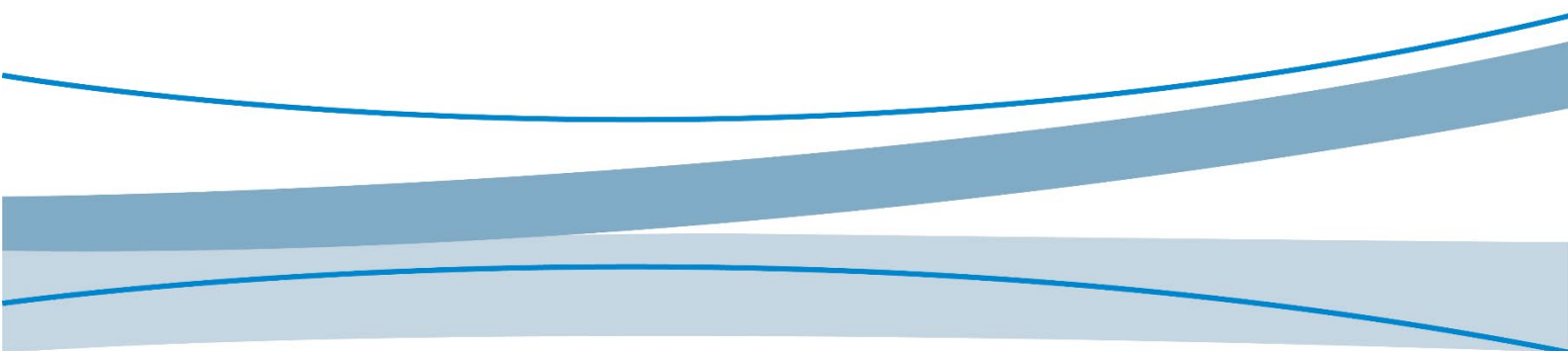




MTC

Application Guide_TCP&UDP

V1.2



Disclaimer

Any action you take in the course of using this document is at your own risk, and Fibocom shall not be liable for any damages or losses under any circumstances. Due to product version upgrade or other reasons, Fibocom reserves the right to modify any information in this document at any time without prior notice and any responsibility. Unless otherwise agreed, all statements, information and suggestions in this document do not constitute any express or implied guarantee.

This document may include the third-party information covering products, services, software, data, and so on. Fibocom does not control and assumes no responsibility for the third-party content, including but not limited to the accuracy, compatibility, reliability, availability, legitimacy, appropriateness, performance, non-infringement, and status update, unless otherwise specified in this document. Fibocom does not provide any guarantee or authorization for the third-party content mentioned or referenced in this document. If you need a third-party license, obtain it in an authorized or legal way, unless otherwise specified in this document.

Copyright Notice

Copyright © 2024 Fibocom Wireless Inc. All rights reserved.

Unless specially authorized by Fibocom, the recipient of the documents shall keep the documents and information received confidential, and shall not use them for any purpose other than the implementation and development of this project. Without the written permission of Fibocom, no unit or individual shall extract or copy part or all of the contents of this document without authorization, or transmit them in any form. Fibocom has the right to investigate legal liabilities for any offense and tort in connection with violation of confidentiality obligations, or unauthorized use or malicious use of the said documents and information in other illegal forms.

Trademark Statement

 The trademark is registered and owned by Fibocom Wireless Inc.

Other trademarks, product names, service names and company names appearing in this document are owned by their respective owners.

Contact Information

Website: <https://www.fibocom.com>

Address: 10/F-14/F, Block A, Building 6, Shenzhen International Innovation Valley, Dashi First Road, Xili Community, Xili Subdistrict, Nanshan District, Shenzhen

