

# EG800Q&EG91xQ Series

## FILE Application Note

**LTE Standard Module Series**

Version: 1.2

Date: 2024-05-22

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](mailto: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](mailto: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.

**Copyright © Quectel Wireless Solutions Co., Ltd. 2024. All rights reserved.**

# About the Document

## Revision History

Version	Date	Author	Description
-	2022-10-10	Egan LI	Creation of the document
1.0	2022-12-01	Egan LI	First official release
1.1	2023-09-05	Tomas QIN	Updated the applicable modules: <ul style="list-style-type: none"><li>● Added EG916Q-GL.</li><li>● Updated EG800Q-EU to EG800Q series.</li></ul>
1.2	2024-05-22	Tomas QIN	Updated EG915Q-NA to EG915Q series.

## Contents

About the Document.....	3
Contents .....	4
Table Index .....	5
<b>1 Introduction .....</b>	<b>6</b>
1.1. Using FILE AT Commands .....	6
1.2. Description of Data Mode.....	7
<b>2 Description of FILE AT Commands.....</b>	<b>8</b>
2.1. AT Command Description .....	8
2.1.1. Definitions.....	8
2.1.2. AT Command Syntax .....	8
2.2. Declaration of AT Command Examples .....	9
2.3. AT Command Description .....	9
2.3.1. AT+QFLDS Get the Space Information of Storage Medium .....	9
2.3.2. AT+QFLST List File Information on Storage Medium .....	10
2.3.3. AT+QFDEL Delete File(s) on Storage Medium .....	11
2.3.4. AT+QFUPL Upload File to Storage Medium .....	12
2.3.5. AT+QFDWL Download File from Storage Medium .....	14
2.3.6. AT+QFOPEN Open a File .....	15
2.3.7. AT+QFREAD Read a File.....	16
2.3.8. AT+QFWRITE Write to a File .....	17
2.3.9. AT+QFSEEK Set File Pointer to Specified Position.....	18
2.3.10. AT+QFPOSITION Get Offset of File Pointer.....	19
2.3.11. AT+QFCLOSE Close a File.....	20
<b>3 Examples .....</b>	<b>21</b>
3.1. Upload and Download Files .....	21
3.1.1. Upload a File .....	21
3.1.1.1. Non-ACK Mode.....	21
3.1.1.2. ACK Mode .....	21
3.1.2. Download a File .....	22
3.2. Write to and Read Files .....	22
3.2.1. Write to and Read a UFS File .....	22
<b>4 Summary of Error Codes .....</b>	<b>23</b>
<b>5 Appendix References .....</b>	<b>25</b>

## Table Index

Table 1: Types of AT Commands .....	8
Table 2: Summary of Error Codes.....	23
Table 3: Terms and Abbreviations .....	25

# 1 Introduction

Quectel EG800Q series and EG91xQ family (EG915Q series and EG916Q-GL) modules support AT commands for working with files on different physical storage mediums. This document is a reference guide to these commands.

Quectel EG800Q series and EG91xQ family modules support only one storage medium – UFS (User File Storage directory) currently. It is a special directory on the flash file system.

## 1.1. Using FILE AT Commands

Follow the steps below to create, read and write to a file in the storage:

1. Upload a file to the storage with **AT+QFUPL**. If necessary, output/download it through the serial interface with **AT+QFDWL**.
2. Open the file with **AT+QFOPEN**. When the file is opened, you can write to it or read from it any time and from any location until the file is closed with **AT+QFCLOSE**.
  - When opening a file with **AT+QFOPEN**, you can set the file into overwrite mode, read-only mode or other modes with **<mode>** (For more information about **<mode>**, see **Chapter 2.3.6**). After opening the file, a **<filehandle>** is assigned to it so that various file operations can be carried out.
  - After opening the file, you can write data to the file with **AT+QFWRITE** and read the data from the current file position with **AT+QFREAD**.
  - Set the file position with **AT+QFSEEK** or query the current file position with **AT+QFPOSITION**.
  - Close the file with **AT+QFCLOSE**, after which the **<filehandle>** will become invalid.

