QSTM32
Test Guide

Version: 2.1

Date: 2025-12-26

Status: Released

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	2025-06-16	Mandy.Wang	Initial Version
2.0	2025-08-26	Mandy.Wang	1. Added IoT Application Protocol Online Testing Platform (Chapter 2.4.1). 2. Updated Save log Test commands (Chapter 3.1) 3. Updated HTTP(S) Test commands (Chapter 3.4). 4. Updated the Test Files (Chapter 4) 5. Updated all log (Chapter 4)
2.1	2025-12-26	Mandy.Wang	1. Added Update Module firmware (Chapter 2.4) 2. Updated PSM Test command (Chapter 3.6) 3. Added FOTA (Chapter 3.8)

Contents

Figure Index

Table Index

1 Introduction

The article mainly illustrates how to test common functions (TCPUDP/FTP(s)/HTTP(s)/PSM/MQTT(s)) in Quectel STM32 LQFP64 EVK V2.0 board and Quectel Wireless Cellular LTE/LPWA Module TE-A board.

2 Test Setup

This chapter introduces how to use the STM32 EVK and module TE-A for testing. Before starting the procedures below, please ensure modules and the STM32 EVK are correctly assembled.

2.1. Preparation

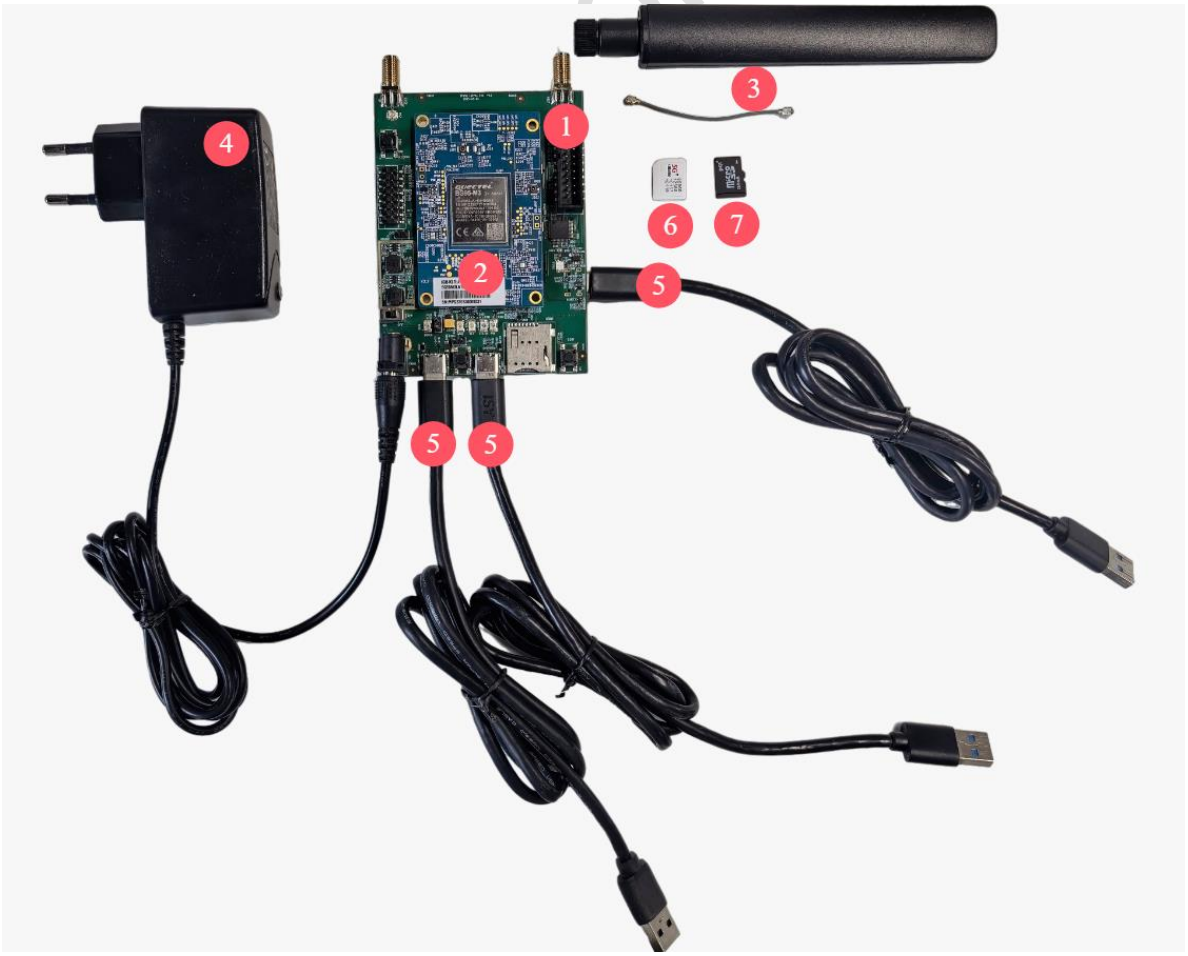

Figure 1: Accessories Assembly

Table 1: Accessories List

ID	Components
1	STM32 LQFP64 EVK V2.0 Adapter board
2	Wireless Cellular Module TE-A board
3	RF cable, and Cellular antenna
4	Power Supply DC Adapter 5V
5	USB Cable, used to download the STM32 firmware, Module firmware and do function test
6	SIM Card
7	SD Card: Store the certificate, test files, and save log

2.2. Operation Procedures

2.2.1. Turn on the device

1. Connect the module TE-A to the STM32 EVK via connectors J101 and J102.
2. Insert a (U)SIM card into the USIM1 card connector on EVK.
3. Insert a SD card into the SD1 connector.
4. Use RF cables to connect the module TE-A to the EVB, and connect the antenna to the EVK. Or connect antennas to the module TE-A directly.
5. Connect the STM32 EVK to a 5 V/ 2 A power, then switch S201 to the “ON” side. Then D206 (power supply ON/OFF indicator) will light up, which indicates that the power supply for the whole EVK board is ready
6. Since the STM32 EVK has already been programmed with a firmware by Quectel, so the pre-installed software program will turn on the module automatically.

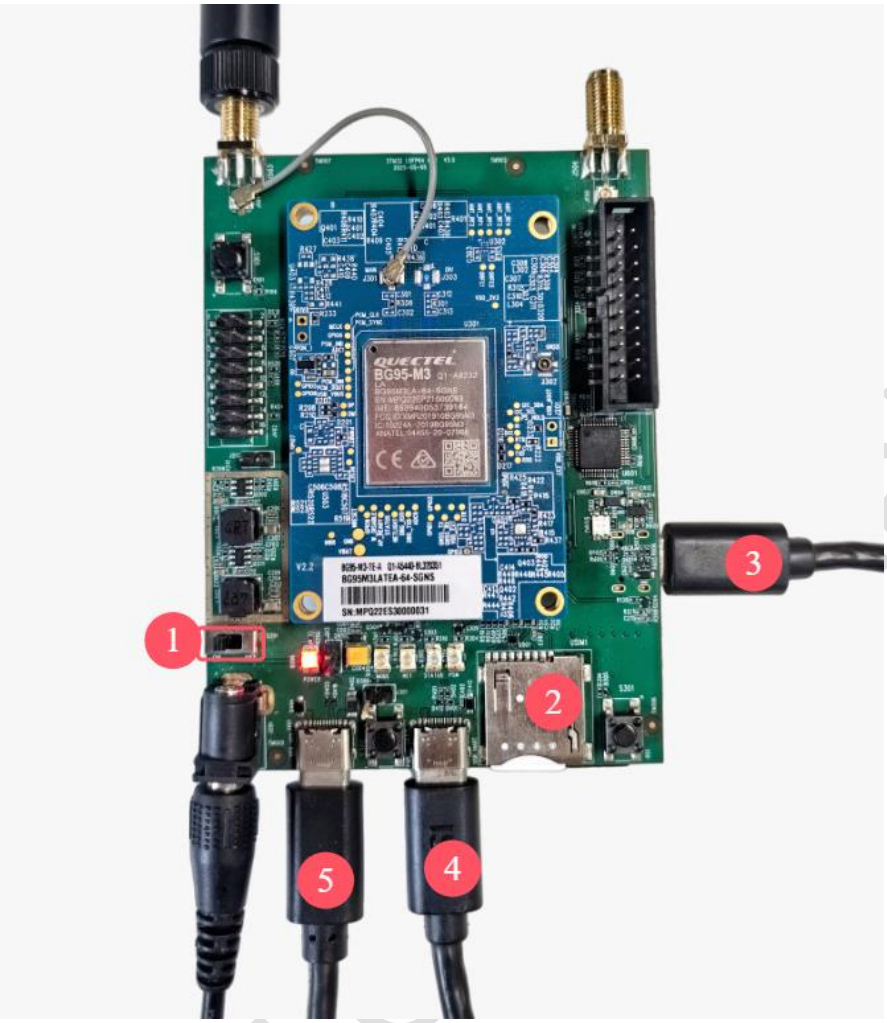

Figure 2: Top View of STM32 EVK

NOTE

❶ S201: Input power control switch
❷ USIM1 Card slot, please insert the SIM card
❸ J602 USB Interface, used to download the firmware
❹ J403 USB Interface, used to do function test, etc.
❺ J401 USB Interface, connected with module USB2.0 interface by default, which can be used to do AT communication, Module firmware update, etc.