Tel: 0755-26733555

Contents

Applicable Model	3
Change History	4
1 About This Document	5
2 Reference AT Command	6
3 TCP&UDP Overview	7
3.1 Internal Dial-up	7
3.2 TCP&UDP AT Command Process	8
3.3 Data Access Modes	8
4 Application Examples	10
4.1 Direct Throughput Mode of TCP	10
4.1.1 IPRFMT Mode 0	10
4.1.1.1 Creating a TCP Connection	10
4.1.1.2 Directly Sending Data with AD Commands	11
4.1.1.3 Sending Data in HEX Mode	11
4.1.1.4 Disconnecting the TCP Connection	13
4.1.2 IPRFMT Mode 1	13
4.1.2.1 Creating a TCP Connection	13
4.1.2.2 Directly Sending Data with AD Commands	14
4.1.2.3 Sending Data in HEX Mode	14
4.1.2.4 Disconnecting the TCP Connection	16
4.1.3 IPRFMT Mode 2	16
4.1.3.1 Creating a TCP Connection	16
4.1.3.2 Directly Sending Data with AD Commands	17
4.1.3.3 Sending Data in HEX Mode	17
4.1.3.4 Disconnecting the TCP Connection	19
4.2 TCP Cache Mode Example	19
4.2.1 Creating a TCP Connection	19
4.2.2 Directly Sending Data with AD Commands	20
4.2.3 Sending Data in HEX Mode	21

4.2.4 Disconnecting the TCP Connection.....	23
4.3 TCP Transparent Transmission Mode Example	23
4.3.1 Creating a TCP Connection.....	23
4.3.2 Sending Data in Transparent Transmission Mode	24
4.3.3 Receiving Data in Transparent Transmission Mode.....	24
4.3.4 Disconnecting the TCP Connection.....	24
4.4 UDP Service Example	24
4.4.1 Enabling the UDP Service	25
4.4.2 Sending UDP Data to the Server	25
4.4.3 Receive data.	25
4.4.4 Disabling the UDP Service and Client.....	26
4.5 TCP Service Example	26
4.5.1 Enabling the TCP Service	26
4.5.2 Sending TCP Data to the Server.....	27
4.5.3 Receive data.	27
4.5.4 Disabling the TCP Service and Client	28
4.6 TCPS One-way Authentication	28
4.7 TCPS Two-way Authentication.....	29
4.8 PING	30
4.9 DNS Resolution.....	31
4.10 Time Synchronization with NTP.....	32
5 +MIPXON and +MIPXOFF Represent.....	33
Appendix A Terms and Abbreviations	34

Applicable Model

No.	Applicability Model	Description
1	All MTC products	\

Change History

V1.2 (2024-09-27)	Fixed some parameter content description errors.
V1.1 (2024-08-15)	Some formatting content has been adjusted, add tcp service example.
V1.0 (2023-12-11)	Initial version

1 About This Document

.....

The module integrates a standard TCP/IP protocol stack. The host computer can send AT commands to the module to access the network and implement data transmission to fulfill related network services. The TCP&UDP function provides TCP client, UDP client, TCP server, and UDP server. To implement TLS/DTLS client services, you can use SSL commands.

2 Reference AT Command

Refer to TCP&UDP, SSL, and universal AT command manuals of the corresponding platform.

3 TCP&UDP Overview

Before using the TCP&UDP 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.