Commands for managing files on storage medium:

1. **AT+QFLDS**: Get storage space information.
2. **AT+QFLST**: List the file information on the storage medium.
3. **AT+QFDEL**: Delete the file(s) on the storage medium.

### NOTE

The file handle obtained after executing AT+QFOPEN must be closed with AT+QFCLOSE after the operation is completed, otherwise the file handle will be leaked.

## 1.2. Description of Data Mode

The COM port of EG800Q series and EG91xQ family modules has two working modes: AT command mode and data mode. In AT command mode, the data inputted via COM port are treated as AT commands; whereas in data mode, they are treated as data.

- **Enter Data Mode**

Once **AT+QFUP**, **AT+QFDWL**, **AT+QFREAD** or **AT+QFWRITE** is executed, the module returns **CONNECT** and the COM port enters data mode. Or you can enter data mode again by executing **ATO**.

- **Exit Data Mode**

If you input **+++** or pull up the DTR pin (**AT&D1** must be executed first) to make the port exit data mode, the execution of these commands will be interrupted before the response is returned. In such a case, the COM port cannot re-enter data mode if you execute **ATO**.

To prevent **+++** from being misinterpreted as data, the following sequence should be followed:

- 1 Do not input any data for at least 1 s after inputting **+++**.
- 2 Input **+++** within 1 s and any other data cannot be inputted.
- 3 Input **+++** and wait until **OK** is returned. Once **OK** is returned, the COM port exits the data mode.



# 2 Description of FILE AT Commands

## 2.1. AT Command Description

### 2.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on the command line.
- **[...]** Optional parameter of a command or an optional part of TA information response.  
Square brackets do not appear on the command line. When an optional parameter is not given in a command, the new value equals its previous value or the default settings, unless otherwise specified.
- **Underline** Default setting of a parameter.

### 2.1.2. AT Command Syntax

All command lines must start with **AT** or **at** and end with **<CR>**. Information responses and result codes always start and end with a carriage return character and a line feed character: **<CR><LF><response><CR><LF>**. In tables presenting commands and responses throughout this document, only the commands and responses are presented, and **<CR>** and **<LF>** are deliberately omitted.

Table 1: Types of AT Commands

Command Type	Syntax	Description
Test Command	<b>AT+&lt;cmd&gt;=?</b>	Test the existence of the corresponding command and return information about the type, value, or range of its parameter.
Read Command	<b>AT+&lt;cmd&gt;?</b>	Check the current parameter value of the corresponding command.
Write Command	<b>AT+&lt;cmd&gt;=&lt;p1&gt;[,&lt;p2&gt;[,&lt;p3&gt;[...]]]</b>	Set user-definable parameter value.
Execution Command	<b>AT+&lt;cmd&gt;</b>	Return a specific information parameter or perform a specific action.

## 2.2. Declaration of AT Command Examples

The AT command examples in this document are provided to help you learn about the use of the AT commands introduced herein. The examples, however, should not be taken as Quectel's recommendations or suggestions about how to design a program flow or what status to set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there is a correlation among these examples, or that they should be executed in a given sequence.

## 2.3. AT Command Description

### 2.3.1. AT+QFLDS Get the Space Information of Storage Medium

This command gets the space information of the specified storage medium.

