# RG50xQ&RM5xxQ Series
# FILE Application Note

**5G Module Series**

Version: 1.1

Date: 2023-02-20

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 |
|---|---|---|---|
| - | 2022-02-10 | Monan TIAN | Creation of the document |
| 1.0 | 2022-06-13 | Monan TIAN | First official release |
| 1.1 | 2023-02-20 | Monan TIAN | 1. Optimized the description of data modes (Chapter 1.3).<br>2. Rephrased the description of the parameter <filename> (Chapter 2.3.2–2.3.6).<br>3. Added the description of the Linux example of commands AT+QFUPL, AT+QFDWL, AT+QFREAD, and AT+QFWRITE (Chapter 2.3.4, 2.3.5, 2.3.7, and 2.3.8). |

# Contents

# Table Index

# 1 Introduction

Quectel RG50xQ family and RM5xxQ family support AT commands to control files on different physical storage mediums. This document is a reference guide to these commands.

The modules support the following storage mediums:

- **UFS:** User File Storage directory. It is a special directory on the flash file system.
- **SD:** SD card directory (only supported by the RG50xQ family).

**NOTES**

The **<filename>** contains the storage location. When it begins with "UFS:" or with no prefix, it means that the file is stored in the UFS. When it begins with "SD:", it means that the file is stored in the SD card.

## 1.1. Applicable Modules

**Table 1: Applicable Modules**

| Module Family | Module |
|---|---|
| RG50xQ | RG500Q Series |
| | RG501Q-EU |
| | RG502Q Series |
| RM5xxQ | RM500Q Series |
| | RM502Q-AE |
| | RM505Q-AE |
| | RM510Q-GL |

## 1.2. Process of Using FILE AT Commands

The general procedure for uploading/downloading, opening/creating, reading and writing to a file in the storage:

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

Use the following commands to manage files on the storage medium:

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

## 1.3. Description of Data Mode

The module COM port has two working modes: the AT command mode and the data mode. In the AT command mode, the data inputted via COM port will be treated as AT commands; while in the data mode, they will be treated as data.

- **Enter Data Mode**

  Execute **AT+QFUPL**, **AT+QFDWL**, **AT+QFREAD** or **AT+QFWRITE**, the module returns **CONNECT**, and the COM port enters the data mode. Or execute **ATO** to re-enter the data mode.

- **Exit Data Mode**

  Input **+++** to make the COM port exit the data mode. After **+++** is inputted, the execution of data-mode-entering commands will be interrupted before the module responds **OK**, and at this time, the COM port cannot re-enter the data mode if you execute **ATO**.

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

  1. Do not input any character for at least 1 second before inputting **+++**.
  2. Finish inputting **+++** within 1 second, during which no other character shall be inputted.
  3. After inputting **+++**, wait until **OK** is returned. The response **OK** indicates that the COM port exits the data mode.

# 2 Description of FILE AT Commands

## 2.1. AT Command Introduction

### 2.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on command line. When an optional parameter is omitted, the new value equals its previous value or its default setting, 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 2: Type of AT Commands**

| Command Type | Syntax | Description |
|---|---|---|
| Test Command | **AT+<cmd>=?** | Test the existence of the corresponding command and return information about the type, value, or range of its parameter. |
| Read Command | **AT+<cmd>?** | Check the current parameter value of the corresponding command. |
| Write Command | **AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]** | Set user-definable parameter value. |
| Execution Command | **AT+<cmd>** | 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 Space Information of Storage Medium

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

| AT+QFLDS Get Space Information of Storage Medium | |
|---|---|
| Test Command<br>**AT+QFLDS=?** | Response<br>**OK** |
| Write Command<br>**AT+QFLDS=<name_pattern>** | Response<br>**+QFLDS: <free_size>,<total_size>**<br><br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Execution Command<br>**AT+QFLDS** | Response<br>Return the UFS information:<br>**+QFLDS: <UFS_file_size>,<UFS_file_number>**<br><br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

**Parameter**

| | |
|---|---|
| **<name_pattern>** | String type. Storage medium type.<br>"UFS" UFS |

| | |
|---|---|
| | "SD" SD card |
| **<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 the UFS. Unit: byte. |
| **<UFS_file_number>** | Integer type. Number of files in the UFS. |
| **<err>** | Integer type. Error code. See ***Chapter 4*** for details. |

