

GSM FILE

AT Commands Manual

GSM_FILE _AT_Commands_Manual_V1.5

Contents

Contents	2
0. Revision history	3
1. Introduction.....	4
1.1. Reference.....	4
1.2. Terms and abbreviations.....	4
2. AT Commands for File	5
2.1. Overview of AT commands for File	5
2.2. Detailed descriptions of commands	5
2.2.1. AT+QFLDS Get data storage size	5
2.2.2. AT+QFLST List files	6
2.2.3. AT+QFUPL Upload file to storage	7
2.2.4. AT+QFDWL Download file from storage	8
2.2.5. AT+QFDEL Delete file in storage	9
2.2.6. AT+QFMOV Move file	10
2.2.7. AT+QFOPEN Open file.....	10
2.2.8. AT+QFREAD Read file.....	12
2.2.9. AT+QFWRITE Write file	12
2.2.10. AT+QFSEEK Seek file	13
2.2.11. AT+QFCLOSE Close file	13
2.2.12. AT+QFPOSITION Get offset of the file pointer	14
2.2.13. AT+QFFLUSH Force to write data remaining in the file buffer.....	14
2.2.14. AT+QFTUCAT Truncate the specified file from the file pointer.....	14
3. Summary of error codes	16
4. Reliable transmission	17
4.1. Calculated checksum to check the file transfer	17
4.2. ACK mode is enabled to stabilize Uploading	18
5. Examples	19
5.1. File uploading and downloading	19
5.2. File moving	19
5.2.1. Move single file	20
5.2.2. Move all files of one storage.....	20
5.3. Read/write file	21

2. AT Commands for File

Module provides AT commands which can operate files in RAM, flash and SD card. And the SD card only supports three file systems: FAT, FAT16 and FAT32.

2.1. Overview of AT commands for File

Command	Description
AT+QFLDS	Get storage data size
AT+QFLST	List files
AT+QFUPL	Upload file to storage
AT+QFDWL	Download file from storage
AT+QFDEL	Delete file in storage
AT+QFMOV	Move file
AT+QFOPEN	Open file
AT+QFREAD	Read file
AT+QFWRITE	Write file
AT+QFSEEK	Seek file
AT+QFCLOSE	Close file
AT+QFPOSITION	Get offset of the file pointer
AT+QFFLUSH	Force to write data remaining in the file buffer
AT+QFTUCAT	Truncate the specified file from the file pointer

2.2. Detailed descriptions of commands

2.2.1. AT+QFLDS Get data storage size

AT+QFLDS Get data storage size	
Test Command AT+QFLDS=?	Response OK
Write Command AT+QFLDS=<namepattern> >	Response +QFLDS: <free size>,<total size>[,<maxalloc size>] OK
	Parameter <namepattern> Pattern "UFS" UFS file in flash "RAM" RAM file

	<p>“SD” SD file</p> <p><free size> Free data size in <namepattern></p> <p><total size> Total data size in <namepattern></p> <p><maxalloc size> The maximum size which can be allocated, only valid for RAM file.</p>
Execution Command AT+QFLDS	<p>Response</p> <p>+QFLDS: <ufs file size>,<ufs file number></p> <p>OK</p> <p>Returns the UFS information</p> <p>Parameter</p> <p><ufs file size> The size in bytes of all files in UFS</p> <p><ufs file number> The number of files in UFS</p>
Reference	

2.2.2. AT+QFLST List files

AT+QFLST List files	
Test Command AT+QFLST=?	<p>Response</p> <p>OK</p>
Write Command AT+QFLST=<namepattern> >	<p>Response</p> <p>+QFLST: <file name>,<file size>[,<ram size>]</p> <p>OK</p> <p>Parameter</p> <p>< namepattern > Pattern for filename</p> <p>“*” All UFS file in flash</p> <p>“RAM:*” All RAM file</p> <p>“SD:*” All SD file</p> <p>“filename” Name of UFS file</p> <p>“RAM:filename” Name of RAM file</p> <p>“SD:filename” Name of SD file</p> <p><file name> Name of the file</p> <p><file size> Size in bytes of the file</p> <p><ram size> Memory size allocated for the file in RAM, only valid for RAM file</p>
Execution Command AT+QFLST	<p>Response</p> <p>+QFLST: <file name>,<file size></p> <p>[+QFLST: <file name>,<file size> [...]]</p> <p>OK</p>

	List files in the UFS directory
	Parameter <file name> Name of the file <file size> Size in bytes of the file
Reference	<i>Note:</i> <i>Only list files in the “Picture” directory of SD card, do not list any directory and any file in the other directories.</i>