AT+QFLDS Get the Space Information of Storage Medium	
Test Command <b>AT+QFLDS=?</b>	Response <b>OK</b>
Write Command <b>AT+QFLDS=&lt;name_pattern&gt;</b>	Response <b>+QFLDS: &lt;free_size&gt;,&lt;total_size&gt;</b>  <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Execution Command <b>AT+QFLDS</b>	Response Return UFS space information: <b>+QFLDS: &lt;UFS_file_size&gt;,&lt;UFS_file_number&gt;</b>  <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

## Parameter

<name_pattern>	String type. Storage medium type. "UFS" UFS
<free_size>	Integer type. Free space size of <name_pattern>. Unit: byte.
<total_size>	Integer type. Total space size of <name_pattern>. Unit: byte.
<UFS_file_size>	Integer type. Size of all files in UFS. Unit: byte.
<UFS_file_number>	Integer type. Number of files in UFS.
<err>	Integer type. Error code. See <b>Chapter 4</b> for details.

## Example

```

AT+QFLDS="UFS" //Query the space information of UFS.
+QFLDS: 578847,917503

OK

```

### 2.3.2. AT+QFLST List File Information on Storage Medium

This command lists the information of a single file or all files on a specified storage medium.

AT+QFLST List File Information on Storage Medium	
Test Command <b>AT+QFLST=?</b>	Response <b>OK</b>
Write Command <b>AT+QFLST=&lt;filename&gt;</b>	Response <b>+QFLST: &lt;filename&gt;,&lt;file_size&gt;</b> <b>[+QFLST: &lt;filename&gt;,&lt;file_size&gt;</b> <b>[...]]</b>  <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Execution Command <b>AT+QFLST</b>	Response Return the space information of UFS files: <b>+QFLST: &lt;filename&gt;,&lt;file_size&gt;</b> <b>[+QFLST: &lt;filename&gt;,&lt;file_size&gt;</b> <b>[...]]</b>  <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>

Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

## Parameter

<b>&lt;filename&gt;</b>	String type. File to be listed. Maximum length: 80 bytes. "*" List all files in UFS "UFS:<file>" List the specified file <file> in UFS "<file>" List the specified file <file> in UFS
<b>&lt;file&gt;</b>	String type. File name.
<b>&lt;file_size&gt;</b>	Integer type. File size of specified file <filename> in UFS. Unit: byte.
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.

## Example

```

AT+QFLST="" //List all files in UFS.
+QFLST: "UFS:1k.txt",1024
+QFLST: "UFS:2k.txt",2048
+QFLST: "UFS:3k.txt",3072
OK
  
```

### NOTE

**AT+QFLST** queries the actual size of the file currently stored on Flash. Use **AT+QFWRITE** (see **Chapter 2.3.8**) to write data. If **AT+QFLST** cannot query file size, execute **AT+QFCLOSE** (see **Chapter 2.3.11**) to close the file and then query the file size.

## 2.3.3. AT+QFDEL Delete File(s) on Storage Medium

This command deletes a single file or all files from a specified storage medium.

### AT+QFDEL Delete File(s) on Storage Medium

Test Command <b>AT+QFDEL=?</b>	Response <b>+QFDEL: &lt;filename&gt;</b>  <b>OK</b>
Write Command <b>AT+QFDEL=&lt;filename&gt;</b>	Response <b>OK</b>  If there is an error:

	<b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

## Parameter

<b>&lt;filename&gt;</b>	String type. Name of the file to be deleted. Maximum length: 80 bytes. "*" Delete all files in UFS (do not delete the directory) "UFS:<file>" Delete the specified file <file> in UFS "<file>" Delete the specified file <file> in UFS
<b>&lt;file&gt;</b>	String type. File name.
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.

## Example

```

AT+QFDEL="" //Delete all files in UFS (do not delete the directory).
OK
AT+QFDEL="UFS:1.txt" //Delete the 1.txt file in UFS.
OK

```

## 2.3.4. AT+QFUPL Upload File to Storage Medium

This command uploads a file to a storage medium. If any file on the storage has the same name as the file to be uploaded, an error will be reported.