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

3.2 TCP&UDP AT Command Process

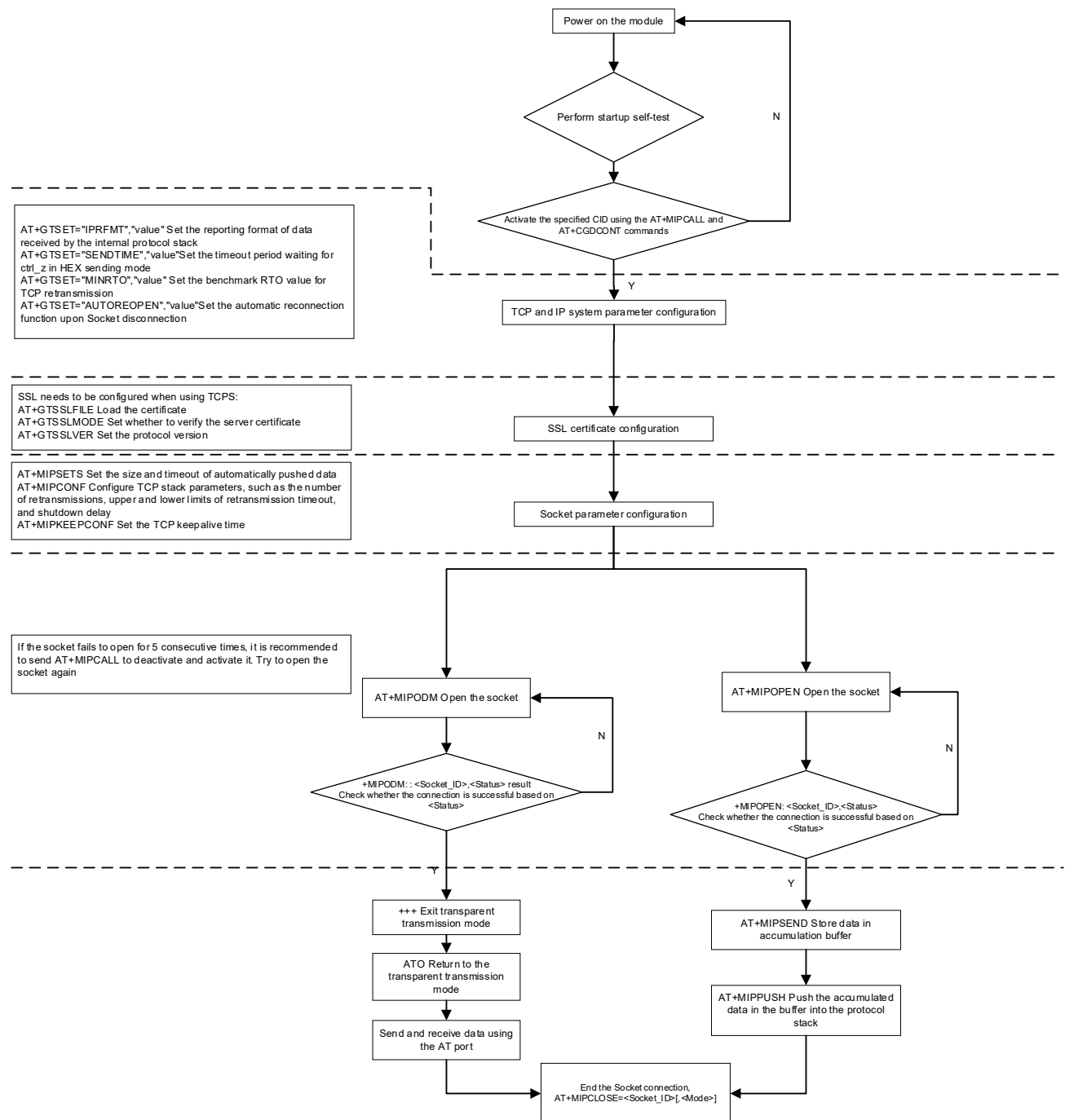


Figure 1. TCP&UDP AT command flowchart

3.3 Data Access Modes

The module supports the following data access modes:

- ① Cache mode
- ② Direct throughput mode

③ Transparent transmission mode

For the cache mode and direct throughput mode, run AT+GTSET="IPRfmt",X to set the data output mode. The Socket connection cannot be switched after it is established. The connection in transparent transmission mode is created using AT+MIPODM.

1. The connection in cache mode is configured using AT+GTSET="IPRfmt",5, and data is sent using AT+MIPSEND. Upon receiving the data from the network side, the module reports +MIPREAD: <Socket_ID>,<DataLen> to the URC. Then, you can read the cached data using AT+MIPREAD.
2. In direct throughput mode, data is sent using AT+MIPSEND. The data received from the network will be output to the AT port. There are three formats for the received data, which are determined by AT+GTSET="IPRfmt",0/1/2, which is set before the socket creation.
 - When AT+GTSET="IPRfmt",0, the received data is +MIPRTCP: <Socket_ID>,<Left >,<hex encoding of original data>, +MIPRUDP: <Source_IP>,<Source_Port>,<Socket_ID >,<Left >,<hex encoding of original data>.
 - When AT+GTSET="IPRfmt",1, the received data is the raw data.
 - When AT+GTSET="IPRfmt",2, the received data is +MIPRTCP: <Socket_ID>,<Data_len>,<Raw data>, +MIPRUDP: <Source_IP>,<Source_Port>,<Socket_ID>,<Data_len>,<Raw data>.
3. In transparent transmission mode, the AT port is occupied, the data received by the AT port is sent directly to the network, and the data received from the network is also directly output from the AT port. You can send +++ to exit the transparent transmission mode. When OK is returned in response to +++, if the parameter <Pseudo-Command Mode> set in MIPODM is 1, the Socket connection is disabled directly. If it is not set or set to 0, the access mode is cache mode. At this time, messages sent using MIPSEND are cached. The size of cached data to be sent to the network is determined by the reported +MIPDATA: <Socket_ID>,<length>. When the module returns to transparent transmission mode through the ATO command, the data cached in the module is sent to the terminal immediately.