2.2.3. AT+QFUPL Upload file to storage

AT+QFUPL Upload file to storage	
Test Command AT+QFUPL=?	Response +QFUPL: "file name",<filesize>,(1-65535),(0,1) OK
	Parameter See Write Command.
Write Command AT+QFUPL=<file name>[,<filesize>[,<timeout>[,<ackmode>]]]	Response CONNECT TA switches to data mode, and the bin data of file can be inputted. When the total size of the input data reaches <file size> (unit: byte) or TA receives “+++” sequence from UART, TA returns to command mode and replies the following codes. +QFUPL: <upload size>,<checksum> OK
	Parameter <file name> The name of the file to be stored. <filesize> The maximum size of the file to be uploaded. Default is 10240. Unit: byte <upload size> The size of the actually uploaded data. Unit: byte <timeout> The time in seconds to wait for data input from UART. Default is 5. <ackmode> Whether to use acknowledge mode or not 0 Turn off the ACT mode. It's default. 1 Turn on the ACT mode <checksum> The checksum of the uploaded data
Reference	<i>Note:</i> <ul style="list-style-type: none"> ● It is strongly recommended to use DOS 8.3 file name format for <file name>. ● <file name>

	<p>“filename” File is uploaded to the UFS directory</p> <p>“RAM:filename” File is uploaded to RAM</p> <p>“SD:filename” File is uploaded to Picture directory in SD card</p> <ul style="list-style-type: none"> ● <checksum> is 16 bit checksum based on bitwise XOR. ● “+++” sequence will cause TA to end the command and switch to command mode; however, data previously entered are still being preserved as the data of the file. ● To execute the command, must appear "CONNECT" before entering the binary data. ● The ACT mode is provided to avoid loss of data when uploading large files in case hardware flow control is not available. The ACT mode works as follows: <ul style="list-style-type: none"> 1) Run command AT+QFUP = "file name", filesize, 5,1 to enable the ACK mode; 2) The module outputs "CONNECT"; 3) MCU sends 1Kbytes data, and then the module will respond with a 'A'; 4) MCU receives the 'A' and then sends the next 1Kbytes data 5) Repeat step 3) and 4) until the transfer is completed.

2.2.4. AT+QFDWL Download file from storage

AT+QFDWL Download file from storage	
Test Command AT+QFDWL=?	Response +QFDWL: "file name" OK
	Parameter See Write Command.
Write Command AT+QFDWL=<file name>	Response CONNECT TA switches to data mode, and the bin data of the file will be outputted. When the file was read over, TA returns to command mode and replies the following codes: +QFDWL: <download size>,<checksum> OK
	Parameter <file name> The name of the file to be downloaded

	<download size> The size of the downloaded data <checksum> The checksum of the downloaded data
Reference	<p><i>Note:</i></p> <ul style="list-style-type: none"> ● <file name> “filename” File is uploaded to the UFS directory “RAM:filename” File is uploaded to RAM (only M33 supports) “SD:filename” File is uploaded to Picture directory in SD card ● “+++” sequence will cause TA to end the command and switch to command mode. ● <checksum> is 16 bit checksum based on bitwise XOR.

2.2.5. AT+QFDEL Delete file in storage

AT+QFDEL Delete file in storage	
Test Command AT+QFDEL=?	Response +QFDEL: "file name" OK
	Parameter See Write Command.
Write Command AT+QFDEL=<file name>	Response OK
	Parameter <file name> The name of the file to be deleted “*” Delete all files in UFS directory (DO not delete the directory) “RAM:” Delete all files in RAM “SD:” Delete all files in Picture directory of SD card “filename” Delete the specified file “filename” in UFS directory “RAM:filename” Delete the specified file “filename” in RAM “SD:filename” Delete the specified file “filename” in Picture directory of SD card
	<p><i>Note:</i></p> <p><i>Only delete files in the Picture directory of SD card, do not delete any directory or any file in the other directories.</i></p>

2.2.6. AT+QFMOV Move file

AT+QFMOV Move file	
Test Command AT+QFMOV=?	Response +QFMOV: "src filename","dest filename",(0,1),(0,1)
	OK
Write Command AT+QFMOV=<src filename>,<dest filename>,<copy>,<overwrite>	Parameter See Write Command.
	Response OK Parameter <src filename> Source file <dest filename> Destination file <copy> Whether or not to delete source file after the file is copied 0 Delete source file after file is copied 1 Do not delete source file after file is copied <overwrite> Whether or not to overwrite existed destination file 0 Do not overwrite the destination file if it exists 1 Overwrite the destination file if it exists
Reference	Note: <ul style="list-style-type: none"> ● This command is supported only in M33. ● Cannot move file from UFS or SD card to RAM. ● AT+QFMOV="RAM:*","SD:*",1,1 Move all files in RAM to SD card ● AT+QFMOV="RAM:filenamea","SD:filenameb",1,1 Move the file named "filenamea" in RAM to SD card, and rename it as "filenameb".

