ST87MXX TCP/UDP-IP Application Note

Introduction

This document gives details of the AT Commands usage for embedded TCP/UDP-IP stack supported by the ST87MXX NB-IoT module.

Document status
V1.0

1 General information

1.1 Acronyms and terms

Table 1. Definitions of terms

Term	Definition
DNS	Domain Name System
TCP	Transport Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol
JTAG	Joint Test Action Group
MBR	Maximum Bit Rate
NCP	Network Control Protocol
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
QCI	QoS Class Identifier
QoS	Quality of Service
UART	Universal Asynchronous Receiver Transmitter
TA	Terminal Adaptor, e.g. a GSM data card (equal to DCE; Data Circuit terminating Equipment)
TE	Terminal Equipment, e.g. a computer (equal to DTE; Data Terminal Equipment)
TFT	Traffic Flow Template
ME	Mobile Equipment

1.2 Reference documents

The documents listed in Table 2 provide further information.

Table 2. **Document references**

Reference	Document
[1]	3GPP TS 24.301 V15.8.0
[2]	3GPP TS 24.008 V15.9.0
[3]	3GPP TS 23.682 V15.10.0
[4]	3GPP TS 23.401 V15.12.0
[5]	3GPP TS 23.682 V15.10.0
[6]	3GPP TS 36.133 V15.15.0
[7]	3GPP TS 31.101 V15.3.0
[8]	3GPP TS 36.106 V12.1.0
[9]	3GPP TS 27.060 V15.0.0
[10]	3GPP TS 29.061 V15.7.0
[11]	3GPP TS 23.203 V15.4.0
[12]	3GPP TS 27.007 V15.3.0
[13]	3GPP TS 23.060 V15.4.0
[14]	3GPP TC 27.005 V10.0.0

1.3 Revision history

Table 3. Document revision history

Date	Version	Changes
2022-04-27	V0.1	First release
2023-12-05	V1.0	Official release

1.4 Table of contents

1.5 List of tables

2 Introduction

2.1 AT command syntax

All the commands start with the "AT" prefix and ends with a carriage return character <CR>.

Responses start with a carriage return and a line feedback character and end with a carriage return and a line feedback character <CR><LF><response><CR><LF>

Throughout the document the <CR> and <LF> characters are omitted for the sake of readability.

The AT Command set implemented in the ST87MXX is a combination of 3GPP TS 27.005, 3GPP TS 27.007 and ITU-T recommendation V.25ter and the AT proprietary commands developed by ST. The proprietary ST commands are formatted as follow: AT#CMD.

Each AT command is described using the following format:

Table 4. AT command format

Command type	Syntax	Description
Test command	AT+CMD=?	Gives information about the type, value, or range of its parameter.
Read command	AT+CMD?	Checks the current parameter value
Write Command	AT+CMD=<param1>[,<param2>,[...]]]	Sets parameter value.
Execution command	AT+CMD	Performs a specific action.

For a given AT command, if a command type is not described in the document this means that it doesn’t exist.

3 TCP-UDP-IP AT commands

3.1 AT#IPCFG – IPv4 or IPv6 configuration

This command is used to configure and activate the IPv4 or IPv6 protocol stack on top of an already established PDP context.
3 ways are possible to setup a new IP address:

- Create a static (manual) IP address. The entered IP address is analysed to determine the IP version (v4 or v6).
  - In case of IPv4 address, only the IP address is specified.
  - In case of IPv6 address, the IP address or prefix, and the prefix length shall be specified.
- Use the IPv6 autoconfiguration mechanism. Only available for IPv6.

AT#IPCFG – IPv4 or IPv6 configuration	
Test Command	AT#IPCFG=?
Response	#IPCFG: <context_id>,<ip_mode>,<ip_address>[,<ipv6_prefix_length>] OK
Read Command	AT#IPCFG?
Response	#IPCFG: <context_id>,<ip_status>[,<ipv4_address>[,<ipv6_address>]] OK
Set Command	AT#IPCFG=<context_id>,<ip_mode>,<ip_address>[,<ipv6_prefix_length>]
Response	OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<context_id>,<ip_mode>,<ip_status>,<ip_address>,<ipv4_address>,<ipv6_address>,<ipv6_prefix_length>
<context_id>	Integer type Context number established on modem side.
<ip_mode>	Integer type 0: Static/Manual IP configuration. Used when the IP address is provided by the network. This address can be read using the AT command AT+CGPADDR 1: DHCP mode. Address is obtained from a DHCP server. This mode is not supported in this version of firmware. 2: IPv6 auto configuration mode. Only available for IPv6.
<ip_status>	Integer type 0: The IP address is down due to the deactivation of PDP context 1: The IP address is active and its related PDP context is also active 2: The IP address is deleted because a new PDP context with different address is active
<ip_address> <ipv4_address> <ipv6_address>	String type. IPv4 or IPv6 address provided by the network: IPv4 example: 192.168.10.30 IPv6 example: cafe:baba::01
<ipv6_prefix_len h>	Integer type Length, in number of bit, of the prefix. Only used in case of IPv6.
<err>	See Error List

Max Response Time	1 sec
Parameter Saving Mode	NA
Reference	Custom

3.1 AT#IPDEL – Delete IP configuration

AT#IPDEL – Delete IP configuration	
Test Command	AT#IPDEL=?
Response	#IPDEL: <context_id> OK
Set Command	AT#IPDEL: <context_id>
Response	OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<context_id>
<context_id>	Integer type Context number established on modem side.
<err>	See Error List
Max Response Time	1 sec
Parameter Saving Mode	NA
Reference	Custom

3.2 AT#IPPARAMS – IP stack configuration

This command is used to configure the IP stack parameters before using it.

