Fibocom

PERFECT WIRELESS EXPERIENCE

MTC

Application Guide_HTTP

V1.0

Disclaimer

Customers shall design and develop their products according to the information provided in the document. Fibocom shall not be liable for any damages caused by failure to comply with the relevant operations, specifications or rules. Due to product version update or other reasons, Fibocom reserves the rights to modify any information in this document at any time without prior notice and without any responsibility. Unless otherwise specified, all the statements, information and suggestions contained in the document do not constitute any explicit or implicit guarantee.

Copyright Statement

Unless specifically authorized by Fibocom, the recipient of the document shall keep the received documents and information confidential and shall not use it for any purpose other than the implementation and development of this project. Without the prior written permission of the copyright holder, any company or individual is prohibited to excerpt, copy any part of or the entire document, or distribute the document in any form. For any violation of confidentiality obligations, unauthorized use or malicious use of the document and information in other illegal forms, Fibocom has the rights to pursue legal responsibility.

Trademark Statement

Fibocom The trademark is registered and owned by Fibocom Wireless Inc.

Other trademarks, product names, service names, and company names appearing in this document are the properties of their respective owners.

Contact

Website: https://www.fibocom.com

Address: Floor 10-14, Block A, Building 6, Shenzhen International Innovation Valley, Dashiyi Road, Xili Community, Xili Street, Nanshan District, Shenzhen

Tel: 0755-26733555

Safety Instruction

Do not operate wireless communication products in areas where use of radio is not recommended without proper equipment certification. These areas include environment where radio interference may occur, such as flammable and explosive environment, medical equipment, aircraft, or any other equipment that may be subject to any form of radio interference.

Any driver of a vehicle must not operate a wireless communication product while controlling the vehicle. Doing so will reduce the driver's control and operation of the vehicle, posing a safety risk.

The wireless communication product does not guarantee a valid connection under any circumstances, for example, when the (U)SIM is in arrears or invalid. In case of emergency, use the emergency call function in power-on state, and make sure the equipment is in an area with sufficient signal strength.

Contents

Applicable Model

No.	Applicability Model	Description
1	All MTC products	/

Change History

V1.0 (2024-5-9)	Initial version

1 About This Document

The module integrates the standard HTTP/IP protocol, and the MCU implements the HTTP/IP transmission function by sending AT commands to the module. To implement TLS/DTLS client services, you can use SSL commands.

2 Reference Document

Refer to HTTP, SSL, and universal AT command manuals of the corresponding platform.

3 HTTP Overview

Before using the HTTP function, you must power on the module and check the module network, and ensure that the module can access the network normally.

3.1 Internal Dial-up

The module needs to initiate internal dial-up before performing network services. Internal dial-up uses the MIPCALL command. The application scenarios mainly include the following:

1. AT+MIPCALL=<operation>

The activated <cid> is 1. When performing dial-up, check whether the dial-up type is IPv4, IPv6, or IPv4v6 or is disconnected according to the <operation> type. After dial-up, the <PDP_type> parameter in CGDCONT will be modified to the corresponding IP type.

2. AT+MIPCALL=<operation>,<cid>

At this time, <operation> only supports 0 and 1, where 0 indicates to disconnect the dial-up connection and 1 indicates to perform dial-up.

- <cid> is 1

When **AT+MIPCALL=1,1**, the IP type is IPv4 upon dial-up. After dial-up, the <PDP_type> parameter in CGDCONT will be modified to the corresponding IP type.

When AT+MIPCALL=0,1, the dial-up connection is disconnected.

- <cid> is not 1

IP types are not differentiated. At this time, the IP type is determined by <PDP_type> configured in AT+CGDCONT=<cid>,<PDP_type>,<APN> or AT+CGDCONT=<cid>,<PDP_type>.

3. AT+MIPCALL=<operation>,<APN>

When performing dial-up, check whether the dial-up type is IPv4, IPv6, or IPv4v6 or is disconnected according to the <operation> type. At this time, the APN modifies the <APN> configuration in CGDCONT.

1. The above describes the case where the internal protocol stack is not enabled.
2. External protocol stacks and internal protocol stacks are not allowed to use the same <cid> at the same time.
3. If CGDCONT exists and <cid> is 0, do not modify the configuration of <cid> 0.

