

# **BG96 FILE**

# **AT Commands Manual**

**LTE Module Series**

Rev. BG96\_FILE\_AT\_Commands\_Manual\_V1.1

Date: 2019-02-02

Status: Released



**Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:**

**Quectel Wireless Solutions Co., Ltd.**

7<sup>th</sup> Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto:info@quectel.com)

**Or our local office. 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 to: [support@quectel.com](mailto:support@quectel.com)

**GENERAL NOTES**

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

**COPYRIGHT**

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

***Copyright © Quectel Wireless Solutions Co., Ltd. 2019. All rights reserved.***

# About the Document

## History

Revision	Date	Author	Description
1.0	2018-01-24	Matt YE/ Kennen WU	Initial
1.1	2019-02-02	Egbert XU/ Matt YE	<ol style="list-style-type: none"><li>1. Enabled EUFS (Extended User File Storage on application side) for the module.</li><li>2. Added AT+QFCRC/+QFCPY commands.</li></ol>

## Contents

About the Document.....	2
Contents.....	3
Table Index.....	4
<b>1 Introduction .....</b>	<b>5</b>
1.1. The Process of Using File AT Commands .....	5
1.2. Description of Data Mode .....	6
<b>2 Description of FILE AT Commands .....</b>	<b>7</b>
2.1. AT+QFLDS Get the Space Information of Storage .....	7
2.2. AT+QFLST List Files .....	8
2.3. AT+QFDEL Delete Files .....	9
2.4. AT+QFUPL Upload a File.....	10
2.5. AT+QFDWL Download a File .....	12
2.6. AT+QFOPEN Open a File .....	13
2.7. AT+QFREAD Read a File.....	14
2.8. AT+QFWRITE Write a File .....	14
2.9. AT+QFSEEK Set File Pointer to a Position.....	15
2.10. AT+QFPOSITION Get the Offset of a File Pointer.....	16
2.11. AT+QFTUCAT Truncate a File from the File Pointer .....	17
2.12. AT+QFCLOSE Close a File.....	17
2.13. AT+QFCRC Calculate the CRC of a Specified UFS File .....	18
2.14. AT+QFCPY Make a Copy of a Specified File.....	19
<b>3 Examples .....</b>	<b>20</b>
3.1. Upload and Download Files.....	20
3.1.1. Upload a File .....	20
3.1.2. Download a File .....	21
3.2. Write and Read Files.....	21
3.2.1. Write and Read a UFS File .....	21
3.2.2. Write and Read a EUFS File.....	22
<b>4 Summary of ERROR Codes .....</b>	<b>23</b>
<b>5 Appendix A Reference.....</b>	<b>25</b>

## Table Index

TABLE 1: SUMMARY OF ERROR CODES .....	23
TABLE 2: TERMS AND ABBREVIATIONS .....	25

# 1 Introduction

Quectel BG96 module provides AT commands to operate files on different physical storage mediums. This document is a reference guide to these commands.

The supported storage mediums are as follows:

- UFS: User File Storage directory on modem side. It is a special directory on the flash file system.
- EUFS: Extended User File Storage on application side.

The file name will indicate the storage location. When the file name begins with “EUFS:”, it means the file is located in EUFS. If the file name begins with “UFS:” or there are no prefix characters in the file name, the file is located in UFS.

## 1.1. The Process of Using File AT Commands

The following illustrates the procedure to create, read and write a file in the storage:

1. A file can be uploaded to the storage by AT+QFUPL, and can be outputted/downloaded through the serial interfaces 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 the file is opened by AT+QFOPEN, it can be set as overwrite mode or read-only mode or others by the parameter <mode>. (For more information about <mode>, please refer to **Chapter 2.6**). After the file is opened, a <filehandle> is assigned to it. Then the file can be operated via this <filehandle>.
  - After the file is opened, it can be written by AT+QFWRITE and read by AT+QFREAD from the current file position.
  - The file position can be set by AT+QFSEEK and the current position can be queried by AT+QFPOSITION.
  - AT+QFTUCAT will truncate the file from the current position to the end of the file.
3. AT+QFCLOSE is used to close the file, after which the <filehandle> becomes meaningless any more.

