EC200U&EG915U Series
FILE Application Note

LTE Standard Module Series

Version: 1.1

Date: 2021-09-01

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.

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
-	2020-06-30	Burols WANG/ Herry GEMG	Creation of the document
1.0	2021-07-22	Burols WANG/ Herry GENG	First official release
1.1	2021-09-01	Herry GENG	1. Added applicable module EG915U series. 2. Added notes for the modules that do not currently support storage mediums "**SD:**" and "**EFS:**" (Chapter 1). 3. Updated the storage medium type of **AT+QFLDS**, and added the note that "**SFS:**" is not supported: (Chapter 2.3.1). 4. Updated the description of **<filename>**. 5. Added the maximum length of **<dirname>**.

Content

Table Index

1 Introduction

Quectel EC200U series and EG915U series modules provide AT commands to operate files on different physical storage mediums. This document is a reference guide to these commands.

Quectel EC200U series and EG915U series modules support the following storage mediums:

- **UFS:** Primary partition. It is used to store user's ordinary files.
- **SFS:** Encrypted file directory. The same storage medium as UFS, used to store user's encrypted files.
- **EFS:** External flash file system partition.
- **SD:** SD card partition.

NOTES

1. The file name indicates the storage location. When the file name begins with "**UFS:**", it means that the file is stored in UFS. When the file name begins with "**SD:**", it means that the file is stored in SD card. And if there are no prefix characters in the file name, then the file is also stored in UFS.
2. Currently EC200U series and EG915U series modules do not support the storage medium "**RAM:**" Random Access Memory.
3. Currently EC200U series module does not support the storage medium "**SD:**" SD card.
4. Currently EG915U series modules does not support the storage mediums "**SD:**" SD Card and "**EFS:**" External Flash File System.

1.1. The Process of Using FILE AT Commands

The following procedures can be followed to create, read and write a file in the storage:

1. Upload a file to the storage by **AT+QFUPL**, and output/download it through the serial interface by **AT+QFDWL**.
2. Open the file by **AT+QFOPEN**, and then the file can be written or read at any time and any location until the file is closed by **AT+QFCLOSE**.
   - When using **AT+QFOPEN** to open a file, you can set the file to overwrite mode, read-only mode or other modes by the parameter **<mode>** (For more details about **<mode>**, see ***Chapter 0***). After opening the file, a parameter **<filehandle>** is assigned to it. Then the file can be operated by **<filehandle>**.

- After opening the file, write the data to the file by **AT+QFWRITE** and read the data by **AT+QFREAD** from the current file position.
- Set the file position by **AT+QFSEEK** and query the current file position by **AT+QFPOSITION**.
- Close the file by **AT+QFCLOSE**, after which the **<filehandle>** turns invalid any more.

Use the following commands to manage files in 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** in time after the operation is completed, otherwise the file handle will be leaked.

1.2. Description of Data Mode

The COM port of EC200U series and EG915U series modules have two working modes: AT command mode and data mode. In AT command mode, the inputted data via COM port will be treated as AT command; while in data mode, it will be treated as data.

Inputting "**+++**" or pulling up DTR (**AT&D1** should be set first) can make the COM port exit data mode. To prevent the "**+++**" from being mistaken for data, the following standards should be followed before using the COM port:

1) Do not input any character within 1 s or longer before inputting "**+++**".
2) Input "**+++**" within 1 s, and no other characters can be inputted during the time.
3) Do not input any character within 1 s after "**+++**" has been inputted.

When **AT+QFUPL**, **AT+QFDWL**, **AT+QFREAD** and **AT+QFWRITE** are executed, the COM port will enter data mode. If you are using "**+++**" or DTR to make the port exit data mode, the executing procedure of these commands will be interrupted before the response is returned. In such case, the COM port cannot reenter data mode by executing **ATO**.

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 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 to 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	**AT+<cmd>=?**	Test the existence of corresponding Write Command and return information about the type, value, or range of its parameter.
Read Command	**AT+<cmd>?**	Check the current parameter value of a corresponding Write 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 familiarize with AT commands and learn how to use them. The examples, however, should not be taken as Quectel's recommendation or suggestions about how you should design a program flow or what status you should set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there exists a correlation among these examples and that they should be executed in a given sequence.