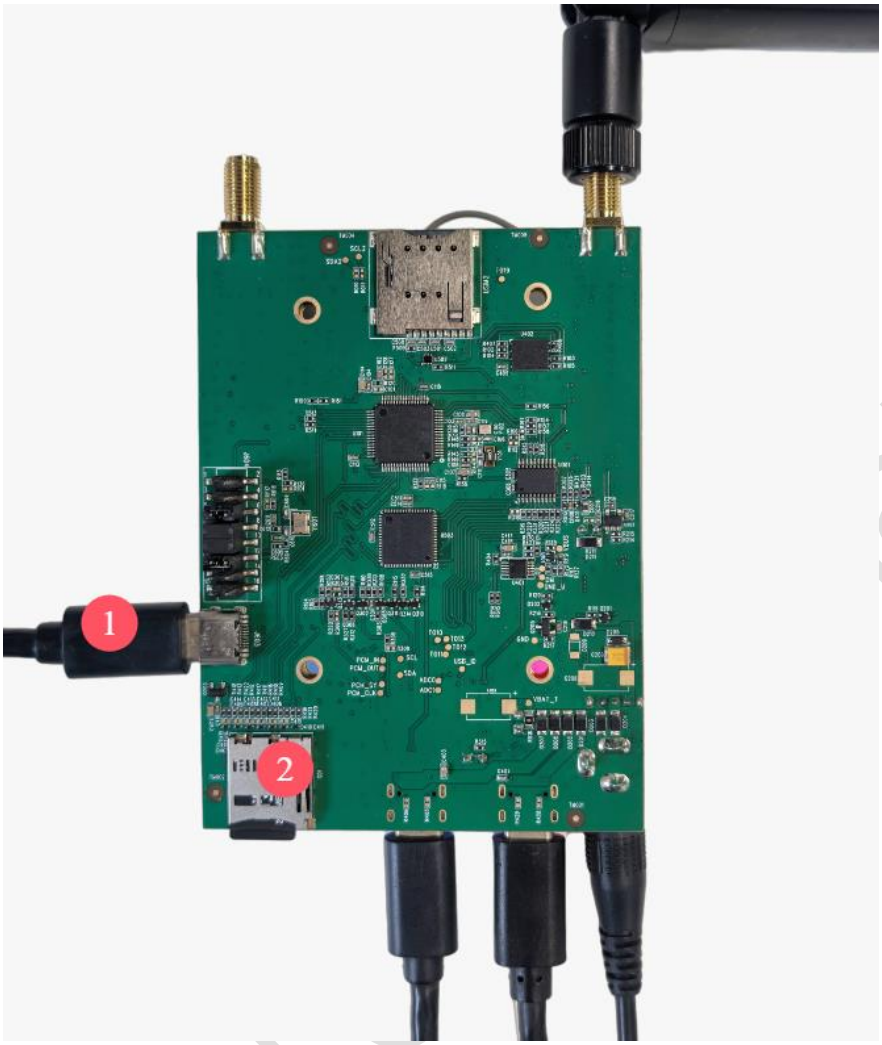

Figure 3: Bottom View of STM32 EVK

NOTE

❶ Jumper, please connect refer to the ***Figure 3***
❷ SD Card slot, please insert the SD card.

2.2.2. Turn off the device

Switch S201 to the “OFF” side, then the device will be turned off.

2.2.3. Port Introduction

1. Turn on the module according to the procedure in ***Chapter 2.2.1***
2. Connect the STM32 EVK and a PC with USB Cable through USB Type-C interface.
   J602 USB interface is used to update STM32 EVK firmware.
   J403 USB interface 1 is connected with MCU debug port by default, which can be used to do function test.

J401 USB interface is connected with module USB2.0 interface by default, which can be used to do AT communication, Module firmware update, etc.

Please refer to ***Figure 2*** to confirm the USB interface position of J602, J403 and J401.

3. Install Driver
   - UART driver: CP210x Universal Windows Driver

Extract the installation package CP210x_Universal_Windows_Driver.zip -> Right-click on **silabser.inf** -> Click **"Install".** Please refer to ***Figure 4*.**

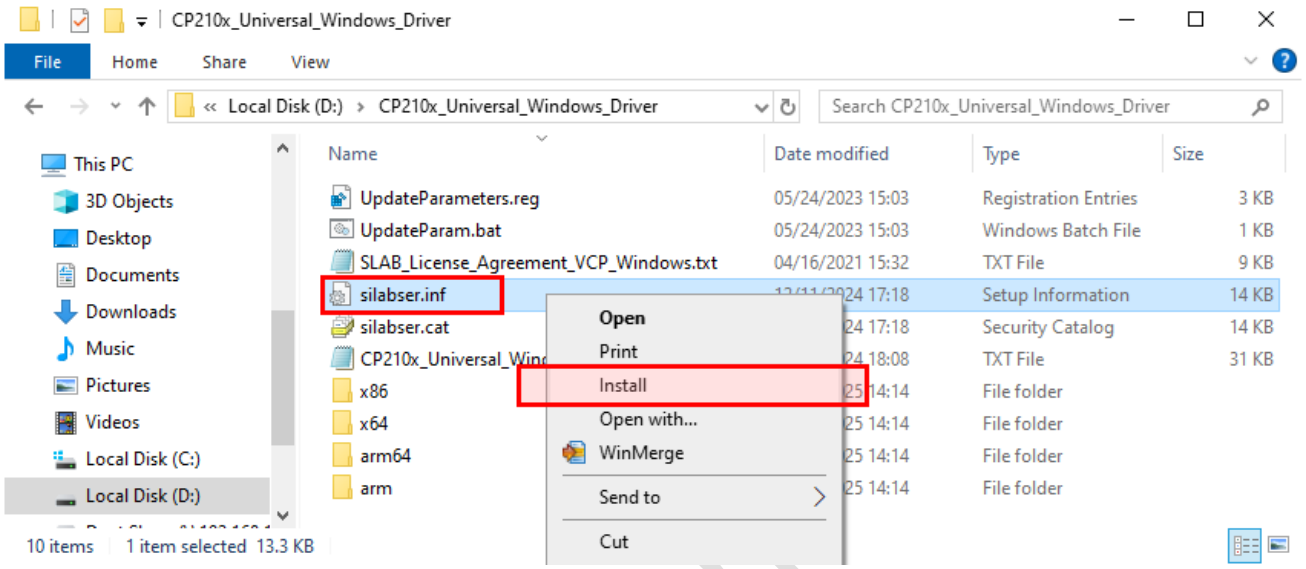

Figure 4: Install the serial port driver

- ST-Link Driver: ST-LINK Windows Driver
- Module USB Driver: Please contact Quectel Technical Support to get the corresponding USB Driver.

4. After the driver is loaded successfully, from PC Device Manager we can find the Ports showed as below.

   The Silicon Labs Quad CP2108 USB to UART Bridge Interface 1 is used to function test.

   The STLink Virtual COM Port is used to download the firmware of STM32. For other port functions, please refer to ***document [1]***.

   Take the Qualcomm module BG95 as an example, the module can be upgraded via the USB DM port.

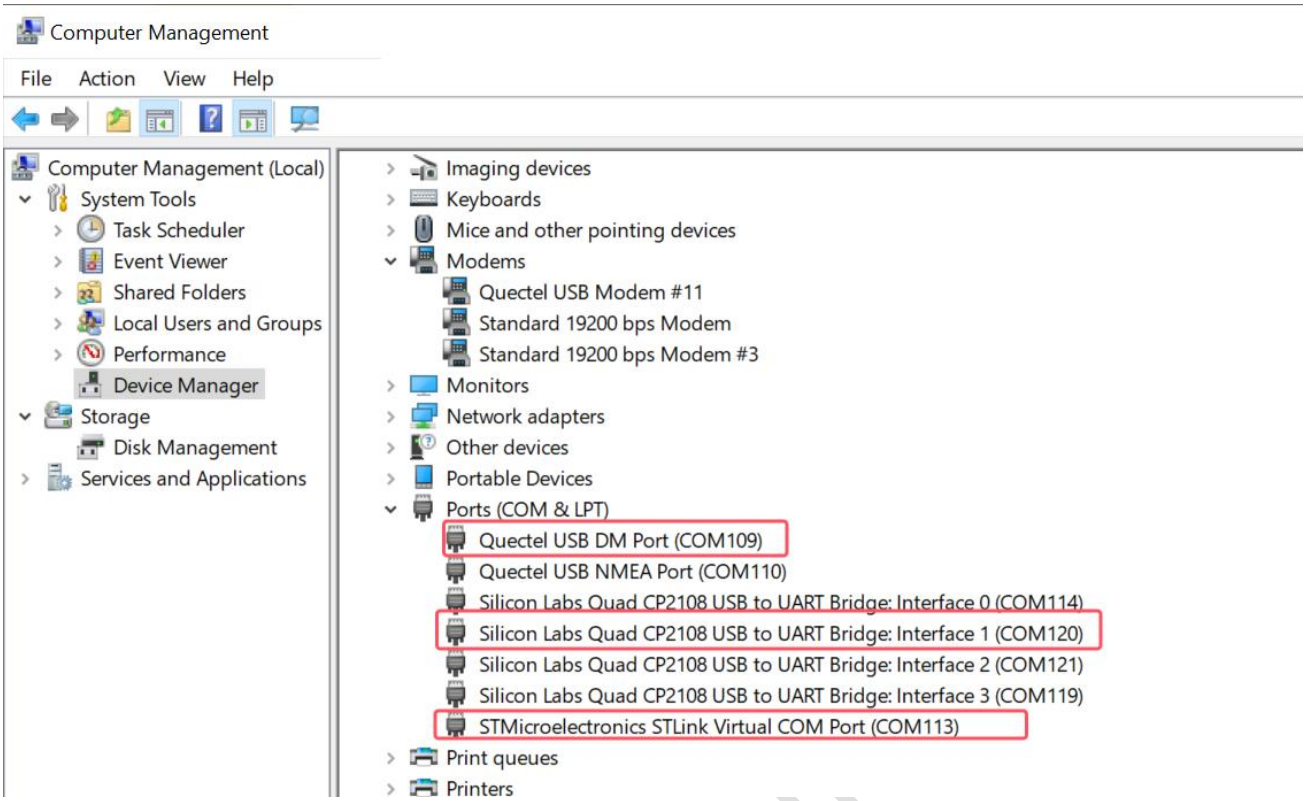

Figure 5: Port Introduction

2.3. Download STM32 firmware

Turn on the device, and connect the USB cable with PC according to the procedure in ***Chapter 2.2.1 and 2.2.3***.