The following are several commonly used commands to manage files in the storage:

1. AT+QFLDS: Get the storage size
2. AT+QFLST: List the file information in the storage.
3. AT+QFDEL: Delete the file(s)

## 1.2. Description of Data Mode

The COM port of BG96 module has 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 from data mode. To prevent the “+++” from being misinterpreted as data, the following sequence should be followed:

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

When AT+QFUPL, AT+QFDWL, AT+QFREAD and AT+QFWRITE are executed, the COM port will enter data mode. If customers are using “+++” or DTR to make the port exit from 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 command.

## 2 Description of FILE AT Commands

### 2.1. AT+QFLDS Get the Space Information of Storage

The command responds the space information of the specified storage.

#### AT+QFLDS Get the Space Information of Storage

Test Command <b>AT+QFLDS=?</b>	Response <b>OK</b>
Write Command <b>AT+QFLDS=&lt;name_pattern&gt;</b>	Response <b>+QFLDS: &lt;freesize&gt;,&lt;total_size&gt;</b>  <b>OK</b>  If there is any error: <b>+CME ERROR: &lt;err&gt;</b>
Execution Command <b>AT+QFLDS</b>	Response Return the UFS information: <b>+QFLDS: &lt;UFS_file_size&gt;,&lt;UFS_file_number&gt;</b>  <b>OK</b>  If there is any error: <b>+CME ERROR: &lt;err&gt;</b>

#### Parameter

<b>&lt;name_pattern&gt;</b>	Storage pattern. “UFS” UFS on modem side “EUFS” Extended UFS on AP side
<b>&lt;freesize&gt;</b>	The free space size of <name_pattern>. Unit: byte.
<b>&lt;total_size&gt;</b>	The total size of <name_pattern>. Unit: byte.
<b>&lt;UFS_file_size&gt;</b>	The size of all files in UFS. Unit: byte
<b>&lt;UFS_file_number&gt;</b>	The number of files in UFS.
<b>&lt;err&gt;</b>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.



## Example

```
AT+QFLDS="UFS" //Query the space information of UFS.
+QFLDS: 12987840,14483456
OK
```

## 2.2. AT+QFLST List Files

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

### AT+QFLST List Files

Test Command	Response
AT+QFLST=?	OK
Write Command AT+QFLST=<name_pattern>	Response +QFLST: <filename>,<file_size> [+QFLST: <filename>,<file_size> [...]]  OK  If there is any error: +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 any error: +CME ERROR: <err>

### Parameter

<name_pattern>	The file to be listed.
"*"	All the files in UFS
"EUS:/ufs/*"	All the files in EUFS's <i>ufs</i> directory
"EUS:/datatx/*"	All the files in EUFS's <i>datatx</i> directory
"EUS:/fota/*"	All the files in EUFS's <i>fota</i> directory
"EUS:/*"	All the files in EUFS's <i>ufs</i> directory

"<filename>"	A specified file <filename> in UFS
"EUFS:<filename>"	A specified file <filename> in EUFS's <i>ufs</i> directory
"EUFS:/ufs/<filename>"	A specified file <filename> in EUFS's <i>ufs</i> directory
"EUFS:/datatx/<filename>"	A specified file <filename> in EUFS's <i>datatx</i> directory
"EUFS:/fota/<filename>"	A specified file <filename> in EUFS's <i>fota</i> directory
<filename>	Name of the file.
<file_size>	Size of the file. Unit: byte.
<err>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

### Example

```
AT+QFLST="*" //List all the files in UFS.
+QFLST: "F_M12-1.bmp",562554
+QFLST: "F_M12-10.bmp",562554
+QFLST: "F_M12-11.bmp",562554
```

## 2.3. AT+QFDEL Delete Files

The command deletes a specified file or all the files in the storage.