AT+QFUPL Upload File to Storage Medium	
Test Command <b>AT+QFUPL=?</b>	Response <b>+QFUPL: &lt;filename&gt;[, (1-&lt;free_size&gt;)] [, (range of supported &lt;timeout&gt;s)] [, (list of supported &lt;ackmode&gt;s)]</b>  <b>OK</b>
Write Command <b>AT+QFUPL=&lt;filename&gt;[, &lt;file_size&gt;[, &lt;timeout&gt;[, &lt;ackmode&gt;]]]</b>	Response <b>CONNECT</b> TA switches to data mode (transparent transmission mode), so the binary data of the file can be inputted. When the total size of the inputted data reaches <file_size> or no data is inputted when <timeout> is reached, TA will go back to AT command mode and return the following code: <b>+QFUPL: &lt;upload_size&gt;,&lt;checksum&gt;</b>  <b>OK</b>

	If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	5 s
Characteristics	The command takes effect immediately. The configurations are not saved.

## Parameter

<b>&lt;filename&gt;</b>	String type. Name of the file to be uploaded. Maximum length: 80 bytes. "UFS:<file>"      Name of the file <file> to be uploaded to UFS "<file>"            Name of the file <file> to be uploaded to UFS
<b>&lt;file&gt;</b>	String type. File name.
<b>&lt;free_size&gt;</b>	Integer type. Free space size of <name_pattern>. See <b>AT+QFLDS</b> for more details on <name_pattern>.
<b>&lt;timeout&gt;</b>	Integer type. Waiting time for inputting data to USB/UART. Range: 1–65535. Default value: 5. Unit: second.
<b>&lt;ackmode&gt;</b>	Integer type. Determines whether to use ACK mode. 0      Turn off ACK mode 1      Turn on ACK mode
<b>&lt;file_size&gt;</b>	Integer type. File size expected to be uploaded. Default value: 10240. Unit: byte. Maximum length is not greater than <free_size>.
<b>&lt;upload_size&gt;</b>	Integer type. Actual size of uploaded data. Unit: byte.
<b>&lt;checksum&gt;</b>	Integer type. Checksum of uploaded data.
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.

### NOTE

- It is strongly recommended to use DOS 8.3 file name format for **<filename>**.
- <checksum>** is a 16-bit checksum based on bitwise XOR.  
If the number of characters is odd, set the last character as the high 8 bit, and the low 8 bit as 0, and then use an XOR operator to calculate the checksum. After switching to command mode, the previously uploaded data will be preserved in the file.
- When executing the command, the data must be entered after **CONNECT** is returned.
- ACK mode is a safeguard against data loss when uploading large files, if hardware flow control does not work. ACK mode works as follows:
  - Execute **AT+QFUPL=<filename>,<file\_size>,<timeout>,1** to enable ACK mode.
  - Module outputs **CONNECT**.
  - MCU sends 1 KB data, to which the module responds with an **A**.
  - MCU receives the **A** and then sends the next 1 KB data.
  - Repeat step 3) and 4) until the transfer is completed.
For an example of ACK mode use, see **Chapter 3.1.1.2**.

### 2.3.5. AT+QFDWL Download File from Storage Medium

This command downloads a specified file from the storage medium.

AT+QFDWL Download File from Storage Medium	
Test Command <b>AT+QFDWL=?</b>	Response <b>+QFDWL: &lt;filename&gt;</b>  <b>OK</b>
Write Command <b>AT+QFDWL=&lt;filename&gt;</b>	Response <b>CONNECT</b> TA switches to data mode, so the binary data of the file can be outputted. After the file is downloaded, TA will go back to AT command mode and return the following codes: <b>+QFDWL: &lt;download_size&gt;,&lt;checksum&gt;</b>  <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	5 s
Characteristics	The command takes effect immediately. The configurations are not saved.

#### Parameter

<b>&lt;filename&gt;</b>	String type. Name of the file to be downloaded. Maximum length: 80 bytes. "UFS:<file>"      Name of the UFS file <file> to be downloaded "<file>"              Name of the UFS file <file>to be downloaded
<b>&lt;file&gt;</b>	String type. File name.
<b>&lt;download_size&gt;</b>	Integer type. Size of downloaded data. Unit: byte.
<b>&lt;checksum&gt;</b>	Integer type. Checksum of downloaded data.
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.