Exiting the transparent transmission mode

You can send +++ to exit the transparent transmission mode. In order to prevent +++ from being sent as data, the following rules must be followed during actual operation:

1. No other data can be entered in 1 second or more before +++ is entered.
2. +++ must be entered within 1 second and no other data can be entered.
3. No other data can be entered within 1 second after +++ is entered.

Switching to transparent transmission

After you exit the transparent transmission mode using the ATO, you can switch back to the transparent transmission mode using the ATO. If CONNECT is returned, the module successfully switches to the transparent transmission mode. If the module has not entered the transparent transmission mode before, ERROR will be returned when ATO is executed.



When MIPSEND sends data that ends with Ctrl-Z, the data cannot carry Ctrl-Z. Otherwise, it cannot be sent normally. If the data contains Ctrl-Z, it is recommended to send it in the form of AT+MIPSEND=<Socket_ID>,<Data_len>.

4 Application Examples

4.1 Direct Throughput Mode of TCP

4.1.1 IPRFMT Mode 0

When receiving data with +MIPRTCP or +MIPRUDP, the module encodes the data.

4.1.1.1 Creating a TCP Connection

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

+MIPCALL: 10.156.151.176 //Wait until IP address is received using
+MIPCALL before proceeding with the following operations

AT+GTSET="IPRFMT",0 //Set the format of the data reported when the
module receives data from the server
OK

AT+MIOPEN? //Query the socket status. Note that the query
can proceed only when the previous asynchronous command is executed
+MIOPEN: 1,2,3,4,5,6

OK

AT+MIOPEN=1,0,"111.231.250.105",20000,0 //Connect to the server through port 20000
OK

+MIOPEN: 1,1 //The asynchronous command is executed,
indicating that the module successfully establishes a link to the server through
SOCKET ID 1
```

4.1.1.2 Directly Sending Data with AD Commands

```
AT+MIPSEND=1,"3131313131"           //Directly send data with the AD commands
+MIPSEND: 1,0,2043                    //The size of cached data sent here changes
from 2048 to 2043

OK

AT+MIPPUSH=1                          //Push the data in the sent data cache to
the server
+MIPPUSH: 1,0

OK

+MIPRTCP: 1,0,3131313131              //Data response sent by the server, displayed
in HEX format
```

4.1.1.3 Sending Data in HEX Mode

- Sending data based on the length

```
AT+MIPSEND=1,5                        //Send 5 bytes of data to the server
>                                     //Enter 11111

OK

+MIPPUSH: 1,0                        //Successfully pushed to the server

+MIPSEND: 1,0,2048                   //The data size in the sent data cache is
restored to 2048

OK

+MIPRTCP: 1,0,3131313131             //Data response sent by the server, displayed
```

in HEX format

- Sending data in Ctrl-Z form

```
AT+MIPSEND=1 //Enable the data channel
> //Enter 11111 and send data ended with Ctrl+Z

OK

+MIPPUSH: 1,0 //Successfully pushed to the server

+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048

OK

+MIPRTCP: 1,0,3131313131 //Data response sent by the server, displayed
in HEX format
```

- Data sending timeout

```
AT+MIPSEND=1 //Enable the data channel
> //Enter 11111 and wait until timeout

OK

+MIPPUSH: 1,0 //Successfully pushed to the server

+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048

OK

+MIPRTCP: 1,0,3131313131 //Data response sent by the server, displayed
in HEX format
```

4.1.1.4 Disconnecting the TCP Connection

```
AT+MIPCLOSE=1 //Disconnect the connection through Socket
ID 1
OK

+MIPCLOSE: 1,0 //Successfully disconnected
```

4.1.2 IPRFMT Mode 1

Only the data is received, and the data is not encoded.

4.1.2.1 Creating a TCP Connection

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

+MIPCALL: 10.156.151.176 //Wait until IP address is received using
+MIPCALL before proceeding with the following operations

AT+GTSET="IPRFMT",1 //Set the format of the data reported when the
module receives data from the server
OK

AT+MIOPEN? //Query the socket status. Note that the query
can proceed only when the previous asynchronous command is executed
+MIOPEN: 1,2,3,4,5,6

OK

AT+MIOPEN=1,0,"111.231.250.105",20000,0 //Connect to the server through port 20000
OK

+MIOPEN: 1,1 //The asynchronous command is executed,
```

indicating that the module successfully establishes a link to the server through SOCKET ID 1

