EC2x&EG25-G&EG9x& EM05 FILE
AT Commands Manual

LTE Standard Module Series

Rev. EC2x&EG25-G&EG9x& EM05_FILE_AT_Commands_Manual_V1.0

Date: 2019-07-05

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.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai, China 200233
Tel: +86 21 5108 6236
Email: 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

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

About the Document

History

Revision	Date	Author	Description
1.0	2019-07-05	Wayne WEI/ Asa WANG/ Mandy WANG	Initial

Contents

Table Index

1 Introduction

Quectel EC2x&EG25-G&EG9x&EM05 modules provide 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. It is a special directory on the flash file system.
- **SD**: SD card directory.
- **RAM**: Random Access Memory. It is faster but much smaller than the UFS. And the files in RAM will be lost when rebooting the module.

The file name will indicate the storage location. When the file name begins with "RAM:", it means that the file is stored in RAM. When the file name begins with "SD:", it means that the file is stored in SD card. If there are no prefix characters in the file name, then the file is stored in UFS.

NOTE

Storage medium **SD** is not supported on EG9x module.

1.1. Applicable Modules

This document is applicable to the following Quectel modules:

- EC2x: EC25/EC21/EC20 R2.0/EC20 R2.1
- EG25-G
- EG9x: EG91/EG95
- EM05

1.2. The Process of Using File AT Commands

The following procedures can be followed 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 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 the file is opened by **AT+QFOPEN**, it can be set as overwrite mode, read-only mode or other modes by the parameter **<mode>**. (For more information about **<mode>**, please refer to *Chapter 2.6*). After the file is opened, a parameter **<filehandle>** is assigned to it. Then the file can be operated by **<filehandle>**.

    ●   After the file is opened, it can be written by **AT+QFWRITE** and can be read by **AT+QFREAD** from the current file position.

    ●   The file position can be set by **AT+QFSEEK** and the current file position can be queried by **AT+QFPOSITION**.

    ●   **AT+QFTUCAT** is used to truncate a file from the file pointer.

    ●   **AT+QFCLOSE** is used to close the file, after which the **<filehandle>** becomes meaningless any more.

There are several commands to manage files in the storage:

1. **AT+QFLDS**: get the space information of the storage.
2. **AT+QFLST**: list the file information in the storage.
3. **AT+QFDEL**: delete the file(s) in the storage.

1.3. Description of Data Mode

The COM port of EC2x/EG25-G/EG9x/EM05 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 the Storage

The command gets the space information of the specified storage.

AT+QFLDS Get the Space Information of the Storage	
Test Command **AT+QFLDS=?**	Response **OK**
Write Command **AT+QFLDS=<name_pattern>**	Response **+QFLDS: <free_size>,<total_size>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+QFLDS**	Response Return the UFS information **+QFLDS: <UFS_file_size>,<UFS_file_number>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**

Parameter

<name_pattern>	Storage pattern. "UFS" UFS "RAM" RAM "SD" SD card
<free_size>	The free space size of **<name_pattern>**.
<total_size>	The total size of **<name_pattern>**.
<UFS_file_size>	The size of all files in UFS. Unit: byte.
<UFS_file_number>	The number of files in UFS.
<err>	The code of an error relating to ME or network. Please refer to **Chapter 4**

for details.

Example

AT+QFLDS="RAM"	//Query the space information of RAM.
+QFLDS: 2616320,2621440	
OK	
AT+QFLDS="UFS"	//Query the space information of UFS.
+QFLDS: 49180672,49205248	
OK	
AT+QFLDS="SD"	//Query the space information of SD card.
+QFLDS: 251920384,253132800	
OK	

2.2. AT+QFLST   List the File Information in the Storage

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

AT+QFLST List the File Information in the Storage	
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>

Parameter

<name_pattern>	The file to be listed.	
	"*"	All the files in UFS.
	"**<filename>**"	The specified file **<filename>** in UFS.
	"RAM:*"	All the files in RAM.
	"RAM:**<filename>**"	The specified file **<filename>** in RAM.
	"SD:*"	All the files in SD card.
	"SD:**<filename>**"	The specified file **<filename>** in SD card.
<filename>	Name of the file.	
<file_size>	Size of the file. Unit: byte.	
<err>	The code of an error relating to ME or network. Please refer to ***Chapter 4*** for details.	