AT#IPPARAMS – IP stack configuration	
Test Command	AT#IPPARAMS=?
Response	#IPPARAMS: <auto ip>,<preferred ip version>,<max ipstack periodicity>,<auto ip timeout> OK
Read Command	AT#IPPARAMS?
Response	#IPPARAMS: <auto_ip>,<preferred_ip_version>,<max_ipstack_periodicity>,<auto_ip_timeout> OK
Set Command	AT#IPPARAMS=<auto_ip>,<preferred_ip_version>,<max_ipstack_periodicity>,<auto_ip_timeout>
Response	OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<auto_ip>,<preferred_ip_version>,<max_ipstack_periodicity>,<auto_ip_timeout>
<auto_ip>	Integer type 0: Automatic configuration of IP link not activated. 1: Automatic configuration of IP link activated. When activated, the stack will, by itself, configure the IPv4 and IPv6 addresses and parameters with the information provided by the network.
<preferred_ip_version>	Integer type 0: IPv4 is the preferred stack 1: IPv6 is the preferred stack. The preferred stack will be used by default in case the parameter provided by the user cannot help to decide.
<max_ipstack_periodicity>	Integer type Max Timer in second for the IP stack periodicity. By default, it is set to 0xFFFF (no periodic wake-up is supported unless except it is managed in the Customer Application mode)
<auto_ip_timeout>	Integer type Timer in second between the module startup and attached PS attachment state for entering in sleep
<err>	See Error List
Max Response Time	1 sec
Parameter Saving Mode	SAVED to NVM using AT#RESET=1 Takes effect after module reboot.
Reference	Custom

3.3 AT#IPPING – ICMPv4/v6 request (ping)

This command is used to send IPv4 or IPv6 ping to a remote device, using an already established IP context.

AT#IPPING – ICMPv4/v6 request (ping)	
Test Command	AT#IPPING=?
Response	#IPPING: <context_id>,<ip_address> [,<timeout>,<number_of_bytes>,<number_of_ping>] OK
Set Command	AT#IPPING=<context_id>,<ip_address>[,<timeout>,[<number_of_bytes>,[<number_of_ping>]]]
Response	1,<result>,<nb_bytes_transmitted>,<transmission_time> 2,<result>,<nb_bytes_transmitted>,<transmission_time> … <number_of_ping>,<result>,<nb_bytes_transmitted>,<transmission_time> #IPPING:<nb_ping_sent>,<nb_ping_received>,<nb_ping_lost>, <min_transmission_time>,<max_transmission_time>, <average_transmission_time> OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<context_id>,<ip_address>,<timeout>,<number_of_bytes>,<number_of_ping>, <result>,<nb_bytes_transmitted>,<transmission_time>,<nb_ping_sent>, <nb_ping_received>,<nb_ping_lost>,<min_transmission_time>, <max_transmission_time>,<average_transmission_time>
<context_id>	Integer type Context number established on modem side.
<ip_address>	String type IPv4 or IPv6 address of the remote target.to ping.
<timeout>	Integer type Timeout value waiting for the ICMP answer (1 to 120 seconds) This parameter is optional. By default it is set to 60 seconds
<number_of_bytes>	Integer type Number of bytes in the ICMP request. The minimum value is 1 and the maximum is 450. This parameter is optional. By default it is set to 10.
<number_of_ping>	Integer type Number of ping to transmit. Minimum number is 1 and maximum is 20. This parameter is optional. By default it is set to 1.
<result>	Integer type Result of the transmission of an ICMP message. Can take the values: 0: Ping OK 1: Ping KO – No answer to ICMP request received before timeout. 2: Bad sequence number received. 3: ICMP transmission failed
<nb_bytes_ transmitted>	Integer type Number of data correctly transmitted in the ICMP packet
<transmission_time>	Integer type Transmission time in milliseconds. In case the ICMP answer is not received before timeout, the returned transmission time is 999999

<nb_ping_sent>	Integer type Final status - Number of ping transmitted by the module
<nb_ping_received>	Integer type Final status - Number of ICMP answer received by the module
<nb_ping_lost>	Integer type Final status - Number of ping transmitted without answer received
<min_transmission_time>	Integer type Final status – Minimum transmission time in milliseconds
<max_transmission_time>	Integer type Final status – Maximum transmission time in milliseconds
<average_transmission_time>	Integer type Final status – Average transmission time in milliseconds
<err>	See Error List
Max Response Time	<timeout> seconds
Parameter Saving Mode	NA
Reference	Custom