2.2.7. AT+QFOPEN Open file

AT+QFOPEN Open file	
Test Command AT+QFOPEN=?	Response +QFOPEN: "filename"[(0-2) [<length>]]
	OK
Read Command	Parameter See Write Command.
	Response

AT+QFOPEN?	<p>+QFOPEN: "filename",<filehandle>,<mode></p> <p>[+QFOPEN: "filename",<filehandle>,<mode> [...]]</p> <p>OK</p>
	<p>Parameter</p> <p>See Write Command.</p>
<p>Write Command</p> <p>AT+QFOPEN=<filename>[, <mode>[,length]]</p>	<p>Response</p> <p>+QFOPEN: <filehandle></p> <p>OK</p> <p>Parameter</p> <p><filename> The file needed to be operated</p> <p><mode> The mode of the file opened, default is 0</p> <p> 0 (Default) If the file doesn't exist, it will be created; if the file exists, it will be opened. And both of them can be read and written</p> <p> 1 If the file exists, it will be created and clear the old file</p> <p> 2 If the file exists, open it and it only can be read.</p> <p><length> The max length of the file. Default is 10240. Unit: byte. It is only used for RAM file.Ignore this parameter if you use UFS or SD file.</p>
Reference	<p>Note:</p> <ul style="list-style-type: none"> ● Can open file from UFS、RAM or SD card ● AT+QFOPEN="filenameb", 0 Open the file named "filenameb" in the UFS ● AT+QFOPEN="SD:filenameb",0 Open the file named "filenameb" in the Picture directory of SD card ● AT+QFOPEN="RAM:filename",0,1024 Open the file named <filename> in RAM. If the file does not exist, then create it, and set the max length of the file with value 1024. If the file exists in RAM, just open it, and ignore the length 1024. ● AT+QFOPEN="RAM:filename",1,10240 Create the file named <filename> in RAM. If the file already exists, first delete it, then create it, and set the max length of the file with value 10240.

2.2.8. AT+QFREAD Read file

AT+QFREAD Read file	
Test Command AT+QFREAD=?	Response +QFREAD: <filehandle>[,<length>]
	OK
Write Command AT+QFREAD=<filehandle>[,<length>]	Parameter See Write Command.
	Response CONNECT < read length > TA switches to data mode. When the total size of the read data reaches <read length> (unit: byte), TA returns to command mode and then replies the following codes. OK
Reference	Parameter <filehandle> The handle of the file needed to be operated <length> The length of the file to be read out, the default is the size of the file. <read length> The actual length to be read out

2.2.9. AT+QFWRITE Write file

AT+QFWRITE Write file	
Test Command AT+QFWRITE=?	Response +QFWRITE: <filehandle>[,<length>[,<timeout>]]
	OK
Write Command AT+QFWRITE=<filehandle> > [,<length>[,<timeout>]]	Parameter See Write Command.
	Response CONNECT TA switches to data mode. When the total size of the written data reaches <length> (unit: byte) or it is timeout, TA returns to command mode and replies the following codes. +QFWRITE: <written length>,<total length> OK
Reference	Parameter <filehandle> The handle of the file needed to be operated

	<length> The length of the file needed to be written, the default length is 10K <timeout> The time in seconds to wait for data input from UART. Default is 5. <written length> The actual length to be written <total length> The total length of the file
Reference	

2.2.10. AT+QFSEEK Seek file

AT+QFSEEK Seek file	
Test Command AT+QFSEEK=?	Response +QFSEEK: <filehandle>,<offset>[,<position>] OK
	Parameter See Write Command.
Write Command AT+QFSEEK=<filehandle>,<offset>[,<position>]	Response OK
	Parameter <filehandle> The handle of the file needed to be operated <offset> Number of bytes to move the file pointer <position> Pointer movement mode. The default is 0 0 File begin 1 Current position of the pointer 2 File end
Reference	

2.2.11. AT+QFCLOSE Close file

AT+QFCLOSE Close file	
Test Command AT+QFCLOSE=?	Response +QFCLOSE: <filehandle> OK
	Parameter See Write Command.
Write Command AT+QFCLOSE=<filehandle> >	Response OK
	Parameter <filehandle> The handle of the file needed to be operated
Reference	

