ][          user_main.c][                 user_main_test():0235][1376][20004650] | socket
[INFO ][          user_main.c][                 user_main_test():0236][1376][20004650] | example:http help
[INFO ][       debug_service.c][  serial_input_parse_thread_proc():0239][1376][20004650] #Xshell
```

Figure 20: Reference Circuit of the Help

4 Test Files

When testing FTPS/HTTPS/MQTTS, we need to put certifications into the SD card. Additionally, the post file such as **test_1k.txt** shall also be placed on the SD card in order to test HTTP.
Please check test files to get these files.

Table 7: Certificates for the Test

Functions	Certificate
ftps	ca.pem
https	ca.pem user.pem user_key.pem
mqtts	/

5 Appendix A References

Table 8: Related Documents

Document Name
[1] Quectel_BG95_Quick_Development Manual
[2] Quectel LPWA&LTE Carrier board of STM32 Nucleo-64 MB1136