We use the STM32CubeProgrammer tool to download STM32 firmware. Please open the STM32CubeProgrammer tool and refer to following steps to download:

1. Click "**Connect**" button to connect the STM32 EVK board.
2. Open file, select the firmware "Quectel_UFP_STM32F413RGT6_A03.bin", which is the STM32 firmware.
3. Click "**Download**" button to download the firmware, then wait "File download complete" message outputted.
4. Remove J602 USB cable, and switch S201 to the "OFF" side, and then "ON" side, then the device will be restarted successfully.

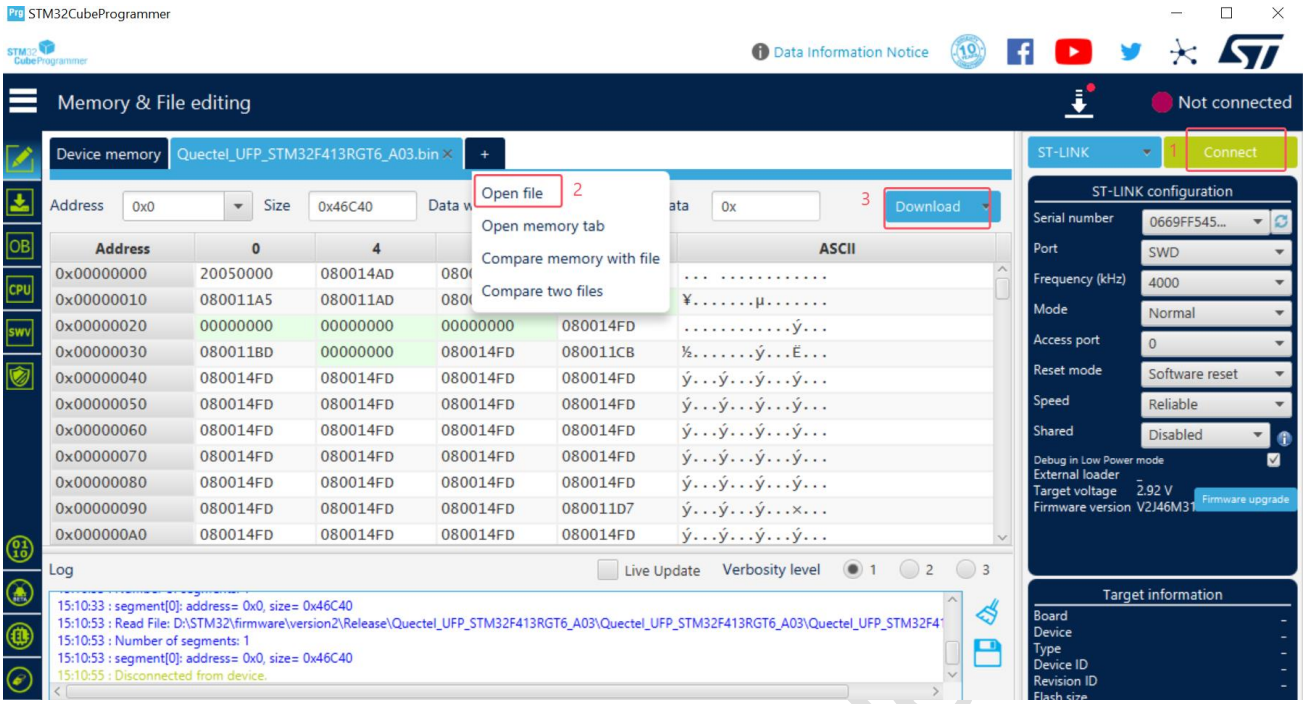

Figure 6: Download STM32 firmware

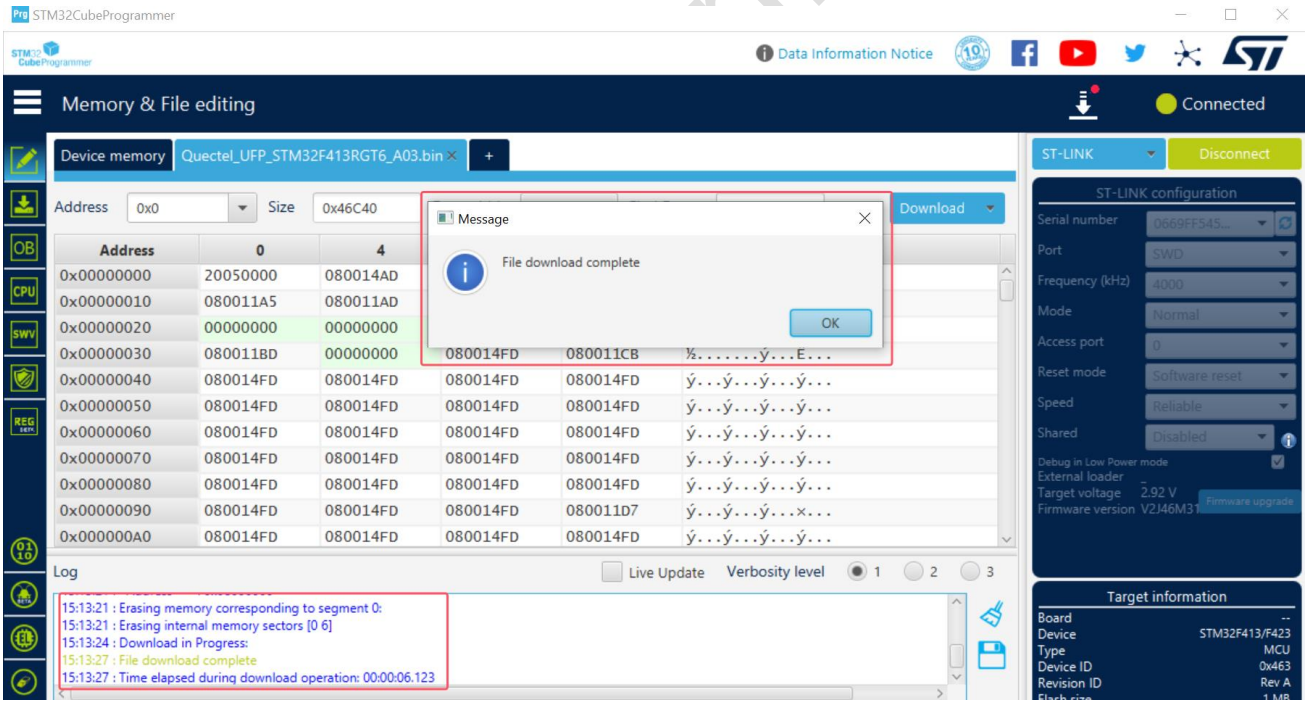

Figure 7: Successful Download

2.4. Update Module firmware

Take the Qualcomm module BG95 as an example, the module can be upgraded via the USB DM port. Please contact Quectel Technical Support to get the QFlash tool.

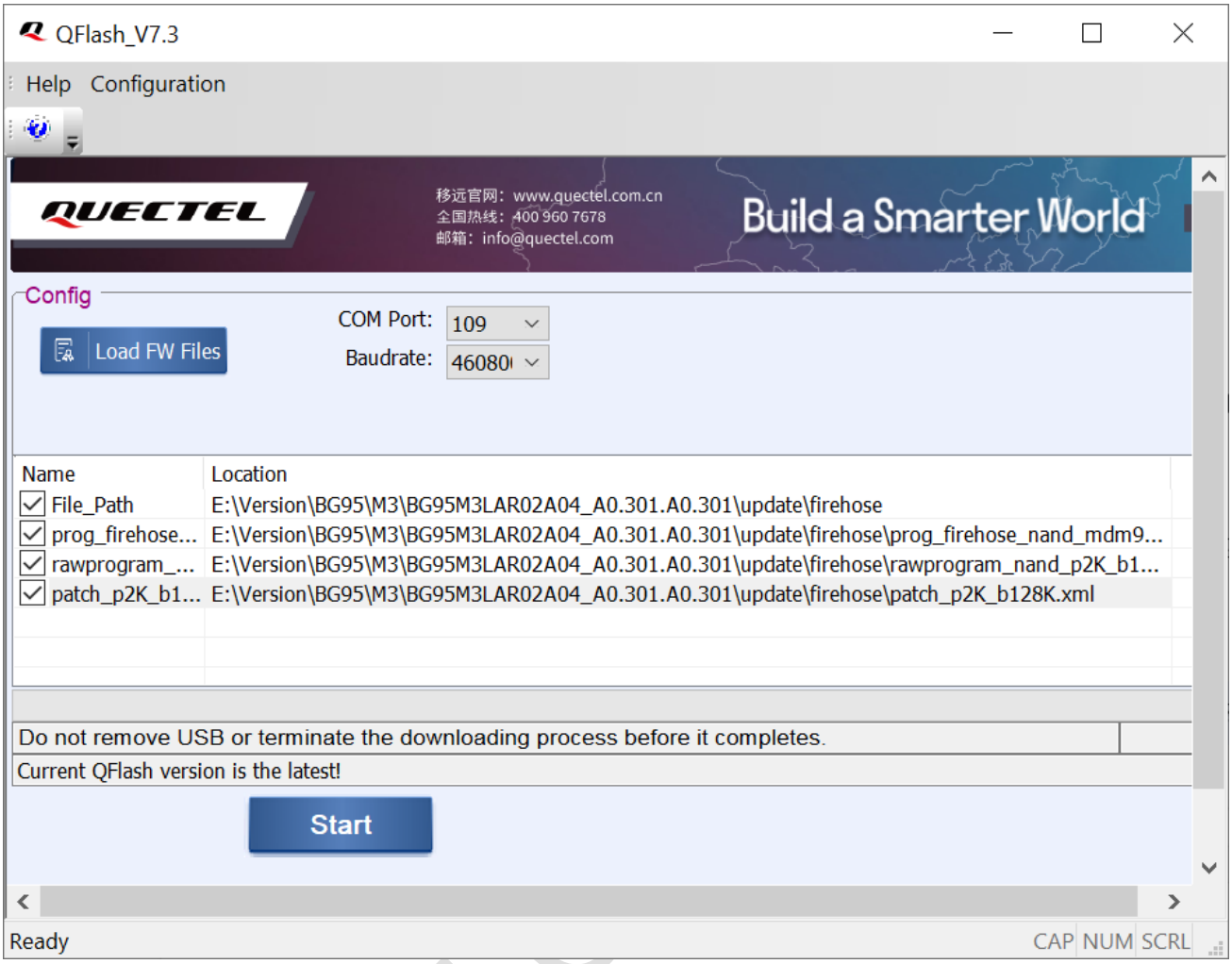

Figure 8: Module firmware Update

2.5. Function Test

Before the function test, please download STM32 firmware according to the procedure in ***Chapter 2.3***. Then Open the Serial communication tool, like Xshell, select the Silicon Labs Quad CP2108 USB to UART Bridge Interface 1. And set baud rate as 115200.