3.2 HTTP AT Command Process

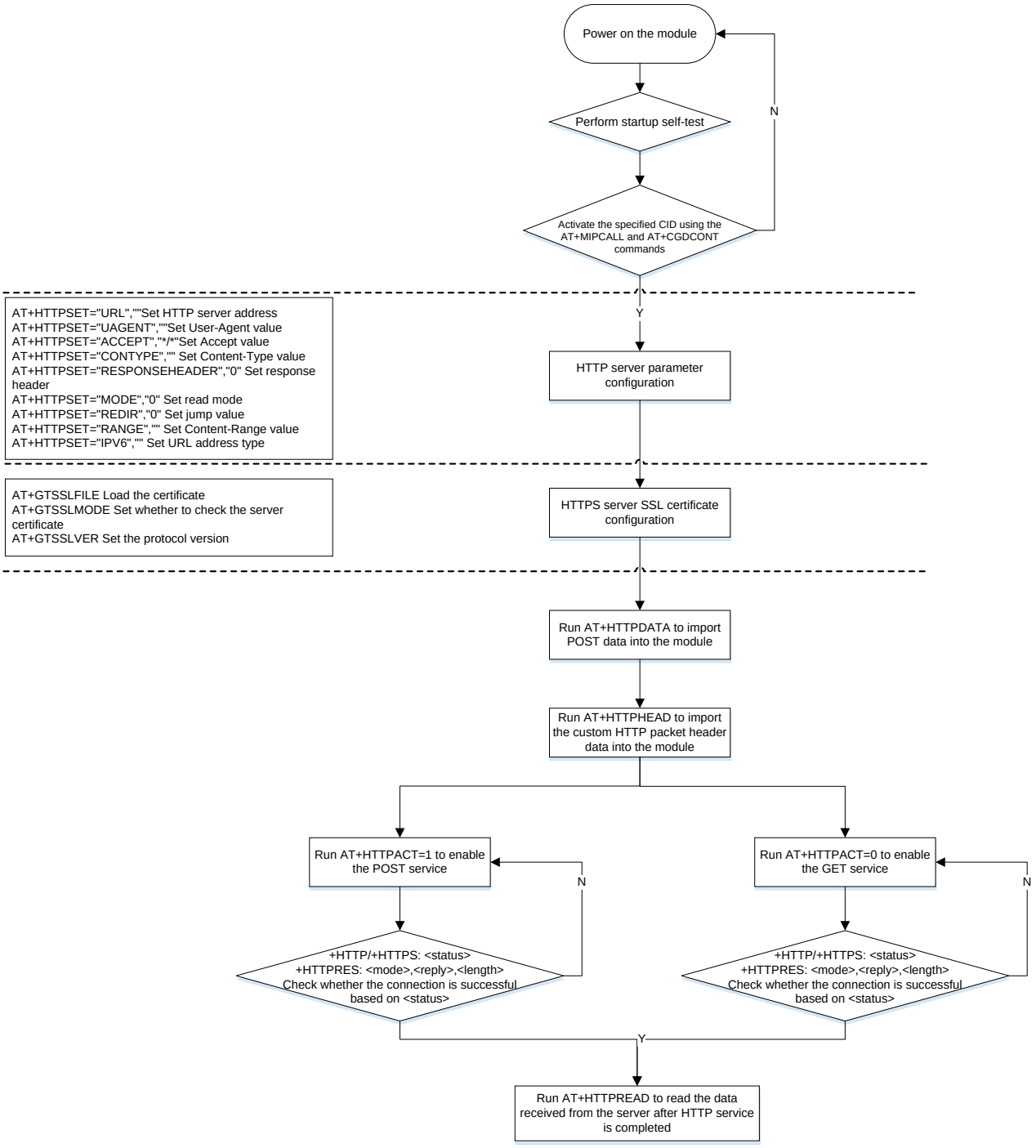

Figure 1. HTTP AT command flowchart

4 Application Examples

4.1 Default Configuration Example

4.1.1 Accessing HTTP Server

4.1.1.1 HTTP GET Service Example

```
AT+MIPCALL=1                          //Set the IP address for PDP activation request
OK

+MIPCALL: 100.73.4.124                    //Wait until IP address is received using +MIPCALL before proceeding with the following operations

AT+HTTPSET="URL","http://47.110.234.36:8451/1K.txt"   //Set the HTTP server address
OK

AT+HTTPACT=0,30                 //HTTP GET service started, request timeout for 30s
OK

+HTTP: 1              //Connection established successfully

+HTTPRES: 0,200,1252            //Subsequent operation can proceed only after +HTTP: 1 and +HTTPRES are returned.

AT+HTTPREAD=0,300               //Read the data with offset 0 and specified length 300 bytes from the module.
//If all is read, send AT+HTTPREAD without subsequent parameters.
+HTTPREAD: 300
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Tue, 09 Apr 2024 03:54:21 GMT
```

```
Content-Type: text/plain; charset=utf-8
Content-Length: 1024
Connection: close
Accept-Ranges: bytes
Last-Modified: Sun, 07 Apr 2024 10:01:56 GMT

36327gf824efeuwh2738oeugf723gduewifg723gyuwgyifg823gwiglf7923gewiufg723g

OK
```