### Example

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

OK
AT+QFLDS="SD"                     //Query the space information of SD card.
+QFLDS: 251920384,253132800

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 the specified storage medium.

| **AT+QFLST List File Information on Storage Medium** | |
|---|---|
| Test Command<br>**AT+QFLST=?** | Response<br>**OK** |
| Write Command<br>**AT+QFLST=<filename>** | Response<br>**+QFLST: <filename>,<file_size>**<br>**[+QFLST: <filename>,<file_size>**<br>**[…]]**<br><br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Execution Command<br>**AT+QFLST** | Response<br>Return the information of the UFS files:<br>**+QFLST: <filename>,<file_size>**<br>**[+QFLST: <filename>,<file_size>**<br>**[…]]**<br><br>**OK**<br><br>If there is an error related to ME functionality: |

| | |
|---|---|
| | **+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

## Parameter

| | |
|---|---|
| **<filename>** | String type. Name pattern of the file(s) to be listed.<br>"*" All files in the UFS<br>"**<file>**" Specified file in the UFS<br>"UFS:*" All files in the UFS<br>"UFS:**<file>**" Specified file in the UFS<br>"SD:*" All files in the SD card<br>"SD:**<file>**" Specified file in the SD card<br>The **<file>** in values above indicates the exact name of the file with a maximum length of 80 bytes. |
| **<file_size>** | Integer type. File size. Unit: byte. |
| **<err>** | Integer type. Error code. See ***Chapter 4*** for details. |

## Example

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

OK
AT+QFLST="SD:*"                       //List all files in the SD card.
+QFLST: "SD:1k.txt",1024
+QFLST: "SD:10k.txt",10240
+QFLST: "SD:100k.txt",102400