4.1.2.2 Directly Sending Data with AD Commands

```
AT+MIPSEND=1,"3131313131"           //Directly send data with the AD commands
+MIPSEND: 1,0,2043                    //The size of cached data sent here changes
from 2048 to 2043

OK

AT+MIPPUSH=1                          //Push the data in the sent data cache to
the server
+MIPPUSH: 1,0

OK

11111                                //Data response sent by the server, displayed
in raw format
```

4.1.2.3 Sending Data in HEX Mode

- Sending data based on the length

```
AT+MIPSEND=1,5                        //Send 5 bytes of data to the server
>                                     //Enter 11111

OK

+MIPPUSH: 1,0                         //Successfully pushed to the server

+MIPSEND: 1,0,2048                    //The data size in the sent data cache is
restored to 2048
```


OK

11111 //Data response sent by the server, displayed
in raw format

- Sending data in Ctrl-Z form

AT+MIPSEND=1 //Enable the data channel
> //Enter 11111 and send data ended with Ctrl+Z

OK

+MIPUSH: 1,0 //Successfully pushed to the server

+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048

OK

11111 //Data response sent by the server, displayed
in raw format

- Data sending timeout

AT+MIPSEND=1 //Enable the data channel
> //Enter 11111 and wait until timeout

OK

+MIPUSH: 1,0 //Successfully pushed to the server

+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048

OK

```
11111 //Data response sent by the server, displayed  
in raw format
```

4.1.2.4 Disconnecting the TCP Connection

```
AT+MIPCLOSE=1 //Disconnect the connection through Socket  
ID 1  
OK  
  
+MIPCLOSE: 1,0 //Successfully disconnected
```

4.1.3 IPRFMT Mode 2

Data is received with MIPRTCP or MIPRUDP, and the data is not encoded.

4.1.3.1 Creating a TCP Connection

```
AT+MIPCALL=1 //Set the IP address for PDP activation request  
OK  
  
+MIPCALL: 10.156.151.176 //Wait until IP address is received using  
+MIPCALL before proceeding with the following operations  
  
AT+GTSET="IPRFMT",2 //Set the format of the data reported when the  
module receives data from the server  
OK  
  
AT+MIOPEN? //Query the socket status. Note that the query  
can proceed only when the previous asynchronous command is executed  
+MIOPEN: 1,2,3,4,5,6  
  
OK
```

```
AT+MIOPEN=1,0,"111.231.250.105",20000,0    //Connect to the server through port 20000
OK

+MIOPEN: 1,1                                //The asynchronous command is executed,
indicating that the module successfully establishes a link to the server through
SOCKET ID 1
```

4.1.3.2 Directly Sending Data with AD Commands

```
AT+MIPSEND=1,"3131313131"                  //Directly send data with the AD commands
+MIPSEND: 1,0,2043                          //The size of cached data sent here changes
from 2048 to 2043

OK

AT+MIPPUSH=1                                //Push the data in the sent data cache to
the server
+MIPPUSH: 1,0

OK

+MIPRTCP: 1,5,11111                        //Data response sent by the server,
displayed in raw format
```

4.1.3.3 Sending Data in HEX Mode

- Sending data based on the length

```
AT+MIPSEND=1,5                             //Send 5 bytes of data to the server
>                                           //Enter 11111

OK

+MIPPUSH: 1,0                             //Successfully pushed to the server
```

```
+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048
```

```
OK
```

```
+MIPRTCP: 1,5,11111 //Data response sent by the server,
displayed in raw format
```

- Sending data in Ctrl-Z form

```
AT+MIPSEND=1 //Enable the data channel
> //Enter 11111 and send data ended with Ctrl+Z
```

```
OK
```

```
+MIPPUSH: 1,0 //Successfully pushed to the server
```

```
+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048
```

```
OK
```

```
+MIPRTCP: 1,5,11111 //Data response sent by the server,
displayed in raw format
```

- Data sending timeout

```
AT+MIPSEND=1 //Enable the data channel
> //Enter 11111 and wait until timeout
```

```
OK
```

```
+MIPPUSH: 1,0 //Successfully pushed to the server
```

```
+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048

OK

+MIPRTCP: 1,5,11111 //Data response sent by the server,
displayed in raw format
```

4.1.3.4 Disconnecting the TCP Connection

```
AT+MIPCLOSE=1 //Disconnect the connection through Socket
ID 1

OK

+MIPCLOSE: 1,0 //Successfully disconnected
```

4.2 TCP Cache Mode Example

Data is received in cache mode.