4.1.1.2 HTTP POST Service Example

```
AT+MIPCALL=1                          //Set the IP address for PDP activation request
OK

+MIPCALL: 100.73.4.124                    //Wait until IP address is received using +MIPCALL before proceeding with the following operations
AT+HTTPSET="URL","http://47.110.234.36:8451/post_1k.txt"     //Set the HTTP server address
OK

AT+HTTPDATA=10                        //Import the data that needs POST to the module through the serial port
>                        //Enter data

OK

AT+HTTPACT=1,30                //HTTP POST service started, request timeout for 30s
OK

+HTTP: 1              //Connection established successfully
```

```
+HTTPRES: 1,200,203            //Subsequent operation can proceed only after +HTTP: 1 and
+HTTPRES are returned.


AT+HTTPREAD=0,203                //Read the data with offset 0 and specified length 203
bytes from the module.
//If all is read, send AT+HTTPREAD without subsequent parameters
+HTTPREAD: 203
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Tue, 09 Apr 2024 03:49:56 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 39
Connection: close

{"msg":"Success!","name":"post_1k.txt"}


OK
```

4.1.2 Accessing HTTPS Server

4.1.2.1 HTTPS GET Service Example

```
AT+MIPCALL=1                          //Set the IP address for PDP activation request
OK


+MIPCALL: 100.73.4.124                     //Wait until IP address is received using
+MIPCALL before proceeding with the following operations


AT+GTSSLVER=4                        //Set the SSL version to 4, indicating TLS1.2
OK


AT+GTSSLMODE=1                        //Set the client verification certificate
OK
```

```
AT+GTSSLFILE="TRUSTFILE",1432        //Upload the CA certificate
>
-----BEGIN CERTIFICATE-----          //Certificate content is omitted here
-----END CERTIFICATE-----

OK

AT+GTSSLFILE="CERTFILE",1302       //Upload the client certificate
>
-----BEGIN CERTIFICATE-----          //Certificate content is omitted here
-----END CERTIFICATE-----

OK

AT+GTSSLFILE="KEYFILE",1675        //Upload the client key
>
-----BEGIN RSA PRIVATE KEY-----          //Key content is omitted here
-----END RSA PRIVATE KEY-----

OK

AT+HTTPSET="URL","https://47.110.234.36:8453/1K.txt"   //Set the HTTPS server address
OK

AT+HTTPACT=0,30                //HTTP GET service started, request timeout for 30s
OK

+HTTPS: 1                //Connection established successfully
```

```
+HTTPRES: 0,200,1252              //Subsequent operation can proceed only after +HTTPS:
1 and +HTTPRES are returned.


AT+HTTPREAD=0,50                 //Read the data with offset 0 and specified length 50 bytes
from the module.
//If all is read, send AT+HTTPREAD without subsequent parameters.
+HTTPREAD: 50
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Tue,


OK
```

4.1.2.2 HTTPS POST Service Example

```
AT+MIPCALL=1                            //Set the IP address for PDP activation request
OK


+MIPCALL: 100.73.4.124                        //Wait until IP address is received using
+MIPCALL before proceeding with the following operations


AT+GTSSLVER=4                          //Set the SSL version to 4, indicating TLS1.2
OK


AT+GTSSLMODE=1                          //Set the client verification certificate
OK


AT+GTSSLFILE="TRUSTFILE",1432           //Upload the CA certificate
>
-----BEGIN CERTIFICATE-----             //Certificate content is omitted here
-----END CERTIFICATE-----
```

```
OK

AT+GTSSLFILE="CERTFILE",1302       //Upload the client certificate
>
-----BEGIN CERTIFICATE-----          //Certificate content is omitted here
-----END CERTIFICATE-----

OK

AT+GTSSLFILE="KEYFILE",1675        //Upload the client key
>
-----BEGIN RSA PRIVATE KEY-----          //key content is omitted here
-----END RSA PRIVATE KEY-----

OK

AT+HTTPSET="URL","https://47.110.234.36:8453/post.txt" //Set the HTTPS server address
OK

AT+HTTPDATA=10                        //Import the data that needs POST to the module
through the serial port
>                         //Enter data

OK

AT+HTTPACT=1,30                //HTTP POST service started, request timeout for 30s
OK

+HTTPS: 1                //Connection established successfully
```

```
+HTTPRES: 1,200,203                //Subsequent operation can proceed only after +HTTPS: 1 and +HTTPRES are returned.


AT+HTTPREAD=0,50                   //Read the data with offset 0 and specified length 50 bytes from the module.
//If all is read, send AT+HTTPREAD without subsequent parameters.
+HTTPREAD: 50
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Tue,


OK
```