2.3. AT Command Description

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

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

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

Parameter

<name_pattern>	String type. Storage medium type. "UFS" UFS "EFS" External Flash "SD" SD card
<freesize>	Integer type. The free space size of **<name_pattern>**.
<total_size>	Integer type. The total space size of **<name_pattern>**.
<UFS_file_size>	Integer type. The size of all files in UFS. Unit: byte.
<UFS_file_number>	Integer type. The number of files in UFS.
<err>	The code of an error relating to ME. 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
```

NOTE

This command does not support getting the space information in the "SFS:" directory.

2.3.2. AT+QFLST List the File Information in the Storage Medium

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

AT+QFLST List the File Information in the Storage Medium	
Test Command **AT+QFLST=?**	Response **OK**
Write Command **AT+QFLST=<name_pattern>**	Response **+QFLST: <filename>,<file_size>** **[+QFLST: <filename>,<file_size>** **[…]]** **OK**

	If there is an error related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+QFLST**	Response Return the information of the UFS files: **+QFLST: <filename>,<file_size>** **[+QFLST: <filename>,<file_size>** **[…]]** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<name_pattern>	String type. The file to be listed.	
	"*"	All the files in UFS
	"**<filename>**"	The specified file **<filename>** in UFS
	"UFS:*"	All the files in UFS
	"UFS:**<filename>**"	The specified file **<filename>** in UFS
	"SFS:*"	All the files in SFS
	"SFS:**<filename>**"	The specified file **<filename>** in SFS
	"EFS:*"	All the files in EFS
	"EFS:**<filename>**"	The specified file **<filename>** in EFS
	"SD:*"	All the files in SD card
	"SD:**<filename>**"	The specified file **<filename>** in SD card
<filename>	String type. File name.	
<file_size>	Integer type. File size. Unit: byte.	
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.	

Example

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

OK
AT+QFLST="SD:*"                       //List all the files in 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. Use **AT+QFWRITE** to write data. If **AT+QFLST** cannot directly query the file size, you need to execute **AT+QFCLOSE** to close the file and then query the file size.

2.3.3. AT+QFDEL Delete the File(s) in the Storage Medium

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

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

Parameter

<filename>	String type. Name of the file to be deleted. **<filename>** can include the file path (that is the directory name) and file name. The maximum length of the file path is 58 bytes, and the maximum length of the file name is 63 bytes.	
	"*"	Delete all the files in UFS (not delete the directory)
	"**<filename>**"	Delete the specified file **<filename>** in UFS
	"UFS:*"	Delete all the files in UFS (not delete the directory)
	"UFS:**<filename>**"	Delete the specified file **<filename>** in UFS
	"SFS:*"	Delete all the files in SFS (not delete the directory)）
	"SFS:**<filename>**"	Delete the specified file **<filename>** in SFS
	"EFS:*"	Delete all the files in EFS (not delete the directory)
	"EFS:**<filename>**"	Delete the specified file **<filename>** in EFS

	"SD:*"	Delete all the files in SD card (not delete the directory)
	"SD:**<filename>**"	Delete the specified file **<filename>** in SD card
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.	

Example

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

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

This command uploads a file to storage medium. If there is any file in the storage which has the same name with the file to be uploaded, an error will be reported. If the inputted file name includes a path, and the path does not exist, this directory will be created under the supported root directory (for example, the UFS: directory), and the file will be uploaded to this directory.

After executing the Write Command and **CONNECT** returns, the module will switch to data mode. When the uploaded data reaches **<file_size>**, or there is no any data inputted when **<timeout>** reaches, then it will exit data mode automatically. During data transmission, you can use "**+++**" or DTR to make the module exit data mode. For more details, see ***Chapter 1.2***.

AT+QFUPL Upload a File to the Storage Medium	
Test Command **AT+QFUPL=?**	Response **+QFUPL: <filename>[,(1-<freesize>)[,(**range of supported **<timeout>**s**)[,(**list of supported **<ackmode>**s**)]]]** **OK**
Write Command **AT+QFUPL=<filename>[,<file_size>[,<timeout>[,<ackmode>]]]**	Response **CONNECT** TA switches to the data mode (transparent access mode), and the binary data of file can be inputted. When the total size of the inputted data reaches **<file_size>** (unit: byte), TA will return to command mode and reply the following codes: **+QFUPL: <upload_size>,<checksum>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**

Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<freesize>	Integer type. The free space size of **<name_pattern>**. See **AT+QFLDS** in ***Chapter 2.3.1*** for more details of **<name_pattern>**.