4.2.1 Creating a TCP Connection

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

OK

+MIPCALL: 10.156.151.176 //Wait until IP address is received using
+MIPCALL before proceeding with the following operations

AT+GTSET="IPRFRMT",5 //Set the format of the data reported when the
module receives data from the server

OK

AT+MIOPEN? //Query the socket status. Note that the query
can proceed only when the previous asynchronous command is executed

+MIOPEN: 1,2,3,4,5,6
```

OK

```
AT+MIOPEN=1,0,"111.231.250.105",20000,0 //Connect to the server through port 20000
```

OK

```
+MIOPEN: 1,1 //The asynchronous command is executed,
indicating that the module successfully establishes a link to the server through
SOCKET ID 1
```

4.2.2 Directly Sending Data with AD Commands

```
AT+MIPSEND=1,"3131313131" //Directly send data with the AD commands
```

```
+MIPSEND: 1,0,2043
```

OK

```
AT+MIPPUSH=1 //Push the data in the sent data cache to
the server
```

```
+MIPPUSH: 1,0
```

OK

```
+MIPREAD: 1,5 //Receive the 5 bytes of data sent by the
server and saved in the cache
```

```
AT+MIPREAD=1,5 //Read 5 bytes of data
```

```
+MIPDATA: 1,5
```

```
11111
```

OK

4.2.3 Sending Data in HEX Mode

- Sending data based on the length

```
AT+MIPSEND=1,5 //Send 5 bytes of data to the server
>

OK

+MIP_PUSH: 1,0 //Successfully pushed to the server

+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048

OK

+MIP_READ: 1,5 //Receive the 5 bytes of data sent by the
server and saved in the cache

AT+MIP_READ=1,5 //Read 5 bytes of data
+MIPDATA: 1,5
11111

OK
```

- Sending data in Ctrl-Z form

```
AT+MIPSEND=1 //Enable the data channel
>

OK

+MIP_PUSH: 1,0 //Successfully pushed to the server
```

```
+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048
```

```
OK
```

```
+MIPREAD: 1,5 //Receive the 5 bytes of data sent by the
server and saved in the cache
```

```
AT+MIPREAD=1,5 //Read 5 bytes of data
```

```
+MIPDATA: 1,5
```

```
11111
```

```
OK
```

- Data sending timeout

```
AT+MIPSEND=1 //Enable the data channel
```

```
>
```

```
OK
```

```
+MIPPUSH: 1,0 //Successfully pushed to the server
```

```
+MIPSEND: 1,0,2048 //The data size in the sent data cache is
restored to 2048
```

```
OK
```

```
+MIPREAD: 1,5 //Read 5 bytes of data
```

```
AT+MIPREAD=1,5 //Read 5 bytes of data
```

```
+MIPDATA: 1,5
```

```
11111
```



```
OK
```

4.2.4 Disconnecting the TCP Connection

```
AT+MIPCLOSE=1 //Disconnect the connection through Socket
ID 1
OK

+MIPCLOSE: 1,0 //Successfully disconnected
```

4.3 TCP Transparent Transmission Mode Example

4.3.1 Creating a TCP Connection

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

+MIPCALL: 10.156.151.176 //Wait until IP address is received using
+MIPCALL before proceeding with the following operations

AT+MIPODM? //Query the socket status. Note that the query
can proceed only when the previous asynchronous command is executed
+MIPODM: 1,2,3,4,5,6

OK

AT+MIPODM=1,0,"111.231.250.105",20000,0 //Create a socket connection in transparent
transmission mode
OK

+MIPODM: 1,1 //Successfully created
```

4.3.2 Sending Data in Transparent Transmission Mode

```
<Data received by serial port>           //Send the data obtained by the serial
port directly to the network

<Enter +++ to exit the transparent transmission mode>

OK

ATO                                     //Return to the transparent transmission mode

CONNECT                               //The transparent transmission mode is successfully
established

<Data received by serial port>           //Send the data obtained by the serial
port directly to the network
```

4.3.3 Receiving Data in Transparent Transmission Mode

```
<The received network data is output directly from the serial port>
```

4.3.4 Disconnecting the TCP Connection

```
AT+MIPCLOSE=1                           //Disconnect the connection through Socket
ID 1

OK

+MIPCLOSE: 1,0                           //Successfully disconnected

AT+MIPODM?                               //View the socket status. AT+MIPODM? can also be used
during transparent transmission

+MIPODM: 1,2,3,4,5,6                     //Proceed with the following operations after the
previous asynchronous command ends

OK
```

4.4 UDP Service Example

In this example, the server and client are in the same module.