#### NOTE

**<checksum>** is a 16-bit checksum based on bitwise XOR.

### 2.3.6. AT+QFOPEN Open a File

This command opens a file and gets the file handle to be used in commands such as **AT+QFREAD**, **AT+QFWRITE**, **AT+QFSEEK**, **AT+QFPOSITION** and **AT+QFCLOSE**.

AT+QFOPEN Open a File	
Test Command <b>AT+QFOPEN=?</b>	Response <b>+QFOPEN: &lt;filename&gt;[(range of supported &lt;mode&gt;s)]</b>  <b>OK</b>
Read Command <b>AT+QFOPEN?</b>	Response <b>+QFOPEN: &lt;filename&gt;,&lt;filehandle&gt;,&lt;mode&gt;</b> <b>[+QFOPEN: &lt;filename&gt;,&lt;filehandle&gt;,&lt;mode&gt;</b> <b>[...]]</b>  <b>OK</b>
Write Command <b>AT+QFOPEN=&lt;filename&gt;[,&lt;mode&gt;]</b>	Response <b>+QFOPEN: &lt;filehandle&gt;</b>  <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

#### Parameter

<b>&lt;filename&gt;</b>	String type. Name of the file to be opened. Maximum length: 80 bytes. "UFS:<file>"      Name of the UFS file <file> to be opened "<file>"              Name of the UFS file <file> to be opened
<b>&lt;file&gt;</b>	String type. File name.
<b>&lt;mode&gt;</b>	Integer type. File opening mode. <u>0</u> If the file does not exist, it is created. If the file exists, it is opened directly. In both cases, the file can be read and written to. 1      If the file does not exist, it is created. If the file exists, it is overwritten and cleared. In both cases, the file can be read and written to. 2      If the file exists, it is opened directly as a read-only file. Otherwise, an error is returned.
<b>&lt;filehandle&gt;</b>	Integer type. File handle to be used. Data type: 4 bytes.
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.



### 2.3.7. AT+QFREAD Read a File

This command reads the data of a file specified by the file handle. The data start from the current position of the file pointer that belongs to the file handle.

AT+QFREAD Read a File	
Test Command <b>AT+QFREAD=?</b>	Response <b>+QFREAD: &lt;filehandle&gt;[,&lt;length&gt;]</b>  <b>OK</b>
Write Command <b>AT+QFREAD=&lt;filehandle&gt;[,&lt;length&gt;]</b>	Response <b>CONNECT &lt;read_length&gt;</b> TA switches to data mode, so the data being read will be outputted. When the total size of the data reaches <b>&lt;length&gt;</b> , TA will go back to AT command mode, display the result and then return the following code: <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	5 s
Characteristics	The command takes effect immediately. The configurations are not saved.

#### Parameter

<b>&lt;filehandle&gt;</b>	Integer type. File handle to be read.
<b>&lt;length&gt;</b>	Integer type. Length of the file to be read. If this parameter is not inputted, the default length is 10 KB. If this parameter is inputted, the actual length of the file is read. Maximum value of this parameter is determined by <b>&lt;free_size&gt;</b> of <b>AT+QFUPL</b> . Default value: 10240. Unit: byte.
<b>&lt;read_length&gt;</b>	Integer type. Actual read length. Unit: byte.
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.

### 2.3.8. AT+QFWRITE Write to a File

This command writes data into a file. The data starts from the current position of the file pointer that belongs to the file handle.