<filename>	String type. Name of the file to be uploaded. **<filename>** can include the file path (that is the directory name) and file name. The maximum length of the file path is 58 bytes, and the maximum length of the file name is 63 bytes. "**<filename>**" Name of the file to be uploaded to UFS "UFS:**<filename>**" Name of the file to be uploaded to UFS "SFS:**<filename>**" Name of the file to be uploaded to SFS "EFS:**<filename>**" Name of the file to be uploaded to EFS "SD:**<filename>**" Name of the file to be uploaded to SD card
<file_size>	Integer type. The file size expected to be uploaded. Default: 10240. Unit: byte.
<upload_size>	Integer type. The actual size of the uploaded data. Unit: byte.
<timeout>	Integer type. The time waiting for data to be inputted to USB/UART. Range: 1–65535. Default: 5. Unit: s.
<ackmode>	Integer type. Whether to use ACK mode. 0 Turn off the ACK mode 1 Turn on the ACK mode
<checksum>	Integer type. The checksum of the uploaded data.
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.

NOTES

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 XOR.
   If the number of the 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. **+++** sequence will cause TA to end the command and switch to command mode. However, the data previously uploaded will be preserved into the file.
3. When executing the command, the data must be inputted after **CONNECT** is returned.
4. The ACK mode is provided to avoid the loss of data when uploading large files, in case 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 bytes data, and then module will respond with an **A**.
   4) MCU receives the **A** and then sends the next 1 KB bytes data.
   5) Repeat step 3) and 4) until the transfer is completed.

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

This command downloads a specified file from the storage medium.

AT+QFDWL Download a File from the Storage Medium	
Test Command **AT+QFDWL=?**	Response **+QFDWL: <filename>** **OK**
Write Command **AT+QFDWL=<filename>**	Response **CONNECT** TA switches to data mode, and the binary data of the file will be outputted. When the file is read over, TA will return to command mode and reply the following codes: **+QFDWL: <download_size>,<checksum>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filename>	String type. Name of the file to be downloaded. **<filename>** can include the file path (that is the directory name) and file name. The maximum length of the file path is 58 bytes, and the maximum length of the file name is 63 bytes. "**<filename>**" Name of the UFS file to be downloaded "UFS:**<filename>**" Name of the UFS file to be downloaded "SFS:**<filename>**" Name of the SFS file to be downloaded "EFS:**<filename>**" Name of the EFS file to be downloaded "SD: **<filename>**" Name of the SD file to be downloaded
<download_size>	Integer type. The size of the downloaded data.
<checksum>	Integer type. The checksum of the downloaded data.
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.

NOTES

1. **+++** sequence will cause TA to end the command and switch to command mode.
2. **<checksum>** is a 16-bit checksum based on bitwise XOR.

3. This command does not return the contents of the files in the "SFS:" directory, and it only returns the file size and checksum.

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 **AT+QFOPEN=?**	Response **+QFOPEN: <filename>[,(**range of supported **<mode>**s**)]** **OK**
Read Command **AT+QFOPEN?**	Response **+QFOPEN: <filename>,<filehandle>,<mode>** **[+QFOPEN: <filename>,<filehandle>,<mode>** **[…]]** **OK**
Write Command **AT+QFOPEN=<filename>[,<mode>]**	Response **+QFOPEN: <filehandle>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filename>	String type. Name of the file to be opened. **<filename>** can include the file path (that is the directory name) and file name. The maximum length of the file path is 58 bytes, and the maximum length of the file name is 63 bytes. "**<filename>**" Name of the UFS file to be opened "UFS:**<filename>**" Name of the UFS file to be opened "EFS:**<filename>**" Name of the EFS file to be opened "SD: **<filename>**" Name of the SD file to be opened
<filehandle>	Integer type. The handle of the file.
<mode>	Integer type. The open mode of the file. 0 If the file does not exist, it will be created. If the file exists, it will be directly

	opened. And both of them can be read and written. 1 If the file does not exist, it will be created. If the file exists, it will be overwritten and cleared. And both of them can be read and written. 2 If the file exists, open it and it can be read only. When the file does not exist, it will respond an error.
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.

NOTE

This command does not support opening files in the "SFS:" directory.

2.3.7. AT+QFREAD Read Data of a File

This command reads data of a file which is specified by the file handle. The data starts from the current position of the file pointer which belongs to the file handle.

AT+QFREAD Read Data of a File	
Test Command **AT+QFREAD=?**	Response **+QFREAD: <filehandle>[,<length>]** **OK**
Write Command **AT+QFREAD=<filehandle>[,<length>]**	Response **CONNECT <read_length>** TA switches to data mode. When the total size of the data reaches **<length>** (unit: byte), TA will return to command mode, display the result and then reply the following codes: **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<length>	Integer type. The expected length of the file to be read. The default length is 10 KB. If the file length is less than 10 KB, the actual length of the file will be read. Unit: bytes.
<read_length>	Integer type. The actual read length. Unit: bytes.