OK
```

**NOTE**

**AT+QFLST** queries the actual size of the file currently stored in flash. After the data is written in with **AT+QFWRITE**, if you fail to query the file size by executing **AT+QFLST**, please close the file with **AT+QFCLOSE** before another query.

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

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

| AT+QFDEL Delete File(s) on Storage Medium | |
|---|---|
| Test Command<br>**AT+QFDEL=?** | Response<br>**+QFDEL: <filename>**<br><br>**OK** |
| Write Command<br>**AT+QFDEL=<filename>** | Response<br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

**Parameter**

| | | |
|---|---|---|
| **<filename>** | String type. Name pattern of the file to be deleted. | |
| | "*" | Delete all files in the UFS (do not delete the directory) |
| | "**<file>**" | Delete the specified file in the UFS |
| | "UFS:*" | Delete all files in the UFS (do not delete the directory) |
| | "UFS:**<file>**" | Delete the specified file in the UFS |
| | "SD:*" | Delete all files in the SD card (do not delete the directory) |
| | "SD:**<file>**" | Delete the specified file in the SD card |
| | The **<file>** in values above indicates the exact name of the file with a maximum length of 80 bytes. | |
| **<err>** | Integer type. Error code. See ***Chapter 4*** for details. | |

**Example**

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

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

This command uploads a file to a storage medium. If any file on the storage medium has the same name

as the file to be uploaded, an error will be reported.

During data transmission, you can send "**+++**" to make the module exit the data mode. For more details, see ***Chapter 1.3***.

For the Linux example of this command, see the package *quec_upload.zip*. For the example introduction, see the Readme file in *quec_upload.zip*. Please contact Quectel Technical Support to obtain the package.

| **AT+QFUPL Upload a File to Storage Medium** | |
|---|---|
| Test Command<br>**AT+QFUPL=?** | Response<br>**+QFUPL: <filename>,(1-<free_size>),(**range of supported **<timeout>**s**),(**list of supported **<ackmode>**s**)**<br><br>**OK** |
| Write Command<br>**AT+QFUPL=<filename>[,<file_size>[,<timeout>[,<ackmode>]]]** | Response<br>**CONNECT**<br>TA switches to the data mode (transparent transmission mode), and the binary data of the file can be inputted. When the total size of the inputted data reaches **<file_size>**, the TA receives **+++**, or the input time reaches **<timeout>**, the TA will return to the AT command mode and respond:<br>**+QFUPL: <upload_size>,<checksum>**<br><br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

## Parameter

| | |
|---|---|
| **<free_size>** | Integer type. Free space size of **<name_pattern>**. See ***Chapter 2.3.1*** for details. |
| **<filename>** | String type. Name pattern of the file to be uploaded.<br>"**<file>**" The specified file to be uploaded to UFS<br>"UFS:**<file>**" The specified file to be uploaded to UFS<br>"SD:**<file>**" The specified file to be uploaded to SD card<br>The **<file>** in values above indicates the exact name of the file with a maximum length of 80 bytes. |
| **<file_size>** | Integer type. File size expected to be uploaded. Default: 10240. Unit: byte. The maximum length is not greater than **<free_size>**. |
| **<upload_size>** | Integer type. Actual size of the uploaded data. Unit: byte. |

| | |
|---|---|
| **<timeout>** | Integer type. Waiting time for inputting data to USB/UART. Range: 1–65535. Default: 5. Unit: second. |
| **<ackmode>** | Integer type. Indicates whether to use ACK mode.<br>0 Turn off the ACK mode<br>1 Turn on the ACK mode |
| **<checksum>** | Integer type. Checksum of the uploaded data. |
| **<err>** | Integer type. Error code. See ***Chapter 4*** for details. |

**NOTE**

1. It is strongly recommended to use DOS 8.3 file name format for **<filename>**.
2. **<checksum>** is a 16-bit checksum based on bitwise Exclusive-OR（XOR). When the inputted data is odd byte size, the XOR operator sets the last inputted byte as the upper 8 bits, and sets the lower 8 bits as 0. The checksum verifies if the data have been uploaded correctly. Inputting **+++** will make the TA end the data transmission and switch to the AT command mode. However, the previously uploaded data will be preserved in the file.
3. When executing the command, the data must be inputted after **CONNECT** is returned.
4. The ACK mode is a safeguard against data loss when uploading large files if the hardware flow control does not work. The ACK mode works as follows:
   1) Run **AT+QFUPL=<filename>,<file_size>,<timeout>,1** to enable the ACK mode.
   2) The module outputs **CONNECT**.
   3) MCU sends 1 KB of data, to which the module responds with an **A**.
   4) MCU receives the **A** and then sends the next 1 KB of data.
   5) Steps 3) and 4) are repeated until the transfer is completed.

   For an example of ACK mode use, see ***Chapter 3.1.1.2***.

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

This command downloads a specified file from the storage medium.

For the Linux example of this command, see the introduction of **AT+QFUPL**.

| **AT+QFDWL Download a File from Storage Medium** | |
|---|---|
| Test Command<br>**AT+QFDWL=?** | Response<br>**+QFDWL: <filename>**<br><br>**OK** |
| Write Command<br>**AT+QFDWL=<filename>** | Response<br>**CONNECT**<br>TA switches to the data mode, so the binary data of the file can be outputted. When the content of the file is read or the TA receives **+++**, the TA will return to the AT command mode and respond: |

| | |
|---|---|
| | **+QFDWL: <download_size>,<checksum>**<br><br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

### Parameter

| | |
|---|---|
| **<filename>** | String type. Name pattern of the file to be downloaded.<br>"**<file>**" The specified UFS file to be downloaded<br>"UFS:**<file>**" The specified UFS file to be downloaded<br>"SD: **<file>**" The specified SD file to be downloaded<br>The **<file>** in values above indicates the exact name of the file with a maximum length of 80 bytes. |
| **<download_size>** | Integer type. Size of the downloaded data. |
| **<checksum>** | Integer type. Checksum of the downloaded data. |