Example

AT+QFLST="*"	//List all the file in UFS.
+QFLST: "UFS:F_M12-1.bmp",562554	
+QFLST: "UFS:F_M12-10.bmp",562554	
+QFLST: "UFS:F_M12-11.bmp",562554	
OK	
AT+QFLST="RAM:*"	//List all the files in RAM.
+QFLST: "RAM:F_EM05-1.bmp",56554	
+QFLST: "RAM:F_EM05-10.bmp",56554	
+QFLST: "RAM:F_EM05-11.bmp",56554	
OK	
AT+QFLST="SD:*"	//List all the files in SD card.
+QFLST: "SD:10b.txt",10	
+QFLST: "SD:50K.txt",51209	
+QFLST: "SD:94k.jpg",95811	
OK	

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

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

AT+QFDEL Delete the File(s) in the Storage	
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>**

Parameter

\<filename>	Name of the file to be deleted. The max length is 80 bytes.
	"*" Delete all the files in UFS (not delete the directory).
	"**\<filename>**" Delete the specified file **\<filename>** in UFS.
	"RAM:*" Delete all the files in RAM.
	"RAM:**\<filename>**" Delete the specified file **\<filename>** in RAM.
	"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 or network. Please refer to ***Chapter 4*** for details.

Example

```
AT+QFDEL="*"
OK
AT+QFLST="RAM:*"
OK
AT+QFLST="SD:*"
OK
```

2.4. AT+QFUPL   Upload a File to the Storage

The command uploads a file to storage. If there is any file in the storage which has the same name with the file to be 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 to the Storage	
Test Command **AT+QFUPL=?**	Response **+QFUPL: <filename>[,(1-<freesize>)[,(1-65535)[,(0,1)]]]** **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>**

Parameter

<freesize>	The free space size of **<name_pattern>**. Please refer to **AT+QFLDS** command for more details of **<name_pattern>**.
<filename>	Name of the file to be uploaded. The maximum length is 80 bytes. "**<filename>**"　　　Name of the file to be uploaded to UFS. "RAM:**<filename>**"　Name of the file to be uploaded to RAM. "SD:**<filename>**"　Name of the file to be uploaded to SD card.
<file_size>	The file size expected to be uploaded. The default value is 10240. Unit: byte.
<upload_size>	The actual size of the uploaded data. Unit: byte.
<timeout>	The time waiting for data to be inputted to USB/UART. The default value is 5. Unit: s.
<ackmode>	Whether to use ACK mode. 0　Turn off the ACK mode by default. 1　Turn on the ACK mode.
<checksum>	The checksum of the uploaded data.
<err>	The code of an error relating to ME or network. Please refer to *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 entered 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 1K bytes data, and then module will respond with an "A".
   4) MCU receives this "A" and then sends the next 1K bytes data;
   5) Repeat step 3) and 4) until the transfer is completed.

2.5. AT+QFDWL   Download a File from the Storage

The command downloads a specified file from the storage.

AT+QFDWL Download a File from the Storage	
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>**

Parameter

<filename>	Name of the file to be downloaded. The maximum length is 80 bytes.
	"**<filename>**"　　　　　Name of the UFS file to be downloaded.

	"RAM: **<filename>**"	Name of the RAM file to be downloaded.
	"SD: **<filename>**"	Name of the SD file to be downloaded.
<download_size>	The size of the downloaded data.	
<checksum>	The checksum of the downloaded data.	
<err>	The code of an error relating to ME or network. Please refer to ***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.

2.6. AT+QFOPEN   Open a File

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

AT+QFOPEN Open a File	
Test Command **AT+QFOPEN=?**	Response **+QFOPEN: <filename>[,(0-2)]** **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>**

Parameter

<filename>	Name of the file to be opened. The maximum length is 80 bytes.	
	"**<filename>**"	Name of the UFS file to be opened.
	"RAM: **<filename>**"	Name of the RAM file to be opened.

	"SD: **<filename>**" Name of the SD file to be opened.
<filehandle>	The handle of the file. The data type is 4 bytes.
<mode>	The open mode of the file.
	<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.
	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 or network. Please refer to *Chapter 4* 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 **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>**

Parameter

<filehandle>	The handle of the file to be operated.
<length>	The length of the file to be read out and the default is the file length.
<read_length>	The actual read length.
<err>	The code of an error relating to ME or network. Please refer to *Chapter 4* for details.

2.8. AT+QFWRITE Write a File

The 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 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>** (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 an error related to ME functionality: **+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**.
<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.
<total_length>	The total length of the file.
<err>	The code of an error relating to ME or network. Please refer to *Chapter 4* for details.