<err>	The code of an error relating to ME. 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 which belongs to the file handle.

AT+QFWRITE Write Data into a File	
Test Command **AT+QFWRITE=?**	Response **+QFWRITE: <filehandle>[,<length>[,<timeout>]]** **OK**
Write Command **AT+QFWRITE=<filehandle>[,<length>[,<timeout>]]**	Response **CONNECT** TA switches to data mode. When the total size of the written data reaches **<length>** or the time reaches **<timeout>**, TA will return to command mode and reply the following codes: **+QFWRITE: <written_length>,<total_length>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<length>	Integer type. The length of the file to be written, and the default length is 10 KB. The maximum value of this parameter is determined by **<freesize>** of **AT+QFUPL**. Unit: bytes.
<timeout>	Integer type. The time waiting for data to be inputted to USB/UART. Default: 5. Unit: s.
<written_length>	Integer type. The actual written length. Unit: bytes.
<total_length>	Integer type. The total length of the file. Unit: bytes.
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.

2.3.9. AT+QFSEEK Set a File Pointer to the Specified Position

This command sets a file pointer to the specified position.

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

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<offset>	Integer type. The number of bytes of the file pointer movement.
<position>	Integer type. Pointer movement mode. 0 The beginning of the file is moved backward 1 The current position of the file pointer moves backward 2 The end of the file moved forward
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.

NOTES

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

2.3.10. AT+QFPOSITION Get the Offset of a 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 **AT+QFPOSITION=?**	Response **+QFPOSITION: <filehandle>** **OK**

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

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<offset>	Integer type. The offset from the beginning of the file.
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.

2.3.11. AT+QFCLOSE Close a File

This command closes a file and ends the operation to the file. After that, the file handle is released and should not be used again, unless the file is opened again by **AT+QFOPEN**.

AT+QFCLOSE Close a File	
Test Command **AT+QFCLOSE=?**	Response **+QFCLOSE: <filehandle>** **OK**
Write Command **AT+QFCLOSE=<filehandle>**	Response **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<filehandle>	Integer type. The handle of the file to be operated.
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.

2.3.12. AT+QFMKDIR Create a Directory

This command creates a directory, and temporarily only supports creation to a secondary directory (that is, only one directory is allowed to be created under a supported directory, and directory creation under a secondary directory is not supported).

AT+QFMKDIR Create a Directory	
Test Command **AT+QFMKDIR=?**	Response **+QFMKDIR: <path>** **OK**
Write Command **AT+QFMKDIR=<dirname>**	Response **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<path>	String type. The directory path to be created.
<dirname>	String type. The name of the directory to be created. The maximum length is 58 bytes. "**<dirname>**" The name of the directory to be created in UFS "UFS:**<dirname>**" The name of the directory to be created in UFS "SFS:**<dirname>**" The name of the directory to be created in SFS "EFS:**<dirname>**" The name of the directory to be created in EFS "SD:**<dirname>**" The name of the directory to be created in SD card
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.

2.3.13. AT+QFRMDIR Delete a Directory

This command deletes a directory, and there are two modes. One is that if there are files in the directory, this command will not delete the directory or the files and will return the corresponding error code; if there is no file in the directory, the directory will be deleted. The other is that when there are files in the directory, this command will delete the directory and the files in the directory. If the execution succeeds, it returns **OK**, and if it fails, it returns the corresponding error code.

AT+QFRMDIR Delete a Directory	
Test Command **AT+QFRMDIR=?**	Response **+QFRMDIR: <path>,(**list of supported **<idelmode>**s**)** **OK**
Write Command **AT+QFRMDIR=<path>[,<idelmode>]**	Response **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; The configurations will not be saved.

Parameter

<path>	String type. The directory path to be deleted.
<idelmode>	Integer type. Delete mode. 0 if there are files in the directory, the delete operation will not be executed and an error code will be returned; if there is no file in the directory, the directory will be deleted. 1 There are files in the directory, the directory and the files in the directory will be deleted.
<err>	The code of an error relating to ME. See ***Chapter 4*** for details.

3 Examples

3.1. Upload and Download a File

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 UFS.
CONNECT
<Input file bin data>
+QFUPL: 10,3938

OK
AT+QFUPL="dir1/test1.txt",10            //Create a directory dir1 in the UFS, and upload test1.txt to this
                                          directory.
CONNECT
+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 be used to prevent the data from being lost. For more details about ACK mode, see **AT+QFUPL**.