| **<err>** | Integer type. Error code. See ***Chapter 4*** for details. |

**NOTE**

1. **+++** will cause the TA to end the data transmission and switch to the AT command mode.
2. **<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<br>**AT+QFOPEN=?** | Response<br>**+QFOPEN: <filename>,(**range of supported **<mode>**s**)**<br><br>**OK** |
| Read Command<br>**AT+QFOPEN?** | Response<br>**+QFOPEN: <filename>,<filehandle>,<mode>**<br>**[+QFOPEN: <filename>,<filehandle>,<mode>**<br>**[…]]** |

| | |
|---|---|
| | **OK** |
| Write Command<br>**AT+QFOPEN=<filename>[,<mode>]** | Response<br>**+QFOPEN: <filehandle>**<br><br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

### Parameter

| | |
|---|---|
| **<filename>** | String type. Name pattern of the file to be opened.<br>"**<file>**" The specified UFS **<file>** to be opened<br>"UFS:**<file>**" The specified UFS **<file>** to be opened<br>"SD:**<file>**" The specified SD **<file>** to be opened<br>The **<file>** in values above indicates the exact name of the file with a maximum length of 80 bytes. |
| **<filehandle>** | Integer type. Handle of the file to be operated. |
| **<mode>** | Integer type. File opening mode.<br>0 If the file exists, it will be opened, otherwise, it will be created. In both cases, the file can be read and written to<br>1 If the file exists, it will be overwritten, otherwise, it will be created. In both cases, the file can be read and written to<br>2 If the file exists, it will be opened as a read-only file, otherwise, an error will be returned |
| **<err>** | Integer type. Error code. See ***Chapter 4*** 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.

For the Linux example of this command, see the introduction of **AT+QFUPL**.

| **AT+QFREAD Read a File** | |
|---|---|
| Test Command<br>**AT+QFREAD=?** | Response<br>**+QFREAD: <filehandle>,<length>**<br><br>**OK** |

| | |
|---|---|
| Write Command<br>**AT+QFREAD=<filehandle>[,<length>]** | Response<br>**CONNECT <read_length>**<br>TA switches to the data mode. When the total size of the data reaches **<length>** or the TA receives **+++**, the TA will return to the AT command mode, display the result, and then respond:<br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

### Parameter

| | |
|---|---|
| **<filehandle>** | Integer type. Handle of the file to be operated. |
| **<length>** | Integer type. Expected length of file to be read. Default: 10240; Unit: byte. If the file length is less than 10240 bytes, the actual length of the file will be read. |
| **<read_length>** | Integer type. Actual read length. Unit: byte. |
| **<err>** | Integer type. Error code. See ***Chapter 4*** for details. |

## 2.3.8. AT+QFWRITE Write Data into 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.

For the Linux example of this command, see the introduction of **AT+QFUPL**.

| **AT+QFWRITE Write Data into a File** | |
|---|---|
| Test Command<br>**AT+QFWRITE=?** | Response<br>**+QFWRITE: <filehandle>,<length>,<timeout>**<br><br>**OK** |
| Write Command<br>**AT+QFWRITE=<filehandle>[,<length>[,<timeout>]]** | Response<br>**CONNECT**<br>TA switches to the data mode. When the total size of the written data reaches **<length>**, the TA receives **+++**, or the time reaches **<timeout>**, the TA will return to the AT command mode and respond<br>**+QFWRITE: <written_length>,<total_length>**<br><br>**OK** |

| | |
|---|---|
| | If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

### Parameter

| | |
|---|---|
| **<filehandle>** | Integer type. Handle of the file to be operated. |
| **<length>** | Integer type. Length of the file which data are written into. Default: 10240; Unit: byte. The maximum value of this parameter is determined by **<free_size>** of **AT+QFUPL**. |
| **<timeout>** | Integer type. Waiting time for inputting data to USB/UART. Default: 5. Unit: second. |
| **<written_length>** | Integer type. Actual written length. Unit: byte. |
| **<total_length>** | Integer type. Total length of the file. Unit: byte. |
| **<err>** | Integer type. Error code. See ***Chapter 4*** 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**.

| **AT+QFSEEK Set File Pointer to Specified Position** | |
|---|---|
| Test Command<br>**AT+QFSEEK=?** | Response<br>**+QFSEEK: <filehandle>,<offset>,<position>**<br><br>**OK** |
| Write Command<br>**AT+QFSEEK=<filehandle>,<offset>[,<position>]** | Response<br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

### Parameter

| | |
|---|---|
| **<filehandle>** | Integer type. Handle of the file to be operated. |

| | |
|---|---|
| **<offset>** | Integer type. Number of bytes of file pointer movement. |
| **<position>** | Integer type. Pointer movement mode.<br>0 Move forward from the beginning of the file<br>1 Move forward from the current position of the pointer<br>2 Move backward from the end of the file |
| **<err>** | Integer type. Error code. See ***Chapter 4*** for details. |

**NOTE**

If the set final position of the pointer exceeds the file size, **ERROR** will be returned.

### 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 Offset of File Pointer** | |
|---|---|
| Test Command<br>**AT+QFPOSITION=?** | Response<br>**+QFPOSITION: <filehandle>**<br><br>**OK** |
| Write Command<br>**AT+QFPOSITION=<filehandle>** | Response<br>**+QFPOSITION: <offset>**<br><br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

#### Parameter

| | |
|---|---|
| **<filehandle>** | Integer type. Handle of the file to be operated. |
| **<offset>** | Integer type. Offset from the beginning of file. |
| **<err>** | Integer type. Error code. See ***Chapter 4*** 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<br>**AT+QFCLOSE=?** | Response<br>**+QFCLOSE: <filehandle>**<br><br>**OK** |
| Write Command<br>**AT+QFCLOSE=<filehandle>** | Response<br>**OK**<br><br>If there is an error related to ME functionality:<br>**+CME ERROR: <err>** |
| Maximum Response Time | 300 ms |
| Characteristics | The command takes effect immediately;<br>The configurations will not be saved. |

### Parameter

| | |
|---|---|
| **<filehandle>** | Integer type. Handle of the file to be operated. |
| **<err>** | Integer type. Error code. See ***Chapter 4*** for details. |

# 3 Examples

## 3.1. Upload and Download Files

### 3.1.1. Upload a File

#### 3.1.1.1. Non-ACK Mode