### AT+QFDEL Delete Files

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 any error: <b>+CME ERROR: &lt;err&gt;</b>

### Parameter

<filename>	Name of the file to be deleted. The maximum length of <filename> is 80 bytes.
"*"	Delete all the files in UFS (not delete the directory)
"EUFS:*"	Delete all the files in EUFS's <i>ufs</i> directory (not delete the directory)
"EUFS:/ufs/*"	Delete all the files in EUFS's <i>ufs</i> directory (not delete the directory)
"EUFS:/datatx/*"	Delete all the files in EUFS's <i>datatx</i> directory (not delete the directory)
"EUFS:/fota/*"	Delete all the files in EUFS's <i>fota</i> directory (not delete the

	directory)
"<filename>"	Delete a specified file <filename> in UFS
"EUPS:<filename>"	Delete a specified file <filename> in EUPS's <i>ufs</i> directory
"EUPS:/ufs/<filename>"	Delete a specified file <filename> in EUPS's <i>ufs</i> directory
"EUPS:/datatx/<filename>"	Delete a specified file <filename> in EUPS's <i>datatx</i> directory
"EUPS:/fota/<filename>"	Delete a specified file <filename> in EUPS's <i>fota</i> directory
<err>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

### Example

```
AT+QFDEL=""  
OK
```

## 2.4. AT+QFUPL Upload a File

The command can be used to upload a file to the storage. If any file in the storage has the same name with the file uploaded, an error will be reported.

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 from data mode automatically. During data transmission, customers can use "+++" or DTR to make the module exit from data mode, and more details are provided in **Chapter 1.2**.

### AT+QFUPL Upload a File

Test Command <b>AT+QFUPL=?</b>	Response <b>+QFUPL: &lt;filename&gt;[(1-&lt;freesize&gt;)][(1-65535)][(0,1)]</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 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: <b>+QFUPL: &lt;upload_size&gt;,&lt;checksum&gt;</b>  <b>OK</b>  If there is any error: <b>+CME ERROR: &lt;err&gt;</b>

## Parameter

<b>&lt;freesize&gt;</b>	The free space size of <name_pattern>. Please refer to AT+QFLDS command for more details of <name_pattern>.
<b>&lt;filename&gt;</b>	Name of the file to be uploaded. The maximum length is 80 bytes. “<filename>” Upload the file to UFS “EUFS:<filename>” Upload the file to EUFS’s <i>ufs</i> directory “EUFS:/ufs/<filename>” Upload the file to EUFS’s <i>ufs</i> directory “EUFS:/datatx/<filename>” Upload the file to EUFS’s <i>datatx</i> directory “EUFS:/fota/<filename>” Upload the file to EUFS’s <i>fota</i> directory
<b>&lt;file_size&gt;</b>	The file size expected to be uploaded. The default value is 10240. Unit: byte.
<b>&lt;upload_size&gt;</b>	The actual size of the uploaded data. Unit: byte.
<b>&lt;timeout&gt;</b>	The time waiting for data to be inputted to USB/UART. The default value is 5. Unit: s.
<b>&lt;ackmode&gt;</b>	Whether to use ACK mode. 0 Turn off the ACK mode by default. 1 Turn on the ACK mode.
<b>&lt;checksum&gt;</b>	The checksum of the uploaded data.
<b>&lt;err&gt;</b>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

## NOTES

- 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 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.
- When executing the command, the data must be entered after “CONNECT” is returned.
- 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:
  - Run AT+QFUPL=<filename>,<file\_size>,<timeout>,1 command to enable the ACK mode.
  - BG96 outputs “CONNECT”.
  - MCU sends 1K bytes data, and then BG96 will respond with an “A”.
  - MCU receives this “A” and then sends the next 1K bytes data;
  - Repeat step 3) and 4) until the transfer is completed.