```
AT+QFUPL="5K.txt",5120,30,1             //Upload the text file 5K.txt to UFS and turn on the ACK
                                          mode. The time waiting for data to be inputted is 30 s.
CONNECT
AAAAA                                   //Each time uploading 1 KB bytes data, the module will
                                          respond an A, and the uploading 5 KB bytes data, the
                                          module will respond 5 A.
+QFUPL: 5120,5e0e                       // A total of 5120 bytes data was successfully uploaded, with
                                          a checksum of 5e0e.
```

```
OK
```

3.1.2. Download a File

```
AT+QFDWL="10.txt"                  //Download the text file 10.txt from UFS.
CONNECT
1234567890                         //File contents.
+QFDWL: 10,3938                    //Get the bytes and the checksum value of the uploaded data.

OK
```

3.2. Write and Read a File

3.2.1. Write and Read a UFS File

```
AT+QFOPEN="ufs.txt"                //Open the file to get the file handle.
+QFOPEN: 1028

OK
AT+QFWRITE=1028,10,10              //Write 10 bytes to the file.
CONNECT
<Write Data>
+QFWRITE: 10,10                    //The actual bytes written and the size of the file are returned.

OK
AT+QFSEEK=1028,0,0                 //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=1028,10                  //Read the data.
CONNECT 10                         //Read 10 bytes.
<Read Data>
OK
AT+QFCLOSE=1028                    //Close the file.
OK
```

3.2.2. Write and Read a SD File

```
AT+QFOPEN="SD:1.txt",1             //Open the file to get the file handle.
+QFOPEN: 20

OK
AT+QFWRITE=20,1024                 //Write 1024 bytes to the file.
CONNECT
```

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

OK
AT+QFSEEK=20,0,0                    //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=20,1024                   //Read the data.
CONNECT
<Read Data>
OK
AT+QFCLOSE=20                       //Close the file.
OK
```

3.2.3. Create a Directory

```
AT+QFMKDIR="UFS:dir1"               //Create the directory dir1 in UFS.
OK
AT+QFMKDIR="dir2"                   //Create the directory dir2 in UFS.
OK
```

3.2.4. Delete a Directory

```
AT+QFLST="dir1"                     //List the specified file dir1 in UFS.
+QFLST: "dir1/upl1.txt",10
+QFLST: "dir1/upl2.txt",10

OK
AT+QFLST="dir2"                     //List the specified file dir2 in UFS.
+QFLST: "dir2/upl2.txt",10

OK
AT+QFRMDIR="dir1"                   //Delete the directory dir1 in UFS.
+CME ERROR: 418                     //There are files in the directory dir1, and an error code is
                                      returned.
AT+QFRMDIR="dir1",1                 //Delete the directory dir1 in UFS.
OK
```

4 Summary of Error Codes

The error code **<err>** indicates an error related to mobile equipment or network. For more details, see the following table. The error codes in the following table are related to the file operations of the applicable modules in this document.

Table 2: Summary of Error Codes

<err>	Meaning
400	Invalid input value
401	Larger than the size of the file
402	Read zero byte
403	Drive full
405	File not found
406	Invalid file name
407	File already exists
409	Fail to write 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	The file read-only
415	Get file size failed
416	Invalid file descriptor
417	Fail to list the file
418	Fail to delete the file

419	Fail to get disk info
420	No space
421	Time out
423	File too large
425	Invalid parameter
426	File already opened

5 Appendix Reference

Table 3: Terms and Abbreviations

Abbreviation	Description
ACK	Acknowledgement
COM	Communication Port
DOS	Disk Operating System
DTR	Data Terminal Ready
EFS	Embedded File System
ME	Mobile Equipment
RAM	Random Access Memory
SD	Secure Digital
SFS	Security File System
TA	Terminal Adapter
UART	Universal Asynchronous Receiver-Transmitter
UFS	User File Storage
USB	Universal Serial Bus
XOR	Exclusive OR