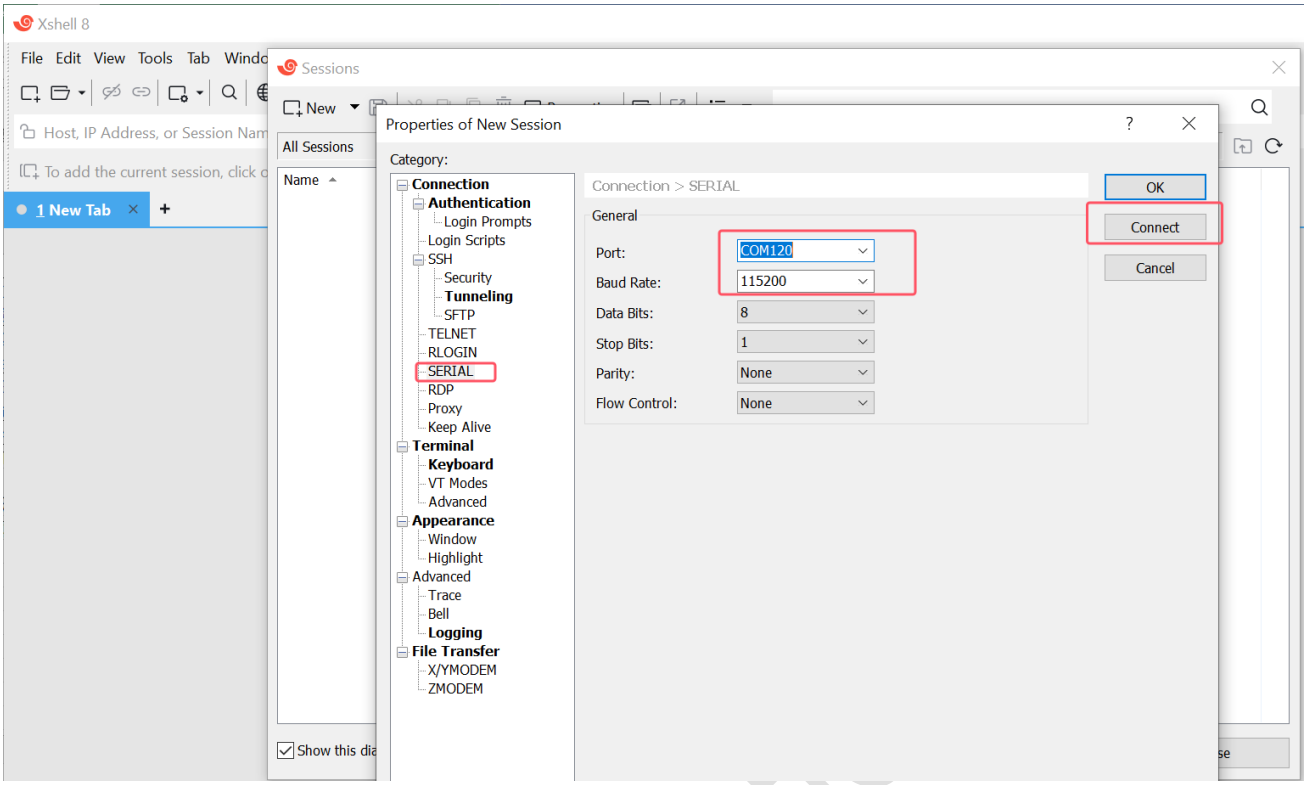

Figure 9: Serial port communication tool

After download STM32 firmware and restart the device, the STM32 will automatically turn on the module. From the Xshell tool, we can check the log is outputted automatically.

```
1970-01-01 00:00:00 [DEBUG]  [user_main.c]      [user_main():15][4680]  Welcome to Quectel User Friendly Project !
1970-01-01 00:00:00 [DEBUG]  [user_main.c]      [user_main():16][4680]  Current version: Quectel_UFP_STM32F413RGT6_A03 @WINDOWS
1970-01-01 00:00:00 [INFO ]  [cli_test_main.c]  [cli_test_main():45][4616]            ============ cli_test_main ============
1970-01-01 00:00:00 [INFO ]  [ql_dev.c  ]       [ql_spi_flash_selftest():75][4448]      ===Detected Flash! DeviceID [0XEF16]
1970-01-01 00:00:00 [INFO ]  [ql_dev.c  ]       [ql_spi_flash_selftest():78][4448]      ===Write: Hello, this is just an external Flash self test code...
1970-01-01 00:00:00 [INFO ]  [ql_dev.c  ]       [ql_spi_flash_selftest():80][4448]      ===Read : Hello, this is just an external Flash self test code...
1970-01-01 00:00:00 [INFO ]  [ql_dev.c  ]       [ql_spi_flash_selftest():84][4448]      ===Matched. Flash test successfully!
1970-01-01 00:00:00 [DEBUG]  [ql_dev.c  ]       [ql_module_hardware_init():31][4448]    Now restart the module...
1970-01-01 00:00:04 [DEBUG]  [ql_dev.c  ]       [ql_module_hardware_init():40][4448]    Restart module done.
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():43][4376] SD card detected !
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():49][4376] Fat System OK
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():52][4376] Initialize SD card successfully!
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():53][4376] SD card information!
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():59][4376] CardCapacity  : 0.23GB
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():67][4376] FreeSpace     : 0.12GB
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():69][4376] CardBlockSize : 512
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():70][4376] LogBlockNbr   : 499712
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():71][4376] LogBlockSize  : 512
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():72][4376] RCA           : 0xB368
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():73][4376] CardType      : 0 (0: <= 2GB; 1: 2GB-32GB; 2: >32GB)
1970-01-01 00:00:04 [DEBUG]  [sd_fatfs.c]       [ql_sd_init():77][4376] ManufacturerID: 0x1a (0x03: SanDisk; 0x1A: ADATA; 0x1B: Samsung; 0x41: Kingston)
1970-01-01 00:00:04 [INFO ]  [sd_fatfs.c]       [ql_sd_init():78][4376] sd card mount success!
1970-01-01 00:00:04 [INFO ]  [broadcast_service.c]      [broadcast_service_create():202][4376]  broadcast_service_create over(20006c28)
1970-01-01 00:00:04 [INFO ]  [debug_service.c]  [debug_cli_service_create():222][4376]  debug_cli_service_create over(2000ad60)
1970-01-01 00:00:05 [DEBUG]  [at_client.c]      [at_client_para_init():1012][4376]      name=at_clnt0
1970-01-01 00:00:05 [DEBUG]  [at_client.c]      [at_client_para_init():1051][4376]      at_client_para_init done.
1970-01-01 00:00:05 [INFO ]  [at_client.c]      [at_client_init():1096][4376]   AT client(V1.3.1) initialize success.
1970-01-01 00:00:05 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][4232]           sendline: AT.
1970-01-01 00:00:05 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2120]           recvline: ..
1970-01-01 00:00:05 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: RDY..
1970-01-01 00:00:05 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: ..
```

Figure 10: Initialization Output

Module initialization like network related log will be outputted. **"Initialization done, do your own business"** message will be displayed till the network registration is done. Then you can execute command to do function test, like TCP/HTTP/FTP/MQTT, etc.

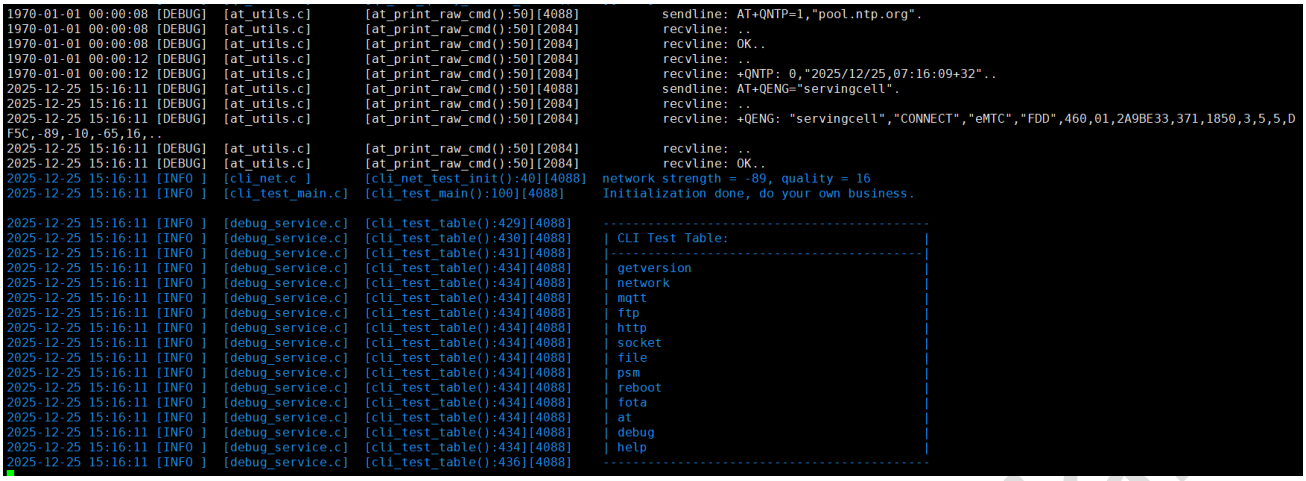

Figure 11: Do your own business

2.5.1. IoT Application Protocol Online Testing Platform

Quectel havs its own IoT Application Protocol Online Testing Platform.
Login address: https://connectlab.phicotek.com. Select Tourist Mode.