4.2 Specific Settings Example

4.2.1 HTTPSET

4.2.1.1 RESPONSEHEADER

AT+HTTPSET="RESPONSEHEADER", "0" is used to set whether to carry the packet header. 0: Carry (default); 1: Not carry. You can check whether the packet header is carried when using AT+HTTPREAD to read data.

- Set AT+HTTPSET="RESPONSEHEADER","0" to carry the packet header.

```
AT+HTTPSET="RESPONSEHEADER","0"
OK


AT+HTTPSET="URL","http://47.110.234.36:8000"
OK


AT+HTTPACT=0
OK


+HTTP: 1
```

```
+HTTPRES: 0,200,2012

AT+HTTPREAD
OK
+HTTPREAD: 2012
HTTP/1.1 200 OK
Date: Mon, 09 Oct 2023 07:20:01 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.0.2s
Last-Modified: Fri, 02 Feb 2018 07:59:52 GMT
ETag: "6d5-5643618ea673e"
Accept-Ranges: bytes
Content-Length: 1749
Connection: close
Content-Type: text/html

<html>
<head>……                                    //Content is omitted
```

- Set AT+HTTPSET="RESPONSEHEADER","1" to not carry the packet header.

```
AT+HTTPSET="RESPONSEHEADER","1"
OK

AT+HTTPSET="URL","http://47.110.234.36:8000"
OK

AT+HTTPACT=0
OK

+HTTP: 1
```

```
+HTTPRES: 0,200,1749


AT+HTTPREAD

OK

+HTTPREAD: 1749

<html>

<head>

<STYLE>

TD {

FONT-SIZE: 20px; COLOR: #ffffff; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif

}.....                                    //Content is omitted
```

4.2.1.2 MODE

AT+HTTPSET= "MODE","0" is used to read the data mode. The default value is 0. When all data is downloaded using AT+HTTPACT=0 GET, the data is stored in the module. You can use AT+HTTPREAD to read data from the module. When the parameter is set to 1, the data obtained using the GET command is directly sent to the UART. You can read data while downloading it.

- Set AT+HTTPSET= "MODE","0".

```
AT+HTTPSET="MODE","0"

OK


AT+HTTPSET="URL","http://47.110.234.36:8000"

OK


AT+HTTPACT=0

OK


+HTTP: 1


+HTTPRES: 0,200,2012

AT+HTTPREAD

OK
```

```
+HTTPREAD: 2012

HTTP/1.1 200 OK

Date: Mon, 09 Oct 2023 07:20:01 GMT

Server: Apache/2.4.41 (Win64) OpenSSL/1.0.2s

Last-Modified: Fri, 02 Feb 2018 07:59:52 GMT

ETag: "6d5-5643618ea673e"

Accept-Ranges: bytes

Content-Length: 1749

Connection: close

Content-Type: text/html


<html>

<head>……                                    //Content is omitted
```

- Set AT+HTTPSET="MODE","1".

```
AT+HTTPSET="MODE","1"

OK


AT+HTTPSET="URL","http://47.110.234.36:8000"

OK


AT+HTTPACT=0

OK


+HTTP: 1


+HTTPRES: 0,200,2012


AT+HTTPREAD

OK
```

```
+HTTPREAD:

HTTP/1.1 200 OK

Date: Mon, 09 Oct 2023 07:20:01 GMT

Server: Apache/2.4.41 (Win64) OpenSSL/1.0.2s

Last-Modified: Fri, 02 Feb 2018 07:59:52 GMT

ETag: "6d5-5643618ea673e"

Accept-Ranges: bytes

Content-Length: 1749

Connection: close

Content-Type: text/html


<html>

<head>.....                                    //Content is omitted
```

4.2.1.3 REDIR

REDIR is a redirection technology that provides multiple URL addresses for pages, forms or entire Web sites/applications. AT+HTTPSET= "REDIR", "0" defaults to 0, indicating that the redirection function is disabled. When it is set to 1, the function is enabled and 302 is returned.