## 2.5. AT+QFDWL Download a File

The command downloads a specified file from the storage.

AT+QFDWL Download a File	
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, 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: <b>+QFDWL: &lt;download_size&gt;,&lt;checksum&gt;</b>  <b>OK</b>  If there is any error: <b>+CME ERROR: &lt;err&gt;</b>

### Parameter

<b>&lt;filename&gt;</b>	Name of the file to be downloaded. The maximum length is 80 bytes. “<filename>” Download the file from UFS “EUPS:<filename>” Download the file from EUPS’s <i>ufs</i> directory “EUPS:/ufs/<filename>” Download the file from EUPS’s <i>ufs</i> directory “EUPS:/datatx/<filename>” Download the file from EUPS’s <i>datatx</i> directory “EUPS:/fota/<filename>” Download the file from EUPS’s <i>fota</i> directory
<b>&lt;download_size&gt;</b>	The size of the downloaded data. Unit: byte.
<b>&lt;checksum&gt;</b>	The checksum of the downloaded data.
<b>&lt;err&gt;</b>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> 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.

## 2.6. AT+QFOPEN Open a File

The command can be used to open a file and get the file handle to be used in commands such as AT+QFREAD, AT+QFWRITE, AT+QFSEEK, AT+QFPOSITION, AT+QFTUCAT and AT+QCLOSE.

AT+QFOPEN Open a File	
Test Command AT+QFOPEN=?	Response <b>+QFOPEN: &lt;filename&gt;[,&lt;mode&gt;]</b>  <b>OK</b>
Read Command AT+QFOPEN?	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 AT+QFOPEN=<filename>[,<mode>]	Response <b>+QFOPEN: &lt;filehandle&gt;</b>  <b>OK</b>  If there is any error: <b>+CME ERROR: &lt;err&gt;</b>

### Parameter

<b>&lt;filename&gt;</b>	Name of the file to be opened. The maximum length is 80 bytes. <ul style="list-style-type: none"> <li>“&lt;filename&gt;”                      Open the file in UFS</li> <li>“EUPS:&lt;filename&gt;”                Open the file in EUPS’s <i>ufs</i> directory</li> <li>“EUPS:/ufs/&lt;filename&gt;”         Open the file in EUPS’s <i>ufs</i> directory</li> <li>“EUPS:/datatx/&lt;filename&gt;”      Open the file in EUPS’s <i>datatx</i> directory</li> <li>“EUPS:/fota/&lt;filename&gt;”        Open the file in EUPS’s <i>fota</i> directory</li> </ul>
<b>&lt;filehandle&gt;</b>	The handle of the file. The data type is 4 bytes.
<b>&lt;mode&gt;</b>	The open mode of the file. <ul style="list-style-type: none"> <li><u>0</u>    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.</li> <li>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.</li> <li>2    If the file exists, open it and it can be read only. When the file does not exist, it will respond an error.</li> </ul>
<b>&lt;err&gt;</b>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

## 2.7. AT+QFREAD Read a File

The command reads the 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 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. 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: <b>OK</b>  If there is any error: <b>+CME ERROR: &lt;err&gt;</b>

### Parameter

<b>&lt;filehandle&gt;</b>	The handle of the file to be operated.
<b>&lt;length&gt;</b>	The length of the file to be read out and the default is the file length. Unit: byte.
<b>&lt;read_length&gt;</b>	The actual read length. Unit: byte.
<b>&lt;err&gt;</b>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

## 2.8. AT+QFWRITE Write a File

The command is used to write data into a file. The data starts from the current position of the file pointer which belongs to the file handle.