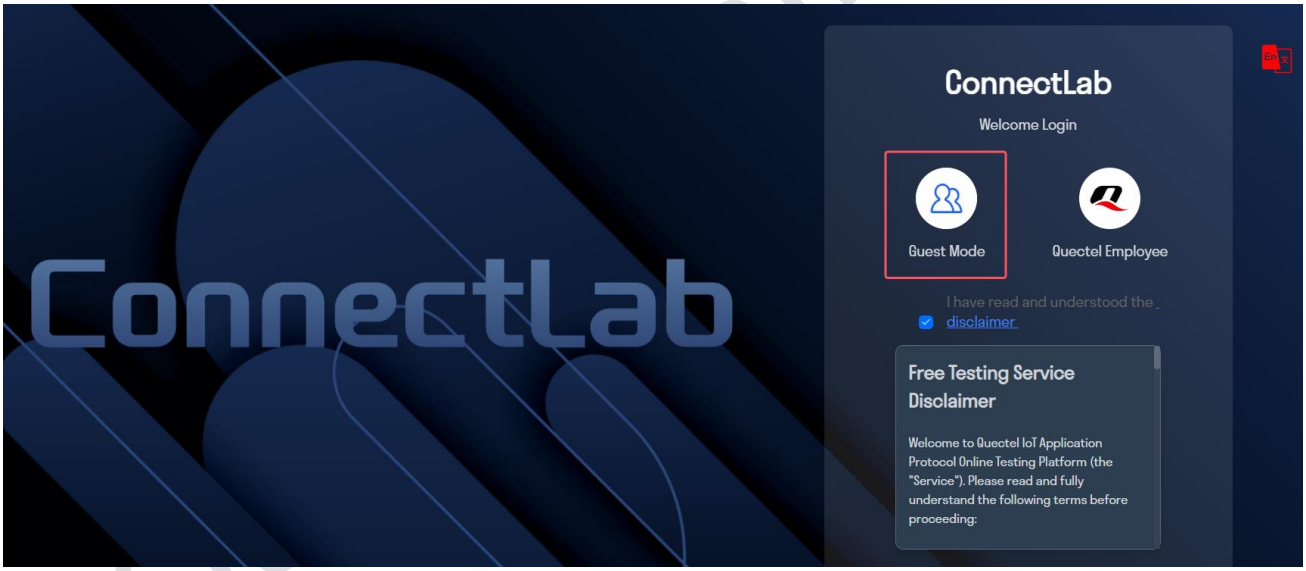

Figure 12: Tourist Mode

If want to create TCP/UDP server, please create TCP/UDP Server as follows.

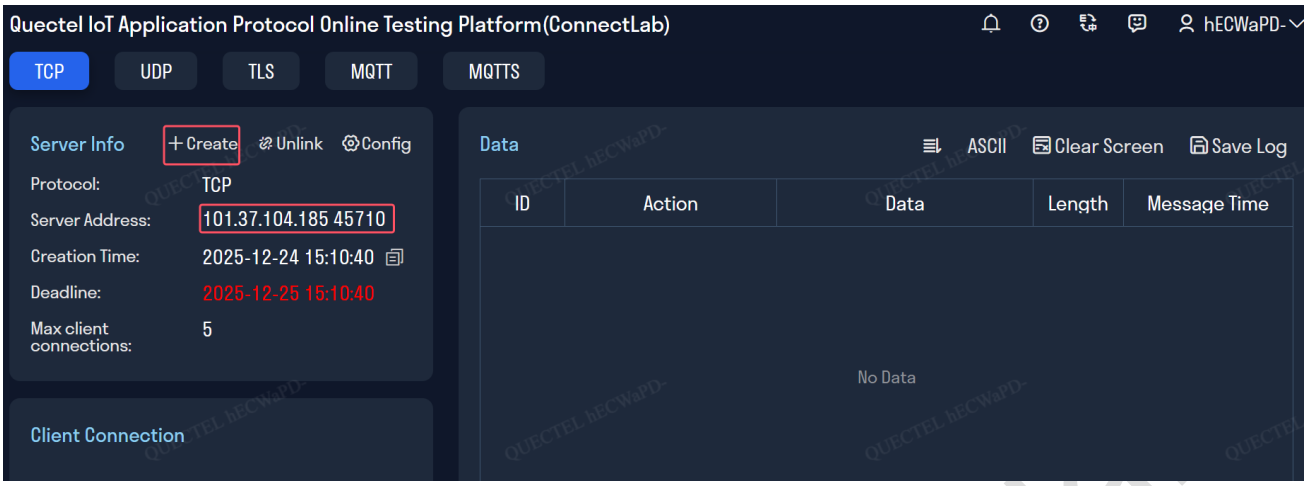

Figure 13: TCP/UDP Server

Support the configuration for automatic sending and automatic reply, such as enable the function of automatically replying to the received data.

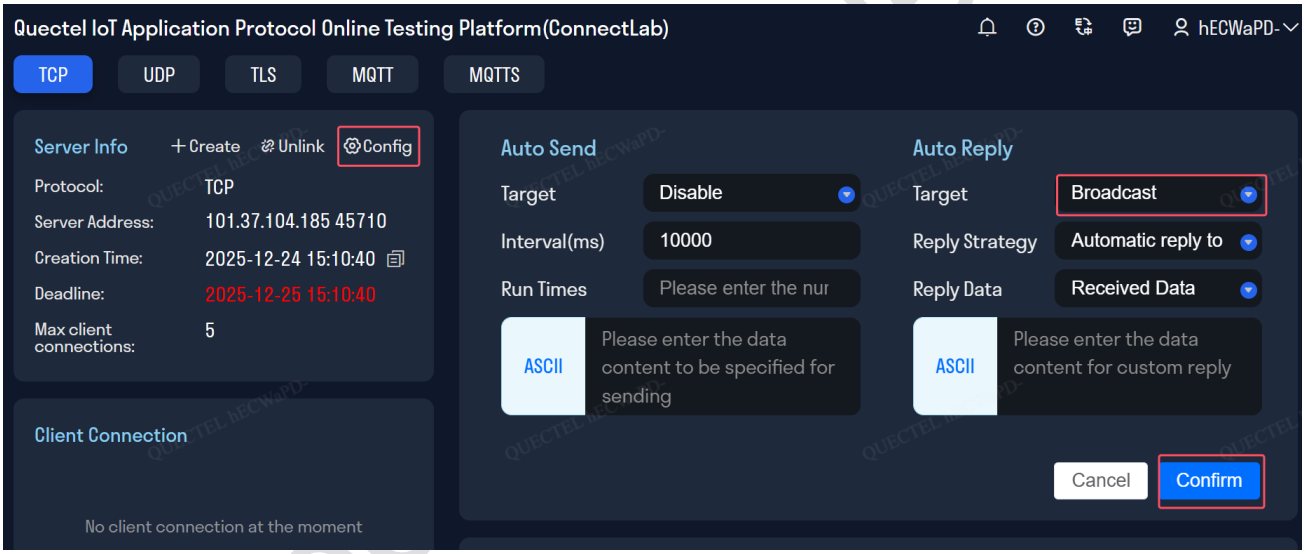

Figure 14: TCP/UDP Server Config

2.5.2. TCP-client Test

1. After turn on the board, it will occur "**Initialization done, do your own business**" after initialization.
2. Execute the command "**socket 0 101.37.104.185 45710 1 5000**". For the introduction of different parameters in the command, please refer to ***Chapter 3.2***, or execute the "socket help" to check. Please refer to ***Chapter 2.4.1*** to create TCP/UDP Server, then it will help you to do TCP/UDP test.

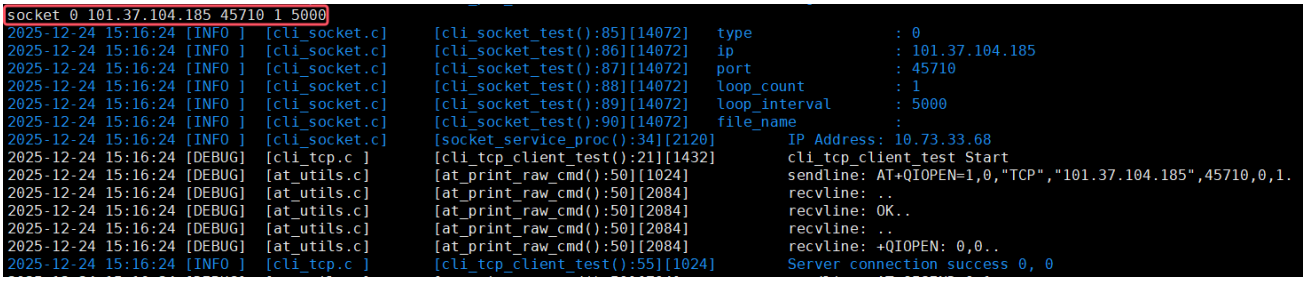

Figure 15: TCP Command

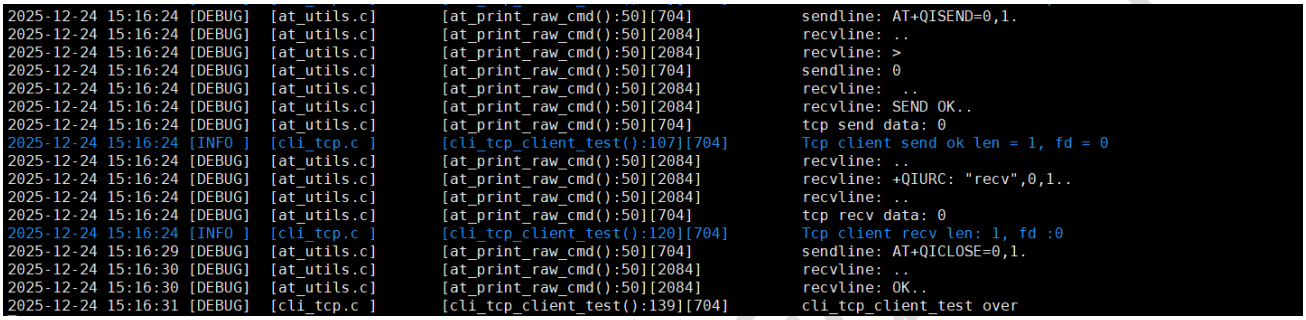

Figure 16: Successful TCP

2.5.3. HTTPS-POST Test

In order to test HTTPS POST (two-way authentication), please make sure that the certificates (**http_ca.pem, http_user.pem, http_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.