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

The command sets a file pointer to the 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 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>

Parameter

<filehandle>	The handle of the file to be operated.
<offset>	The number of bytes of the file pointer movement.
<position>	Pointer movement mode.
	<u>0</u> The beginning of the file.
	1 The current position of the pointer.
	2 The end of the file.
<err>	The code of an error relating to ME or network. Please refer to **Chapter 4** 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 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>

Parameter

<filehandle>	The handle of the file to be operated.
<offset>	The offset from the beginning of the file.
<err>	The code of an error relating to ME or network. Please refer to *Chapter 4* for details.

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

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

AT+QFTUCAT Truncate a File from the File Pointer	
Test Command **AT+QFTUCAT=?**	Response **+QFTUCAT: <filehandle>** **OK**
Write Command **AT+QFTUCAT=<filehandle>**	Response **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**

Parameter

<filehandle>	The handle of the file to be operated.
<err>	The code of an error relating to ME or network. Please refer to *Chapter 4* 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 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>**

Parameter

<filehandle>	The handle of the file to be operated.
<err>	The code of an error relating to ME or network. Please refer to ***Chapter 4*** for details.

2.13. AT+QFMOV   Move a File

The command moves a file from one storage to another. It only supports moving the file from UFS or SD to RAM, not from RAM to UFS or SD.

AT+QFMOV Move a File	
Test Command **AT+QFMOV=?**	Response **+QFMOV: <srcfilename>,<destfilename>,(0,1),(0,1)** **OK**
Write Command **AT+QFMOV=<srcfilename>,<destfilename>,<copy>,<overwrite>**	Response **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**

Parameter

<srcfilename>	The source file. The max length is 80 bytes.
	"**<srcfilename>**" The source file named **<srcfilename>** in the UFS.
	"SD:**<srcfilename>**" The source file named **<srcfilename>** in the SD.
<destfilename>	The destination file. The max length is 80 bytes.

	"RAM:**<destfilename>**"	The destination file named **<destfilename>** in the RAM.
<copy>	Whether or not to delete the source file after the file is moved.	
	0	Delete the source file after it is moved, i.e., cut this file.
	1	Do not delete the source file after it is moved, i.e., copy this file.
<overwrite>	Whether or not to overwrite the file if the destination file exists.	
	0	Do not overwrite the destination file if it exists.
	1	Overwrite the destination file if it exists.
<err>	The code of an error relating to ME or network. Please refer to *Chapter 4* for details.	

NOTE

When executing **AT+QFMOV="filenamea","RAM:flienameb",<copy>,<overwrite>**, it means moving a single file, and users can move the file named "filenamea" in UFS to RAM, and rename it as "filenameb".

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="RAM:test1.txt",10   //Upload the text file *RAM:test1.txt* to RAM.
CONNECT
<Input file bin data>
+QFUPL: 10,6466   //Get the bytes and the checksum value of the uploaded data.
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="RAM:test.txt",3000,5,1   //Upload the text file *RAM:test.txt* to RAM.
CONNECT
<input file bin data of 1024bytes>
A   //After receiving 1024 bytes data, the module 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="RAM:test.txt"   //Download the text file *RAM:test.txt* from RAM.
CONNECT
<Output Data>
+QFDWL: 10,613e   //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="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 RAM File

AT+QFLDS="RAM"   //Query the space information of RAM.
+QFLDS: 2606080,2621440

OK

```
AT+QFOPEN="RAM:1",0          //Open the file in the RAM.
+QFOPEN: 3000

OK
AT+QFWRITE=3000,10           //Write 10 bytes to the file.
CONNECT
<Write Data>
+QFWRITE: 10,10              //The actual written bytes and the size of the file are returned.

OK
AT+QFSEEK=3000,0,0           //Set the file pointer to the beginning of the file.
OK
AT+QFREAD=3000,10            //Read the data.
CONNECT
<Read Data>

OK
AT+QFCLOSE=3000              //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 EM05 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 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 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
ACK	Acknowledgement
COM	Communication Port
DOS	Disk Operating System
DTR	Data Terminal Ready
ME	Mobile Equipment
RAM	Random Access Memory
SD	Secure Digital
TA	Terminal Adapter
UART	Universal Asynchronous Receiver-Transmitter
UFS	User File Storage
USB	Universal Serial Bus
XOR	Exclusive OR