2.2.12. AT+QFPOSITION Get offset of the file pointer

AT+QFPOSITION Get offset of the file pointer	
Test Command AT+QFPOSITION=?	Response +QFPOSITION: <filehandle>
	OK
	Parameter See Write Command.
Write Command AT+QFPOSITION= <filehandle>	Response +QFPOSITION: <offset>
	OK
	Parameter <filehandle> The handle of the file needed to be operated <offset> The offset from the beginning of the file to the current position
Reference	

2.2.13. AT+QFFLUSH Force to write data remaining in the file buffer

AT+QFFLUSH Force to write data remaining in the file buffer	
Test Command AT+QFFLUSH=?	Response +QFFLUSH: <filehandle>
	OK
	Parameter See Write Command.
Write Command AT+QFFLUSH=<filehandle >	Response OK
	Parameter <filehandle> The handle of the file needed to be operated
Reference	

2.2.14. AT+QFTUCAT Truncate the specified file from the file pointer

AT+QFTUCAT Truncate the specified file from the file pointer	
Test Command AT+QFTUCAT=?	Response +QFTUCAT: <filehandle>
	OK
	Parameter

	See Write Command.
Write Command AT+QFTUCAT=<filehandle> >	Response OK
	Parameter <filehandle> The handle of the file needed to be operated
Reference	

3. Summary of error codes

Final result code **+CME ERROR: <err>** indicates an error related to mobile equipment or network. The operation is similar to **ERROR** result code. Neither **ERROR** nor **OK** result code shall be returned. The listed **<err>** codes here are just related with File. About other **<err>** codes, please refer to document [1].

Code of <err>	Meaning
3765	Invalid input value
3915	Non-existent address
3916	UFS storage full
3917	Drive full
3918	Drive error
3919	File not found
3920	Invalid file name
3921	File already existed
3922	Failed to create file
3923	Failed to write file
3924	Failed to open file
3925	Failed to read file
4000	Exceed max length
4001	Open file fail
4002	Write file fail
4003	Get size fail
4004	Read fail
4005	List file fail
4006	Delete file fail
4007	Get Disk info fail
4008	No space
4009	Time out
4010	File not found
4011	File too large
4012	File already exist
4013	Invalid parameter
4014	Driver error
4015	Create fail
4016	Access denied

4. Reliable transmission

4.1. Calculated checksum to check the file transfer

For reliable transmission, when using "AT+QFUPL" and "AT+QFDWL" to upload and download file, it is recommended that users turn on hardware flow control capabilities, while also open MCU hardware flow control function. Open the hardware flow control function for the module via the **AT+IFC=2,2<CRLF>** command, which is enabled by default.

As general serial transmission is reliable, in order to further reliability, we offer additional ways to verify the data transmission reliability by the command's response information.

When using "AT+QFUPL=<file name> [,<file size>]" command to upload a file, the module will report "+QFUPL: <upload size>, <checksum>" information tips at the end of data transmission. Then MCU can judge whether the data has lost by comparing value of <upload size> with <checksum>.

<upload size> is the data length which the module received. MCU compares the <upload size> with the actual length of the file. If unequal, it means the module lost data.

<checksum> is calculated by doing XOR for every 2 bytes. Similarly MCU calculates the actual file's checksum as below example, and then compares this value with <checksum> which module reports. If not equal, the received data may be problematic. User can re-upload the data.

Example for calculating checksum:

If the uploaded file data length is 9, the 16 hex values are as follows:

0x23 0x13 0x65 0x B6 0x76 0x88 0xA3 0xEF 0x55

So, checksum is calculated as follows:

checksum = 0x2313 XOR 0x65B6 XOR 0x7688 XOR 0xA3EF XOR 0x5500

Every two data form a group and do XOR with another group. If the last group is less than 2 bytes, supplement with 0x00.

Similarly, the module will report the "+QFDWL: <download size>,<checksum>" information when command "AT+QFDWL=<file name>" is completed. <download size> is the actual size of downloaded data, MCU can calculate received data length, and compare it with <download size>. If not equal, the data is lost. MCU also can do checksum calculation and comparison with <checksum>. If not equal, it is needed to re-download the file.

4.2. ACK mode is enabled to stabilize Uploading

When it is needed to use the AT + QFUPL command to upload large files, and UART hardware flow control is turned off, it is recommended to use the ACK Mode.

The ACT mode works as follows:

- 1) Run command AT+QFUPL = "file name", filesize, 5,1 to enable the ACK mode;
- 2) The module outputs "CONNECT";
- 3) MCU sends 1Kbytes data, and then the module will respond with an 'A';
- 4) MCU receives the 'A' and then sends the next 1Kbytes data;
- 5) Repeat step 3) and 4) until the transfer is completed.