- Set AT+HTTPSET= "REDIR","0" to disable the redirection function.

```
AT+HTTPSET="REDIR","0"

OK


AT+HTTPSET="URL","http://www.so.com"

OK


AT+HTTPACT=0

OK


+HTTP: 1


+HTTPRES: 0,302,411
```

- Set AT+HTTPSET= "REDIR","1" to enable the redirection function.

```
AT+HTTPSET="REDIR","1"

OK


AT+HTTPSET="URL","http://www.so.com"

OK


AT+HTTPACT=0

OK


+HTTP: 1


+HTTPRES: 0,302,411


+HTTPS: 1


+HTTPRES: 0,200,178031
```

4.2.1.4 RANGE

RANGE is used to set the value range of the downloaded content. For example, the original content has 1000 bytes. You can set to download only the first 100 bytes with RANGE. The format is At+httpset="RANGE","bytes=0-100".

At+httpset="RANGE","bytes=200-" Download all data after 200 bytes

At+httpset="RANGE","bytes=-50" Download the last 50 bytes of data

Each time it is used, it needs to be input according to the protocol standard, such as "byte=0-12". Value 206 will be returned upon successful operation and the record from the previous time will be overwritten. This value is not valid for POST.

```
AT+HTTPSET="RANGE","bytes=0-200"

OK


AT+HTTPSET="URL","http://47.110.234.36:8000"

OK
```

```
AT+HTTPACT=0
OK

+HTTP: 1

+HTTPRES: 0,206,509

AT+HTTPREAD
OK
+HTTPREAD: 509
HTTP/1.1 206 Partial Content
Date: Mon, 09 Oct 2023 08:13:28 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.0.2s
Last-Modified: Fri, 02 Feb 2018 07:59:52 GMT
ETag: "6d5-5643618ea673e"
Accept-Ranges: bytes
Content-Length: 201
Content-Range: bytes 0-200/1749
Connection: close
Content-Type: text/html

<html>
<head>
<STYLE>
TD {
FONT-SIZE: 20px; COLOR: #ffffff; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif
}
</STYLE>
<SCRIPT language=JavaScript>
```

```
function tick() {

var years,months,days,h
```

4.2.1.5 IPV6

IPV6 is used to set the URL address type for access. AT+HTTPSET= "IPV6", "0". 0 indicates the IPv4 address, and 1 indicates the IPv6 address.

```
AT+MIPCALL=2,"CTNET"

OK


+MIPCALL: 0.0.0.0,fe80:0000:0000:0000:178c:63a1:218e:7612


AT+HTTPSET="IPV6","1"

OK


AT+HTTPSET="URL","[2400:3200:1300::c6a]:10080"

OK


AT+HTTPACT=0

OK


+HTTP: 1


+HTTPRES: 0,200,2012
```

5 Data Link Disconnected

To check for the possible causes of the disconnection, do as follows:

1. Run AT+CGREG? to check whether the data service is still available. Run AT+CGREG? to check whether the following information is returned: +CGREG: 2,0. If not, try to resend the command for at most 300s. 300s later, the module restarts.
2. Run the AT+CSQ? command to check the signal strength. Check whether the timeout is less than 1s. The first value in the +CSQ: xx,xx response indicates the signal value, which varies with the network environment. This value is used to determine whether the antenna or the server is the cause of the abnormal data sending and receiving. The greater the value is within 0-31, the better the signal quality is. A value smaller than or equal to 12 indicates a weak signal, and a value greater than or equal to 21 indicates a strong signal. The value 99 indicates unknown or unavailable network. When the command is sent, it is not necessary to determine the first value in the response, because as long as the step (1) can be executed, dial-up can be performed. If you want to increase the success rate of dial-up, you can add a judgment condition: The first parameter value is greater than 15 (inclusive) and less than 32. Repeat the query for a maximum of 90s. Otherwise, continue to query or restart the module. If the signal strength keeps low, check whether an appropriate antenna is selected and whether the antenna connection is loose or damaged.
3. Run the AT+MIPCALL? command to heck whether the module still has an IP address. Proceed if an IP address is available. If no IP address is available, run AT+MIPCALL=1,"CMNET". The timeout duration is 30s. Generally, the command can be repeated for 3 times. If this does not work, continue to query or restart the module.

If the connection fails all the time, the cause may be SIM card arrears, poor signal, incorrect antenna, or server exception. Solve the problem according to the actual cause.

6 Appendix Terms and Abbreviations

Table 1. Term and abbreviations

Acronyms	Full Spelling
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
PDP	Packet Data Protocol
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
SSL	Secure Sockets Layer
TLS	Transport Layer Security