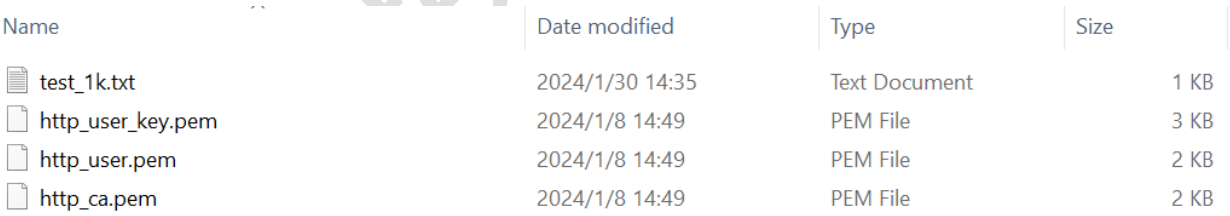

Figure 17: Test Files

1. Execute the command "**http 1 0 0 1 0 60 20 https://112.31.84.164:8303/upload.php 1 1 0 0 test_1k.txt 1 0 0x0035 2 1**". For the introduction of different parameters in the command, please refer to ***Chapter 3.4***, or execute the "http help" to check.
2. After post successfully, check the path and name of the posted file.
3. Open the HTTP server and check the posted file **a0c5b150-56ba-14d1-a8ef-54cd28f487fc**.

```
http 1 0 0 1 0 60 20 https://112.31.84.164:8303/upload.php 1 1 0 0 test 1k.txt 1 0 0x0035 2 1
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():152][11976]          contextid     : 1
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():153][11976]          requestheader : 0
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():154][11976]          responseheader: 0
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():155][11976]          contenttype   : 1
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():156][11976]          custom_header : 0
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():157][11976]          rsptime       : 60
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():158][11976]          wait_time     : 20
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():159][11976]          request_url   : https://112.31.84.164:8303/upload.php
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():160][11976]          method        : 1
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():161][11976]          request_mode  : 1
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():162][11976]          username      : 0
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():163][11976]          password      : 0
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():164][11976]          sslenble      : 1
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():176][11976]          sslctxid      : 0
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():177][11976]          ciphersuite   : 0x35
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():178][11976]          seclevel      : 2
2025-08-16 11:31:50 [INFO ]  [cli_http.c]        [cli_http_test():179][11976]          sslversion    : 1
2025-08-16 11:31:50 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][11976]           sendline: AT+QHTTPCFG="contextid",1..
2025-08-16 11:31:50 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]            recvline: ..
2025-08-16 11:31:50 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]            recvline: OK..
2025-08-16 11:31:50 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][11976]           sendline: AT+QHTTPCFG="requestheader",0..
2025-08-16 11:31:50 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]            recvline: ..
2025-08-16 11:31:50 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]            recvline: OK..
```

Figure 18: HTTPS-POST Command

```
2025-08-16 11:31:51 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][11976]          sendline: AT+QHTTPURL=37,60..
2025-08-16 11:31:51 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:31:51 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: CONNECT..
2025-08-16 11:31:51 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][11976]          sendline: https://112.31.84.164:8303/upload.php..
2025-08-16 11:31:51 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:31:51 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: OK..
2025-08-16 11:31:51 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][11976]          sendline: AT+QHTTPPOST=1024,60,60..
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: CONNECT..
2025-08-16 11:31:53 [INFO ]  [at_client.c]       [at_client_obj_send():546][11976]       sendline 1024 bytes data
2025-08-16 11:31:53 [INFO ]  [ql_http.c ]        [quectel_http_post():308][11976]        send size 1024
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: OK..
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline:
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: +QHTTPPOST: 0,200,87..
```

Figure 19: POST Successfully

```
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][11976]          sendline: AT+QHTTPREAD=20..
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1008]           urc_recv: CONNECT..
2025-08-16 11:31:53 [INFO ]  [cli_http.c]        [user_http_callback():122][1008]        open file: 0:post_info.txt
2025-08-16 11:31:53 [DEBUG]  [at_client.c]       [at_client_obj_recv():627][1008]        urc_recv 87 bytes data
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1008]           http recv data: text/plain{"size":1024,"path":"upload/2025/08/16/a0c5b150-56b
a-14d1-a8ef-54cd28f487fc"}
2025-08-16 11:31:53 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1008]           urc_recv: ..OK..
2025-08-16 11:31:53 [INFO ]  [cli_http.c]        [user_http_callback():136][1008]        recv total len = 87
2025-08-16 11:31:54 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1008]           recvline: ..
2025-08-16 11:31:54 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1008]           recvline: +QHTTPREAD: 0..
2025-08-16 11:31:54 [DEBUG]  [cli_http.c]        [cli_http_test():228][11976]    quectel_http_request 0
2025-08-16 11:31:54 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][11976]          sendline: AT+QHTTPSTOP..
2025-08-16 11:31:54 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1008]           recvline: ..
2025-08-16 11:31:54 [DEBUG]  [at_utils.c]        [at_print_raw_cmd():50][1008]           recvline: OK..
```

Figure 20: Path of the Posted File

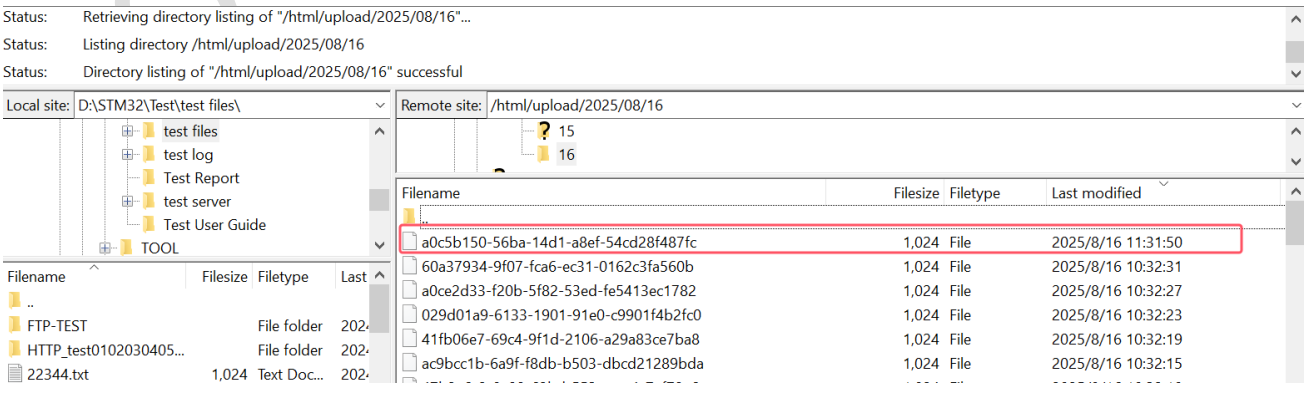

Figure 21: Post File Successfully

2.5.4. FTP Upload Test

In order to test FTP upload function, please make sure the upload file (**src_3k.txt**) are stored in the SD card.

Execute the command "**ftp 1 test test 1 1 100 "112.31.84.164" 8309 3 "/FTP-TEST" "0:src_3k.txt" "dst_3k.txt" 0**". For the introduction of different parameters in the command, please refer to ***Chapter 3.3***, or execute the "ftp help" to check.

```
ftp 1 test test 1 1 100 "112.31.84.164" 8309 3 "/FTP-TEST" "0:src_3k.txt" "dst_3k.txt" 0
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():109][14432]          contextid         : 1
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():110][14432]          username          : test
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():111][14432]          password          : test
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():112][14432]          filetype          : 1
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():113][14432]          transmode         : 1
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():114][14432]          rsptimeout        : 100
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():115][14432]          request_url       : 112.31.84.164
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():116][14432]          port              : 8309
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():117][14432]          ftp_type          : 3
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():118][14432]          directoryToSet    : /FTP-TEST
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():119][14432]          local_name        : 0:src_3k.txt
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():120][14432]          rem_name          : dst_3k.txt
2025-08-16 11:23:55 [INFO ]  [cli_ftp.c ]       [cli_ftp_test():121][14432]          sslenble          : 0
2025-08-16 11:23:55 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][14008]          sendline: AT+QFTPCFG="transmode",1..
2025-08-16 11:23:55 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: ..
2025-08-16 11:23:55 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: OK..
2025-08-16 11:23:55 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][14008]          sendline: AT+QFTPCFG="contextid",1..
2025-08-16 11:23:55 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: ..
2025-08-16 11:23:55 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: OK..
2025-08-16 11:23:55 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][14008]          sendline: AT+QFTPCFG="filetype",1..
```

Figure 22: FTP Command

The file is uploaded successfully.

```
2025-08-16 11:23:56 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][11976]          sendline: AT+QFTPCWD="/FTP-TEST"..
2025-08-16 11:23:56 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: ..
2025-08-16 11:23:56 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: OK..
2025-08-16 11:23:56 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: ..
2025-08-16 11:23:56 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: +QFTPCWD: 0,0..
2025-08-16 11:23:56 [INFO ]  [ql_ftp.c  ]       [quectel_ftp_upload():865][11976]       start upload 0:src_3k.txt
2025-08-16 11:23:56 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][11976]          sendline: AT+QFTPPUT="dst_3k.txt","COM:",0..
2025-08-16 11:23:56 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][2084]           recvline: ..
2025-08-16 11:23:56 [INFO ]  [ql_ftp.c  ]       [quectel_ftp_upload_cb():279][1280]     quectel_ftp_upload_cb: CONNECT

2025-08-16 11:23:56 [INFO ]  [at_client.c]      [at_client_obj_send():546][1224]        sendline 1024 bytes data
2025-08-16 11:23:56 [INFO ]  [at_client.c]      [at_client_obj_send():546][1224]        sendline 1024 bytes data
2025-08-16 11:23:56 [INFO ]  [at_client.c]      [at_client_obj_send():546][1224]        sendline 1024 bytes data
2025-08-16 11:23:58 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           sendline: +++
2025-08-16 11:23:58 [DEBUG]  [ql_ftp.c  ]       [quectel_ftp_upload_cb():299][1032]     close file
2025-08-16 11:23:59 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:23:59 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: OK..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: +QFTPPUT: 0,3072..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][11976]          sendline: AT+QFTPSTAT..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: OK..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: +QFTPSTAT: 0,1..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][11976]          sendline: AT+QFTPCLOSE..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: OK..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: ..
2025-08-16 11:24:00 [DEBUG]  [at_utils.c]       [at_print_raw_cmd():50][1032]           recvline: +QFTPCLOSE: 0,0..
```

Figure 23: Successful Uploading

By opening the FTP Server folder, it is vivid the src_3k.txt from SD card is uploaded to the FTP-TEST/dst_3k.txt.