AT+QFWRITE Write to a File	
Test Command <b>AT+QFWRITE=?</b>	Response <b>+QFWRITE: &lt;filehandle&gt;[,&lt;length&gt;[,range of supported &lt;timeout&gt;s]]</b>  <b>OK</b>
Write Command <b>AT+QFWRITE=&lt;filehandle&gt;[,&lt;length&gt;[,&lt;timeout&gt;]]</b>	Response <b>CONNECT</b> TA switches to data mode, so the hexadecimal data of the file can be written to. When the total size of the written data reaches <b>&lt;length&gt;</b> or the written time reaches <b>&lt;timeout&gt;</b> , TA will go back to AT command mode and return the following code: <b>+QFWRITE: &lt;written_length&gt;,&lt;total_length&gt;</b>  <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	5 s
Characteristics	The command takes effect immediately. The configurations are not saved.

#### Parameter

<b>&lt;filehandle&gt;</b>	Integer type. File handle to be written to.
<b>&lt;length&gt;</b>	Integer type. Length of the file to be written to. If this parameter is not inputted, the default length is 10 KB. If this parameter is inputted, the actual length of the file is written to. Maximum value of this parameter is determined by <b>&lt;free_size&gt;</b> of <b>AT+QFUPL</b> . Default value: 10240. Unit: byte.
<b>&lt;timeout&gt;</b>	Integer type. Waiting time for inputting data to USB/UART. Range:1–65535. Default value: 5. Unit: second.
<b>&lt;written_length&gt;</b>	Integer type. Actual written length. Unit: byte.
<b>&lt;total_length&gt;</b>	Integer type. Total file length. Unit: byte.
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.

### 2.3.9. AT+QFSEEK Set File Pointer to Specified Position

This command sets a file pointer to the specified position. This will decide the starting position of commands, such as **AT+QFREAD**, **AT+QFWRITE** and **AT+QFPOSITION**.

<b>AT+QFSEEK Set a File Pointer to Specified Position</b>	
Test Command <b>AT+QFSEEK=?</b>	Response <b>+QFSEEK: &lt;filehandle&gt;,&lt;offset&gt;[,range of supported &lt;position&gt;s]</b>  <b>OK</b>
Write Command <b>AT+QFSEEK=&lt;filehandle&gt;,&lt;offset&gt;[,&lt;position&gt;]</b>	Response <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

#### Parameter

<b>&lt;filehandle&gt;</b>	Integer type. File handle for which a pointer is set.
<b>&lt;offset&gt;</b>	Integer type. Number of bytes of the file pointer movement.
<b>&lt;position&gt;</b>	Integer type. Pointer movement mode. <ul style="list-style-type: none"> <li><u>0</u> Move forward from the beginning of the file</li> <li>1 Move forward from the current position of the pointer</li> <li>2 Move backward from the end of the file</li> </ul>
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.

#### NOTE

If the set final position of the pointer exceeds the file size, executing this command will return **ERROR**.

### 2.3.10. AT+QFPOSITION Get Offset of File Pointer

This command gets the offset of a file pointer from the beginning of the file.

AT+QFPOSITION Get the Offset of a File Pointer	
Test Command <b>AT+QFPOSITION=?</b>	Response <b>+QFPOSITION: &lt;filehandle&gt;</b>  <b>OK</b>
Write Command <b>AT+QFPOSITION=&lt;filehandle&gt;</b>	Response <b>+QFPOSITION: &lt;offset&gt;</b>  <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

#### Parameter

<b>&lt;filehandle&gt;</b>	Integer type. File handle for which the offset of file pointer is to be gotten.
<b>&lt;offset&gt;</b>	Integer type. Offset from the beginning of the file.
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.

### 2.3.11. AT+QFCLOSE Close a File

This command closes a file and ends all file operations. After that, the file handle is released and should not be used again, unless the file is re-opened with **AT+QFOPEN**.