AT+QFWRITE Write a File	
Test Command <b>AT+QFWRITE=?</b>	Response <b>+QFWRITE: &lt;filehandle&gt;[,&lt;length&gt;[,&lt;timeout&gt;]]</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. When the total size of the written

data reaches <length> (unit: byte) 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 any error:

**+CME ERROR: <err>**

## Parameter

<filehandle>	The handle of the file to be operated.
<length>	The length of the file to be written, and the default length is 10K. The maximum value of this parameter is determined by <freesize> of AT+QFUPL. Unit: byte.
<timeout>	The time waiting for data to be inputted to USB/UART. The default value is 5. Unit: s.
<written_length>	The actual written length. Unit: byte.
<total_length>	The total length of the file. Unit: byte.
<err>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

## 2.9. AT+QFSEEK Set File Pointer to a Position

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

### AT+QFSEEK Set File Pointer to a Position

Test Command <b>AT+QFSEEK=?</b>	Response <b>+QFSEEK: &lt;filehandle&gt;,&lt;offset&gt;[,&lt;position&gt;]</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 any error: <b>+CME ERROR: &lt;err&gt;</b>

## Parameter

<filehandle> The handle of the file to be operated.



<b>&lt;offset&gt;</b>	The number of bytes of the file pointer movement.
<b>&lt;position&gt;</b>	Pointer movement mode. <u>0</u> The beginning of the file. 1      The current position of the pointer. 2      The end of the file.
<b>&lt;err&gt;</b>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

#### NOTES

1. If <position> is 0 and <offset> exceeds the file size, the command will return "ERROR".
2. If <position> is 1 and the total size of <offset> and the current position of the pointer exceed the file size, the command will return "ERROR".
3. If <position> is 2, the handle will move forth.

## 2.10. AT+QFPOSITION Get the Offset of a File Pointer

The 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 any error: <b>+CME ERROR: &lt;err&gt;</b>

#### Parameter

<b>&lt;filehandle&gt;</b>	The handle of the file to be operated.
<b>&lt;offset&gt;</b>	The offset from the beginning of the file.
<b>&lt;err&gt;</b>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

## 2.11. AT+QFTUCAT Truncate a File from the File Pointer

The command will truncate all the data behind the position that the file pointer indicates.

### AT+QFTUCAT Truncate a File from the File Pointer

Test Command <b>AT+QFTUCAT=?</b>	Response <b>+QFTUCAT: &lt;filehandle&gt;</b>  <b>OK</b>
Write Command <b>AT+QFTUCAT=&lt;filehandle&gt;</b>	Response <b>OK</b>  If there is any error: <b>+CME ERROR: &lt;err&gt;</b>

#### Parameter

<b>&lt;filehandle&gt;</b>	The handle of the file to be operated.
<b>&lt;err&gt;</b>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

## 2.12. AT+QFCLOSE Close a File

The 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 open the file again 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 any error: <b>+CME ERROR: &lt;err&gt;</b>

#### Parameter

<b>&lt;filehandle&gt;</b>	The handle of the file to be operated.
---------------------------	--

**<err>** An error code relating to ME or network. Please refer to **Chapter 4** for details.

## 2.13. AT+QFCRC Calculate the CRC of a Specified UFS File

The command is used to calculate the CRC16, CRC16\_CCITT and CRC32 checksum of a specified UFS file.

### AT+QFCRC Calculate the CRC of a Specified UFS File

Test Command  
**AT+QFCRC=?**

Response  
**+QFCRC: <filename>**  
  
**OK**

Write Command  
**AT+QFCRC=<filename>**

Response  
When the specified file exists, the command returns 32 bits CRC, 16 bits CRC CCITT and 16 bits CRC values:  
**+QFCRC: <crc32>,<crc16>,<crc16\_ccitt>**  
  
**OK**  
  
When the file exists but has no byte:  
**+QFCRC: 0,0,0**  
  
If there is any error:  
**+CME ERROR: <err>**

#### Parameter

<b>&lt;filename&gt;</b>	The "filename" input format is "UFS:filename" where "UFS:" can be omitted.
<b>&lt;crc32&gt;</b>	The 32 bits CRC value.
<b>&lt;crc16&gt;</b>	The 16 bits CRC value.
<b>&lt;crc16_ccitt&gt;</b>	The 16 bits CRC CCITT value.
<b>&lt;err&gt;</b>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

#### NOTE

This command is applicable to UFS files only.