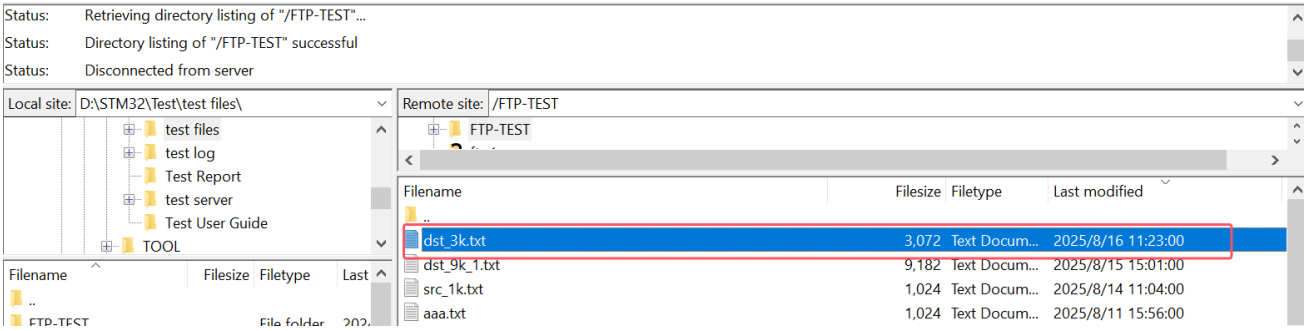

Figure 24: Location of the uploaded file

3 Functions Test Commands

3.1. Save Log in SD Card

Table 2: SD Function Definition

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

Commands Description

Command format:

debug mode <val>

<val> 0: debug

1: release

```
debug save <val>
           <val>  0: off
                  1: on
                  2: status
debug level <val>
           <val>  0: Verbose
                  1: Debug
                  2: Info
                  3: Warn
                  4: Error
debug test
```

3.2. TCP&UDP

Table 3: TCP&UDP Function Definition

Function	Commands	Log
TCP-Client	socket 0 101.37.104.185 45710 1 5000	tcp-client.txt
UDP-Client	socket 1 101.37.104.185 40492 1 5000	udp-client.txt
TCP-Server	socket 2 127.0.0.1 2020 100 1000 5	tcp-server.txt
UDP-Server	socket 3 127.0.0.1 2023 100 1000	udp-server.txt

Commands Description

1. Command format:

socket socket_type ip port count interval_ms max_connect_num

Parameter:

socket_type:

0: TCP

1: UDP

2: TCP SERVER

3: UDP SERVER

ip: ip address

port: port

count: Number of times the TCP/UDP client sends data

interval_ms: Time interval between TCP/UDP client data transmissions

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

2. When do TCP-Client and UDP-Client test, please confirm that the TCP/UDP server has enabled the function of automatically replying to the received data.

3.3. FTP(S)

Table 4: FTP(S) Function Definition

Function	Commands	Log
FTP-list	ftp 1 test GJbMlzZB65 1 1 100 "112.31.84.164" 8309 1 "/FTP-TEST" "0:ftplist.txt" "ftplist.txt" 0	ftp-list.txt
FTP-download	ftp 1 test GJbMlzZB65 1 1 100 "112.31.84.164" 8309 2 "/FTP-TEST" "0:dst_1k.txt" "src_1k.txt" 0	ftp-download.txt
FTP-upload	ftp 1 test GJbMlzZB65 1 1 100 "112.31.84.164" 8309 3 "/FTP-TEST" "0:src_3k.txt" "dst_3k.txt" 0	ftp-upload.txt
FTPS-list	ftp 1 test GJbMlzZB65 1 1 100 "112.31.84.164" 8311 1 "/FTP-TEST" "0:ftpslist.txt" "ftpslist.txt" 1 1 0 0xffff 1 4	ftps-list.txt
FTPS-download	ftp 1 test GJbMlzZB65 1 1 100 "112.31.84.164" 8311 2 "/FTP-TEST" "0:dst_1k.txt" "src_1k.txt" 1 1 0 0xffff 1 4	ftps-download.txt
FTPS-upload	ftp 1 test GJbMlzZB65 1 1 100 "112.31.84.164" 8311 3 "/FTP-TEST" "0:src_3k.txt" "dst_3k.txt" 1 1 0 0xffff 1 4	ftps-upload.txt

Commands Description

1. Command format:

ftp contextid username password filetype transmode rsptimeout hostname port ftp_type directoryToSet local_name rem_name sslenble sslctxid ciphersuite seclevel sslversion

Parameter:

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 upload
directoryToSet: The directory of the server
local_name: The file name in SD card
rem_name: The file name in 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

2. If test FTPS related function, please make sure that the certificates (**ftp_ca.pem, ftp_user.pem, ftp_user_key.pem**). Please refer to ***Chapter 4*** to get test files
3. If test FTP/FTPS upload function, please make sure the upload file (**src_3k.txt**) are stored in the SD card.

3.4. HTTP(S)

Table 5: HTTP(S) Function Definition

Function	Example commands	Log
HTTP-POST	http 1 0 0 1 0 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 20 http://112.31.84.164:8300/1024.txt 0 1 0 0 get_1k.txt 0	http-get.txt
HTTPS-POST	http 1 0 0 1 0 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 https://112.31.84.164:8303/1024.txt 0 1 0 0 get_1k.txt 1 0 0x0035 2 1	https-get.txt

Commands Description

1. **Command format:**

http contextid requestheader responseheader contenttype custom_header rsptime wait_time request_url method request_mode username password sd_card_path sslenble sslctxid ciphersuite seclevel sslversion

Parameter:

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

2. If test HTTPS related function, please make sure that the certificates (**http_ca.pem, http_user.pem, http_user_key.pem**). Please refer to ***Chapter 4*** to get test files
3. If test HTTP/HTTPS post function, please make sure the post file (**test_1k.txt**) are stored in the SD card.

3.5. MQTT(S)

Table 6: MQTT(S) Function Definition

Function	Example commands	Log
MQTT Open	mqtt 1 0 Test a1vvrmkn43t.iot-as-mqtt.cn-shanghai.aliyuncs.com 1883 a1vvrmkn43t NiFtKoHMcu6j0VIXtC6e 3115a9a768482d98a28d7390e7b9376b 0	mqtt-open.txt
MQTT Subscribe	mqtt 2 0 /a1vvrmkn43t/NiFtKoHMcu6j0VIXtC6e/user/get	mqtt-subscribe.txt
MQTT Publish	mqtt 3 0 /a1vvrmkn43t/NiFtKoHMcu6j0VIXtC6e/user/get 1234567890	mqtt-publish.txt
MQTT Disconnect	mqtt 4 0	mqtt-disconnect.txt

MQTTS Open	mqtt 1 1 quectel001 a9ohm2zbim3d5-ats.iot.us-east-1.amazonaws.com 8883 0 0 0 1 0XFFFF 2 4	mqtts-open.txt
MQTTS Subscribe	mqtt 2 0 aws/quectel001/data/report/message	mqtts-subcribe.txt
MQTTS Publish	mqtt 3 0 aws/quectel001/data/report/message 1234567890	mqtts-publish.txt
MQTTS Disconnect	mqtt 4 0	mqtts-disconnect.txt

Commands Description

1. open mqtt

Command format: mqtt test_type Server_type 0:ALP 1:other Client_ID server port ProductKey/username DeviceName/password DeviceSecret sslenble ciphersuite seclevel sslversion

example: mqtt 1 0 Test a1vvrmkn43t.iot-as-mqtt.cn-shanghai.aliyuncs.com 1883 a1vvrmkn43t NiFtKoHMcu6j0VIXtC6e 3115a9a768482d98a28d7390e7b9376b 0

2. subscribe topic

Command format: mqtt test_type mqtt_fd topic_name

example: mqtt 2 0 /a1vvrmkn43t/NiFtKoHMcu6j0VIXtC6e/user/tre1

3. public topic

Command format: mqtt test_type mqtt_fd topic_name mssagec

example: mqtt 3 0 /a1vvrmkn43t/p1U1UtVAPjZhkOEZnlUt/user/get 111

4. disconnect mqtt

Command format: mqtt test_type mqtt_fd

example: mqtt 4 0

Parameter:

test_type:

0: open mqtt

1: subscribe topic

2: public topic

3: disconnect mqtt

Server_type:

0: Alibaba Cloud

1: others

Client_ID: The client identifier string

server: The address of the server

port: The port of the server

ProductKey/username:
If it is Alibaba Server, need to configure Product key issued by Alibaba Cloud
Others, configure User name of the client

DeviceName/password:
If it is Alibaba Server, need to configure Device name issued by Alibaba Cloud.
Others, configure Password corresponding to the user name of the client