4.4.1 Enabling the UDP Service

```
AT+MIPOPEN=1,5000,"0.0.0.0",0,1 //Enable the UDP service
```

```
OK
```

```
+MIPOPEN: 1,1
```

4.4.2 Sending UDP Data to the Server

```
AT+MIPOPEN=2,5001,"10.157.8.35",5000,1 //Create a UDP client and configure a  
fixed local port
```

```
OK
```

```
+MIPOPEN: 2,1 //Successfully created the client
```

```
AT+MIPSEND=2,"3131313131" //Send 5 bytes of data to the cache
```

```
+MIPSEND: 2,0,2043
```

```
OK
```

```
AT+MIPPUSH=2,"10.157.8.35",5000 //Send 5 bytes of data in the cache to  
the server. The server address is 10.157.8.35  
and the port number is 5000
```

```
+MIPPUSH: 2,0 //Data is sent successfully
```

```
OK
```

```
+MIPRUDD: 10.157.8.35,5001,1,0,3131313131 //The server receives 5 bytes of data
```

4.4.3 Receive data.

```
AT+MIPSEND=1,"3131313131" //The server sends 5 bytes of data to  
the cache
```

```
+MIPSEND: 1,0,2043
```

OK

```
AT+MIPUSH=1,"10.157.8.35",5001 //The server sends 5 bytes of data to  
the cache. The client address is 10.157.8.35 and the port number is 5001
```

```
+MIPUSH: 1,0 //Data is sent successfully
```

OK

```
+MIPUDP: 10.157.8.35,5000,2,0,31313131 //The client receives the 5 bytes of  
data sent by the server
```

4.4.4 Disabling the UDP Service and Client

```
AT+MIPCLOSE=1 //Disable the server socket
```

OK

```
+MIPCLOSE: 1,0
```

```
AT+MIPCLOSE=2 //Disable the client socket
```

OK

```
+MIPCLOSE: 2,0
```

4.5 TCP Service Example

In this example, the server and client are in the same module.

4.5.1 Enabling the TCP Service

```
AT+MIOPEN=1,5000,"0.0.0.0",0,1 //Enable the UDP service
```

OK

```
+MIOPEN: 1,1
```

4.5.2 Sending TCP Data to the Server

```
AT+MIOPEN=2,5001,"100.80.246.142",5000,0 //Create a TCP client and configure a
                                         fixed local port

OK

+MIOPEN: 3,1,100.80.246.142,5001 //Server open data channel

+MIOPEN: 2,1 //Successfully created the client

AT+MIPSEND=2,"3131313131" //Send 5 bytes of data to the cache

+MIPSEND: 2,0,2043

OK

AT+MIPPUSH=2 //Send 5 bytes of data in the cache to
              the server.

+MIPPUSH: 2,0 //Data is sent successfully

OK

+MIPRTCP: 3,0,3131313131 //The server receives 5 bytes of data
```

4.5.3 Receive data.

```
AT+MIPSEND=3,"3131313131" //The server sends 5 bytes of data to
                           the cache

+MIPSEND: 3,0,2043

OK

AT+MIPPUSH=3 //The server sends 5 bytes of data to
              the cache.

+MIPPUSH: 3,0 //Data is sent successfully
```

OK

```
+MIPRTCP: 2,0,3131313131 //The client receives the 5 bytes of  
data sent by the server
```

4.5.4 Disabling the TCP Service and Client

```
AT+MIPCLOSE=1 //Disable the server socket
```

OK

```
+MIPCLOSE: 1,0
```

```
AT+MIPCLOSE=2 ///Disable the client socket
```

OK

```
+MIPCLOSE: 2,0
```

```
+MIPSTAT: 3,2
```

4.6 TCPS One-way Authentication

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

OK

```
+MIPCALL: 11.152.65.135 //Wait until IP address is received using  
+MIPCALL before proceeding with the following operations
```

```
AT+GTSSLMODE=0 //Set SSL not to validate the server  
certificate
```