For example:

```
AT+QFUPL="test.txt",3000           // Upload the text file "test.txt" to UFS
CONNECT
<input file bin data of 1024bytes>
A                                   // After receiving 1024bytes data, the module
                                   will respond with an "A", then next 1024 bytes
                                   data can be input
<input file bin data of 1024bytes>
A
<input the rest file bin data>
+QFUPL: 3000,B34A
OK
```


5. Examples

5.1. File uploading and downloading

```
AT+QFUPL="test.txt",3222           // Upload the text file "test.txt" to UFS
CONNECT
<input file bin data>
+QFUPL: 3222,B3E4

OK

AT+QFDWL="test.txt"                // Download the file "test.txt" from UFS
CONNECT
<output file bin data>
+QFDWL: 3222,B3E4

OK

AT+QFUPL="RAM:test2.txt",4222       // Upload the text file "test2.txt" to RAM
CONNECT
<input file bin data>
+QFUPL: 4222,13E4

OK

AT+QFDWL="SD:pic1.jpg",13222        // Download the picture file "pic1.jpg" from
CONNECT                               SD card
<input file bin data>
+QFDWL: 13222,D5E4

OK
```

5.2. File moving

User can move file(s) among RAM, UFS and SD card by command "AT+QFMOV". Please note that the file cannot be moved from UFS or SD card to RAM. Here list examples of moving single file and all files.

5.2.1. Move single file

```
AT+QFLST="RAM:*"           //RAM has file "Pic.jpg"
+QFLST: "RAM:Pic.jpg",63388,75000

OK
AT+QFLST="*"               // UFS also has one file "pic.jpg", but the file size is
                             different
+QFLST: "pic.jpg",62076

OK
AT+QFMOV="RAM:pic.jpg","pic.jpg",1,0
+CME ERROR: 3921           // Move file "pic.jpg" from RAM to UFS. "1" means
                             NOT deleting source file after file is copied, "0"
                             means Do not overwrite the destination file if it
                             exists. Because the destination file has existed,
                             so it responds ERROR 3921 (File already existed).
                             The file is not moved. If user confirms the file can be
                             overwritten, set the last parameter as "1" as below.

AT+QFMOV="RAM:pic.jpg","pic.jpg",1,1 //Move file successfully. The source file
                                         "pic.jpg" in RAM is not deleted. The
                                         destination file "pic.jpg" in UFS has been
                                         overwritten.

OK
AT+QFLST="RAM:*"
+QFLST: "RAM:Pic.jpg",63388,75000

OK
AT+QFLST="*"
+QFLST: "pic.jpg",63388

OK
```

5.2.2. Move all files of one storage

```
AT+QFLST="RAM:*"
+QFLST: "RAM:pic0.jpg",59024,75000

+QFLST: "RAM:pic1.jpg",62592,75000

+QFLST: "RAM:pic2.jpg",57168,75000

+QFLST: "RAM:pic3.jpg",63216,75000
```

```

+QFLST: "RAM:pic4.jpg",64600,75000

+QFLST: "RAM:pic5.jpg",60284,75000

OK
AT+QFLST="SD:*"
OK
AT+QFMOV="RAM:*","SD:*",0,0
OK                                     // Move all files in RAM to SD, the first "0" means
                                     deleting source files after files are copied, the
                                     second "0" means Do not overwrite the destination
                                     file if it exists. SD has not these files, so the last
                                     parameter is meaningless in this case.

AT+QFLST="RAM:*"
OK                                     // Source files are deleted.
AT+QFLST="SD:*"
+QFLST: "SD:pic0.jpg",59024
                                     // All files in RAM have been moved to SD card.
+QFLST: "SD:pic1.jpg",62592
+QFLST: "SD:pic2.jpg",57168
+QFLST: "SD:pic3.jpg",63216
+QFLST: "SD:pic4.jpg",64600
+QFLST: "SD:pic5.jpg",60284

OK

```

5.3. Read/write file

```

AT+QFOPEN="test.txt", 0               // Open file test.txt
+QFOPEN: 12451840

OK

AT+QFWRITE=12451840, 10               // Set write length
CONNECT
<input data>                          // Input data
+QFWRITE: 10, 10

OK

```

AT+QFSEEK=12451840, 0, 0

// Seek to the origination of the file

OK

AT+QFREAD=12451840

// Read the test.txt

CONNECT 10

// The data of 10 bytes will be read out.

<output data>

OK