## 2.14. AT+QFCPY Make a Copy of a Specified File

The command can be used to copy the whole content of a source file to a destination file.

- When the destination file has already existed, the command will return an error unless <enable> is set into 1.
- When the destination file has the same name as the source file, the command will return an error.
- While the remaining space of the file system is smaller than the file size to be copied, the command also will return an error.

### AT+QFCPY Make a Copy of a Specified File

Test Command <b>AT+QFCPY=?</b>	Response <b>+QFCPY: &lt;src_file&gt;,&lt;dest_file&gt;[,&lt;enable&gt;]</b>  <b>OK</b>
Write Command <b>AT+QFCPY="src_file","dest_file"[,"enable"]</b>	Response <b>OK</b>  If there is any error: <b>+CME ERROR: &lt;err&gt;</b>

#### Parameter

<src_file>	The source file name. The file name input format is "UFS:filename" where "UFS:" can be omitted.
<dest_file>	The destination file name. The file name input format is "UFS:filename" where "UFS:" can be omitted.
<enable>	Determine whether to overwrite the existing file which has the same name as the destination file. 0 Disable 1 Enable
<err>	An error code relating to ME or network. Please refer to <b>Chapter 4</b> for details.

#### NOTE

This command is applicable to UFS files only.

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

OK

AT+QFUPL="EUFS:test.txt",4 //Upload the text file "test.txt" to EUFS directory.
CONNECT
<Input file bin data>
+QFUPL: 4,6a05

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, please refer to AT+QFUPL command.

```
AT+QFUPL="test.txt",3000,5,1 //Upload the text file "test.txt" to UFS.
CONNECT

<input file bin data of 1024bytes>
A //After receiving 1024 bytes data, BG96 will respond
an "A". And then the next 1024 bytes data can be
inputted.
```

```
<input file bin data of 1024bytes>  
A  
<input the rest file bin data>  
+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>  
  
+QFDWL: 10,613e //Get the bytes of the downloaded data and the checksum.  
  
OK
```

## 3.2. Write and Read Files

### 3.2.1. Write and Read a UFS File

```
AT+QFOPEN="test",0 //Open the file to get the file handle.  
+QFOPEN: 0  
  
OK  
AT+QFWRITE=0,10 //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=0,0,0 //Set the file pointer to the beginning of the file.  
OK  
AT+QFREAD=0,10 //Read 10 bytes from the file.  
CONNECT 10  
<Read Data>  
  
OK  
AT+QFCLOSE=0 //Close the file.  
OK
```

### 3.2.2. Write and Read a EUFS File

```
AT+QFLDS="EUFS" //Query the space information of EUFS.
+QFLDS: 10985472,11012096
OK

AT+QFOPEN="EUFS:test",0 //Open the file to get the file handle.
+QFOPEN: 20000

OK
AT+QFWRITE=20000,10 //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=20000,0,0 //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=20000,10 //Read 10 bytes from the file.
CONNECT 10

<Read Data>
OK
AT+QFCLOSE=20000 //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 described in the following table, and these error codes are only related to file operation of BG96 module.

**Table 1: 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 existed
409	Fail to write the file
410	Fail to open the file
411	Fail to read the file
413	Reach the max number of file allowed to be opened
414	The file read-only
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 A Reference

**Table 2: Terms and Abbreviations**

Abbreviation	Description
COM	Communication Port
DOS	Disk Operating System
EUFS	Extended User File Storage
ME	Mobile Equipment
TA	Terminal Adapter
UART	Universal Asynchronous Receiver-Transmitter
UFS	User File Storage
USB	Universal Serial Bus