OK

```
AT+MIOPEN=1,,"47.110.234.36",8887,2 //Enable the TCP connection encrypted
OK

+MIOPEN: 1,1 //The asynchronous command is executed,
indicating that the module successfully establishes a link to the server through
SOCKET ID 1

AT+MIPCLOSE=1 //Disconnect the connection through Socket
ID 1
OK

+MIPCLOSE: 1,0 //Successfully disconnected
```

4.7 TCPS Two-way Authentication

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

+MIPCALL: 11.152.65.135 //Wait until IP address is received using
+MIPCALL before proceeding with the following operations

AT+GTSSLFILE="TRUSTFILE",<certificate length> //Set the CA certificate. The
length of the certificate needs to map the certificate
>
<Enter the CA certificate>

OK

AT+GTSSLFILE="CERTFILE",<certificate length> //Set the client certificate. The
length of the certificate needs to map the certificate
>
<Enter the client certificate>
```

OK

```
AT+GTSSLFILE="KEYFILE",<key length> //Set the client key. The length of the key
needs to map the key
>
<Enter the client key>
```

OK

```
AT+GTSSLMODE=1 //Set the SSL verifies the server
certificate
```

OK

```
AT+MIOPEN=1,,"47.110.234.36",8888,2 //Enable the TCP connection encrypted
```

OK

```
+MIOPEN: 1,1 //The asynchronous command is executed,
indicating that the module successfully establishes a link to the server through
SOCKET ID 1
```

```
AT+MIPCLOSE=1 //Disconnect the connection through Socket
ID 1
```

OK

```
+MIPCLOSE: 1,0 //Successfully disconnected
```

4.8 PING

```
AT+MPING=1,"www.baidu.com" //Ping www.baidu.com
```

OK


```
+MPING: "110.242.68.4",0,0,68

+MPING: "110.242.68.4",0,0,109

+MPING: "110.242.68.4",0,0,108

+MPING: "110.242.68.4",0,0,106

+MPINGSTAT: 0,"110.242.68.4",4,4,97
```

4.9 DNS Resolution

```
AT+MIPDNS="www.baidu.com" //Request to resolve www.baidu.com
+MIPDNS: "www.baidu.com",110.242.68.4

OK

AT+MIPDNS="www.baidu.com",0 //Request to resolve IPv4 address
of www.baidu.com
+MIPDNS: "www.baidu.com",110.242.68.4

OK

AT+MIPDNS="www.baidu.com",1 //Request to resolve IPv6 address of
www.baidu.com
+MIPDNS: "www.baidu.com",2408:871A:2100:3:0:FF:B025:348D

OK

AT+MIPDNS="www.baidu.com",2 //Request to resolve IPv4 and IPv6
addresses of www.baidu.com
+MIPDNS: "www.baidu.com",110.242.68.3,2408:871A:2100:3:0:FF:B025:348D
```

OK

4.10 Time Synchronization with NTP

```
AT+MIPNTP="ntp.ntsc.ac.cn",123          //Use NTP server "ntp.ntsc.ac.cn:123"
to synchronize time
```

OK

```
+MIPNTP: 1                               //Time synchronization is successful
```

```
AT+CCLK?                                 //Use the CCLK command to query the
current time
```

```
+CCLK: "23/09/04,15:17:41+32"
```

OK

5 +MIPXON and +MIPXOFF Represent

In case of network congestions, +MIPXOFF is reported if data cannot be sent. It reminds the MCU to stop sending data to the module's buffer. When +MIPXON is reported, it means that you can send data with **AT+MIPSEND**. Any forced sending may cause socket release. It is recommended to wait for 30 seconds at most after +MIPXON is reported, then send **AT+MIPCLOSE** to disconnect the socket, send another command to recover the data connection, or perform other operations.

Appendix A Terms and Abbreviations

Table 1. Terms and abbreviations

Acronyms	Full Spelling
DNS	Domain Name System
IP	Internet Protocol
NTP	Network Time Protocol
PDP	Packet Data Protocol
TCP	Transmission Control Protocol
TTL	Time to Live
UDP	User Datagram Protocol
URC	Unsolicited Result Code