AT+QFCLOSE Close a File	
Test Command <b>AT+QFCLOSE=?</b>	Response <b>+QFCLOSE: &lt;filehandle&gt;</b>  <b>OK</b>
Write Command <b>AT+QFCLOSE=&lt;filehandle&gt;</b>	Response <b>OK</b>  If there is an error: <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

#### Parameter

<b>&lt;filehandle&gt;</b>	Integer type. File handle to be closed.
<b>&lt;err&gt;</b>	Integer type. Error code. See <b>Chapter 4</b> for details.

# 3 Examples

## 3.1. Upload and Download Files

### 3.1.1. Upload a File

#### 3.1.1.1. Non-ACK Mode

<b>AT+QFUPL="test1.txt",10</b>	//Upload the text file <i>test1.txt</i> to UFS.
<b>CONNECT</b>	
<b>&lt;Input file bin data&gt;</b>	//Input binary data of the text file in data mode.
<b>+QFUPL: 10,3938</b>	
<b>OK</b>	

#### 3.1.1.2. ACK Mode

The ACK mode can make the data transmission more reliable. When transmitting a large file without hardware flow control, the ACK mode is recommended to prevent data loss. For more details on ACK mode, see **AT+QFUPL**.

<b>AT+QFUPL="test.txt",3000,5,1</b>	//Upload the text file <i>test.txt</i> to UFS.
<b>CONNECT</b>	
<b>&lt;input file bin data of 1024bytes&gt;</b>	//Input 1024 bytes of binary data of the text file in data mode.
<b>A</b>	//After receiving 1024 bytes of binary data of the text file, the module returns an <b>A</b> .
<b>&lt;input file bin data of 1024bytes&gt;</b>	//Input next 1024 bytes of binary data of the text file in data mode.
<b>A</b>	
<b>&lt;input the rest file bin data&gt;</b>	//Input the rest of binary data of the text file in data mode.
<b>+QFUPL: 3000,B34A</b>	
<b>OK</b>	

### 3.1.2. Download a File

```

AT+QFDWL="test.txt"           //Download the text file test.txt from UFS.
CONNECT
<Output Data>                 //Output binary data of the text file in data mode.
+QFDWL: 10,613e               //Return size and checksum value of the downloaded data.

OK

```

## 3.2. Write to and Read Files

### 3.2.1. Write to and Read a UFS File

```

AT+QFOPEN="test.txt",1       //Open the file to get the file handle.
+QFOPEN: 1

OK
AT+QFWRITE=1,10              //Write 10 bytes to the file.
CONNECT
<Write Data>                 //Write hexadecimal data to the file in data mode.
+QFWRITE: 10,10              //Return the actual written bytes and the size of the file.

OK
AT+QFSEEK=1,0,0              //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=1,10               //Read data.
CONNECT 10
<Read Data>                  //Output data to be read in data mode.

OK
AT+QFCLOSE=1                 //Close the file.
OK

```

# 4 Summary of Error Codes

<err> indicates the error codes related to mobile equipment. See the table below for details. The error codes indicated below are related to the file operation of the modules.

**Table 2: Summary of Error Codes**

<err>	Meaning
400	Invalid input value
401	Larger than the size of the file
402	Zero-byte read
403	Drive full
405	File not found
406	Invalid file name
407	File already exists
409	Fail to write to the file
410	Fail to open the file
411	Fail to read the file
413	Reach the max. number of files allowed to be opened
414	Read-only file
416	Invalid file descriptor
417	Failed to list the file
418	Failed to delete the file
419	Failed to get disk info
420	No space



---

421	Time-out
423	File too large
425	Invalid parameter
426	File already opened

---

# 5 Appendix References

**Table 3: Terms and Abbreviations**

Abbreviation	Description
ACK	Acknowledgement
COM	Communication Port
DOS	Disk Operating System
DTR	Data Terminal Ready
MCU	Microprogrammed Control Unit
ME	Mobile Equipment
RAM	Random Access Memory
TA	Terminal Adapter
UART	Universal Asynchronous Receiver-Transmitter
UFS	User File Storage
USB	Universal Serial Bus
XOR	Exclusive OR