```
AT+QFUPL="test1.txt",10                 //Upload the text file test1.txt to the UFS.
CONNECT
<Input file bin data>                   //Input the binary data of the file.
+QFUPL: 10,3938

OK
```

#### 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 about ACK mode, see **AT+QFUPL**.

```
AT+QFUPL="test.txt",3000,5,1            //Upload the text file test.txt to the UFS.
CONNECT
<input file bin data of 1024 bytes>     //Input 1024 bytes of the binary data of the file.
A                                       //After MCU sends 1024 bytes of data, the module will respond
                                          an A. Then the next 1024 bytes of data can be inputted.
<input file bin data of 1024 bytes>     //Input 1024 bytes of the binary data of the file.
A
<input the rest file bin data>          //Input the rest binary data of the file.
+QFUPL: 3000,B34A

OK
```

### 3.1.2. Download a File

```
AT+QFDWL="test.txt"                 //Download the text file test.txt from UFS.
CONNECT
<Output data>                       //Binary data of the file is outputted.
+QFDWL: 10,613e                     //Size and checksum value of the downloaded data are returned.

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: 3023

OK
AT+QFWRITE=3023,10                  //Write 10 bytes to the file.
CONNECT
<Write data>                        //Write in the file data in the data mode.
+QFWRITE: 10,10                     //The actual written bytes and the size of the file are returned.

OK
AT+QFSEEK=3023,0,0                  //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=3023,10                   //Read 10 bytes from the file.
CONNECT 10
<Read data>                         //Read data in the data mode.

OK
AT+QFCLOSE=3023                     //Close the file.
OK
```

### 3.2.2. Write to and Read an SD File

```
AT+QFOPEN="SD:1.txt",1              //Open the file to get the file handle.
+QFOPEN: 3024

OK
AT+QFWRITE=3024,1024                //Write 1024 bytes to the file.
CONNECT
<Write data>                        //Write in the file data in the data mode.
```

```
+QFWRITE: 1024,1024                 //The actual written bytes and the size of the file are returned.

OK
AT+QFSEEK=3024,0,0                  //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=3024,1024                 //Read data from the file.
CONNECT 1024
<Read data>                         //Read data in the data mode.
OK
AT+QFCLOSE=3024                     //Close the file.
OK
```

# 4 Summary of Error Codes

The error code **<err>** indicates an error related to mobile equipment or network. The details about **<err>** are presented in the following table.

**Table 3: Summary of Error Codes**

| <err> | Description |
|---|---|
| 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 | Failed to write to the file |
| 410 | Failed to open the file |
| 411 | Failed to read the file |
| 413 | Reached 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 4: Terms and Abbreviations**

| Abbreviation | Description |
|---|---|
| ACK | Acknowledgement |
| COM | Communication Port |
| DOS | Disk Operating System |
| ME | Mobile Equipment |
| SD | Secure Digital |
| TA | Terminal Adapter |
| UART | Universal Asynchronous Receiver-Transmitter |
| UFS | User File Storage |
| USB | Universal Serial Bus |
| XOR | Exclusive OR |