DeviceSecret:
If it is Alibaba Server, need to configure Device verification certificate issued by Alibaba Cloud.
Others, configure 0

sslenble: Whether ssl is enabled
0: Disable SSL
1: Enable SSL

ciphersuite: Numeric type in HEX format. SSL cipher suites
0x0035: TLS_RSA_WITH_AES_256_CBC_SHA
0x002F: TLS_RSA_WITH_AES_128_CBC_SHA
0x0005: TLS_RSA_WITH_RC4_128_SHA
0x0004: TLS_RSA_WITH_RC4_128_MD5
0x000A: TLS_RSA_WITH_3DES_EDE_CBC_SHA
0x003D: TLS_RSA_WITH_AES_256_CBC_SHA256
0xC002: TLS_ECDH_ECDSA_WITH_RC4_128_SHA
0xC003: TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
0xC004: TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
0xC005: TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
0xC007: TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
0xC008: TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
0xC009: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
0xC00A: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
0xC011: TLS_ECDHE_RSA_WITH_RC4_128_SHA
0xC012: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
0xC013: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
0xC014: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
0xC00C: TLS_ECDH_RSA_WITH_RC4_128_SHA
0xC00D: TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
0xC00E: TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
0xC00F: TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
0xC023: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

0xC024: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
0xC025: TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
0xC026: TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
0xC027: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
0xC028: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
0xC029: TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
0xC02A: TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
0xC02B: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
0xC02F: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
0xC0A8: TLS_PSK_WITH_AES_128_CCM_8
0x00AE: TLS_PSK_WITH_AES_128_CBC_SHA256
0xC0AE: TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
0xFFFF: ALL

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
2: TLS1.1
3: TLS1.2
4: ALL

If test MQTTS related function, please make sure that the certificates (**mqtt_ca.pem, mqtt_user.pem, mqtt_user_key.pem**). Please refer to ***Chapter 4*** to get test files

3.6. PSM

Table 7: PSM Function Definition

Function	Example commands	Log
PSM Setting	psm setting 3600 20	psm-setting.txt
PSM Disable	psm disable	psm-disable.txt
PSM Drive PON_TRIG	psm pon_trig 1 psm pon_trig 0	psm-pon_trig.txt
PSM Pull down POWERKEY	psm powerkey	psm-powerkey.txt

PSM stat	psm stat	psm-stat.txt

Commands Description

1. Command format:

psm disable
psm setting TAU/active time (ex setting 3600 60)
psm stat – show psm status
psm pon_trig value (0 or 1)
1: enter psm
0: exit psm
psm powerkey – exit psm

2. BG770AGL module Test PSM Method, which is also used for SONY LPWA modules. Firstly, make sure the network supports PSM. And on the TE-A board, please set PON_TRIG switch to GND, and connect UART_DEB to the PC using a MICRO-USB cable.

a) Enter into PSM method:
   i. Set **psm setting 3600 20**
   ii. Set **psm pon_trig 1**
   iii. After a few seconds (the RRC release and the expiration of T3324), URC "PSM POWER DOWN" will be outputted.

b) Exit PSM method (Only one method of awakening is listed here):
   i. Set **psm pon_trig 0**
   ii. Then URC "RDY" will be outputted, the module exits PSM.

3. BG95 module Test PSM Method, which is also used for LPWA Qualcomm modules. Firstly, make sure the network supports PSM.

a) Enter into PSM method:
   i. Set **psm setting 3600 20**
   ii. After a few seconds (the RRC release and the expiration of T3324), URC "PSM power down" will be outputted.

b) Exit PSM method (Only one method of awakening is listed here):
   i. Set **psm powerkey**
   ii. Then URC "RDY" will be outputted, the module exits PSM.

3.7. FILE

Table 8: FILE Function Definition

Function	Example commands	Log

FILE-query free	file 2 "UFS"	file-query-free.txt
FILE-open	file 3 “test.txt” 0	file-open.txt
FILE-write	file 4 1 10 abcd123!@#	file-write.txt
FILE-close	file 5 1	file-close.txt
FILE-query file	file 0 “*”	file-query-file.txt
FILE-delete	file 1 “ test.txt”	file-delete.txt

Commands Description

Command format:

0. query files
example: file 0 "*"

1. del files
example: file 1 "123.txt"

2. query free
example: file 2 "UFS"

3. open files
example: file 3 “text” 0
example: file 3 filename mode
mode Integer type. The open mode of the file
0 If the file does not exist, it is created. If the file exists, it is opened directly. In any case, the file can be read and written.
1 If the file does not exist, it is created. If the file exists, it is overwritten. In any case, the file can be read and written.
2 If the file exists, it is opened directly and is read only. If the file does not exist, an error is returned.
3 If the file does not exist, it is created. If the file exists, write data to the file. In any case, the file can be read and written.

4. write files
example: file 4 1 5 12345
example: file 4 filehandle length DATA
<filehandle> Integer type. The handle of the file to be operated.
<length> Integer type. The length of the file to be written.
<DATA> WIRTE DATA

5. close files

example: file 5 1
example: file 5 filehandle

6. read files
example: file 6 1 5
example: file 6 filehandle length

3.8. FOTA

Table 9: FOTA Function Definition

Function	Example commands	Log
FOTA HTTP	fota 0 0 1 http://112.31.84.164:8300/QSTM32/FOTA/Quectel_UFP_STM32F413RGT6_A03.bin 0 FotaFile.bin 0	fota-http.txt
FOTA HTTPS	fota 0 0 1 https://112.31.84.164:8301/QSTM32/FOTA/Quectel_UFP_STM32F413RGT6_A03.bin 0 FotaFile.bin 1 1 0xFFFF 2 4	fota-https.txt
FOTA FTP	fota 1 0 1 112.31.84.164 8309 test GJbMlzZB65 /QSTM32/FOTA Quectel_UFP_STM32F413RGT6_A03_FOTA.bin 0 FotaFile.bin 0	fota-ftp.txt
FOTA FTPS	fota 1 1 1 112.31.84.164 8311 test GJbMlzZB65 /QSTM32/FOTA Quectel_UFP_STM32F413RGT6_A03_FOTA.bin 0 FotaFile.bin 1 1 1 0xFFFF 1 4	fota-ftps.txt

Commands Description

Command format:
fota by http
fota download_type async contextid url location save_path sslenble sslctxid ciphersuite seclevel sslversion
download_type : 0-http(s)
async : 0-false, 1-true
contextid : PDP context ID, range: 1-16
url : HTTP(S) server URL
location : 0-scard, 1-flash, now only support sdcard

save_path : FotaFile.bin(File path on the storage)
sslenble : Whether ssl is enabled, 0-disabled, 1-enabled
sslctxid : SSL context ID used for https, 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
sslversion : SSL Version
0: SSL3.0
1: TLS1.0
2: TLS1.1
3: TLS1.2
4: ALL

fota by ftp

fota download_type async contextid address port username password wrok_dir rem_name location save_path sslenble ssltype sslctxid ciphersuite seclevel sslversion

download_type : 1-ftp(s)
async : 0-false, 1-true
contextid : PDP context ID, range: 1-16
address : FTP(S) server address
port : FTP(S) server port
username : Username for logging in to the ftp(s) server
password : Password for logging in to the Ftp(S) server
wrok_dir : Working directory on the FTP(S) server
rem_name : The file name of the server
location : 0-scard, 1-flash, now only support sdcard
save_path : FotaFile.bin(File path on the storage)
sslenble : Whether ssl is enabled, 0-disabled, 1-enabled
ssltype : Module used as FTP client or FTPS client
0 FTP clients
1 FTPS implicit encryption
2 FTPS explicit encryption
sslctxid : SSL context ID used for HTTPS, 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
sslversion : SSL Version
0: SSL3.0
1: TLS1.0
2: TLS1.1

3: TLS1.2
4: ALL

3.9. Function Help

Execute the command “**Function Help**” and you can get the information of the command. For example, following shows **socket help**.

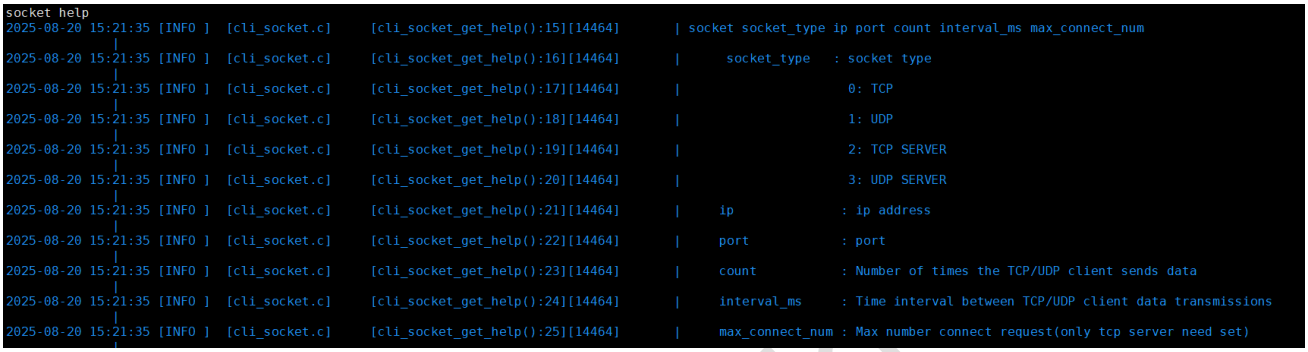

Figure 25: Reference Circuit of the Help

4 Test Files

When testing FTPS/HTTPS/MQTTS, we need to put certifications into the SD card. Additionally, the http post file **test_1k.txt** and ftp upload file **src_3k.txt** shall also be placed on the SD card in order to test HTTP.
Please check test files to get these files.
Note: The test server and related certificates provided above are only for testing purposes. If for commercial use, please create your own server.

Table 10: Certificates for the Test

Functions	Certificate
ftps	ftp_ca.pem
https	http_ca.pem http_user.pem

	http_user_key.pem
mqtts	mqtt_ca.pem mqtt_user.pem mqtt_user_key.pem

5 Appendix References

Table 11: Related Documents

SN	Document Name
[1]	STM32 LQFP64 EVK User Guide
[2]	Quectel_QSTM32_SDK_Quick_Start_Guide