3.4 AT#IPREAD – Read data available in TCP or UDP receiving buffer

AT#IPREAD – Read data available in TCP or UDP receiving buffer	
Test Command	AT#IPREAD=?
Response	#IPREAD: <context_id>,<socket_id> OK
Set Command	AT#IPREAD=<context_id>,<socket_id>
Response	<data> OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<context_id>,<socket_id>
<context_id>	Integer type Context number established on modem side. The IP stack has been configured for this context with the AT#IPCFG command.
<socket_id>	Integer type Socket number created on this context with the AT#SOCKETCREATE command.
<err>	See Error List
Max Response Time	30 sec
Parameter Saving Mode	NA
Reference	Custom

3.5 AT#IPSENDTCP – Send data over an established socket using TCP protocol

This command is used to send data in TCP mode using a previously established IP socket.

AT#IPSENDTCP – Send data over an established socket using TCP protocol	
Test Command	AT#IPSENDTCP=?
Response	#IPSENDTCP: <context id>,<socket id>,<data type>,[<data length>,][<data>] OK
Set Command	AT#IPSENDTCP=<context_id>,<socket_id>,<data_type>,[<data_length>,][<data>]
Response	OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<context_id>,<socket_id>,<data_type>,<data_length>,<data>
<context_id>	Integer type Context number established on modem side. The IP stack has been configured for this context with the AT#IPCFG command.
<socket_id>	Integer type Socket number created on this context with the AT#SOCKETCREATE command for a TCP connection. The TCP link shall be established before calling this command.
<data_type>	Integer type Type of data to transmit: 0: ASCII String as next parameter 1: Binary data. Next parameter is the data length followed by <CR><LF> Data in binary format is entered just after. 2: Hex data in text mode: 01A34B… = 0x01 0xA3 0x4B
<data_length>	Integer type Number of data. Only present for binary mode. Can take value from 1 to 512.
<data>	String or Hex or Binary type Maximum size is 512 bytes.
<err>	See Error List
Max Response Time	10 sec
Parameter Saving Mode	NA
Reference	Custom

3.6 AT#IPSENDUDP – Send data using UDP protocol.

This command is used to send data in UDP mode using a previously established IP link.

AT#IPSENDUDP – Send data using UDP protocol.	
Test Command	AT#IPSENDUDP=?
Response	#IPSENDUDP: <context_id>,<socket_id>,<ip_address>,<port>,[<rai>],<data_type>,[<data_length>,][<data>] OK
Set Command	AT#IPSENDUDP=<context_id>,<socket_id>,<ip_address>,<port>,[<rai>],<data_type>,[<data_length>,][<data>]
Response	OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<context_id>,<socket_id>,<ip_address>,<port>,<rai>,<data_type>,<data_length>,<data>
<context_id>	Integer type Context number established on modem side. The IP stack has been configured for this context with the AT#IPCFG command.
<socket_id>	Integer type Socket number created on this context with the AT#SOCKETCREATE command for an UDP communication.
<ip_address>	String type IPv4 or IPv6 address of the remote target.
<port>	Integer type Port number
<rai>	Integer type Specifies release assistance information: 0: No information available (or none of the other options apply) 1: TE will send only 1 UL packet and no DL packets expected 2: TE will send only 1 UL packet and only 1 DL packet expected
<data_type>	Integer type Type of data to transmit: 0: ASCII String as next parameter 1: Binary data. Next parameter is the data length follow by <CR><LF> Data in binary format is entered just after. 2: Hex data in text mode: 01A34B… = 0x01 0xA3 0x4B
<data_length>	Integer type Number of data. Only present for binary mode. Can takes value from 1 to 512
<data>	String or Hex or Binary type Maximum size if 512 bytes.
<err>	See Error List
Max Response Time	10 sec
Parameter Saving Mode	NA
Reference	Custom

3.7 AT#SOCKETCLOSE – Close a socket identify by the context number

AT#SOCKETCLOSE – Close a socket identify by the context number	
Test Command	AT#SOCKETCLOSE=?
Response	#SOCKETCLOSE: <context_id>,<socket_id> OK
Set Command	AT#SOCKETCLOSE=<context_id>,<socket_id>
Response	OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<context_id>,<socket_id>
<context_id>	Integer type Context number established on modem side. The IP stack has been configured for this context with the AT#IPCFG command.
<socket_id>	Integer type Socket number created on this context with the AT#SOCKETCREATE command. If a TCP link is connected on this socket, it will be automatically closed.
<err>	See Error List
Max Response Time	60 sec
Parameter Saving Mode	NA
Reference	Custom

3.8 AT#SOCKETCREATE – Create a new socket on top of an existing IPv4/v6 link

This command is used to configure and activate the IPv4 or IPv6 protocol stack on top of an already established PDP context.

AT#SOCKETCREATE – Create a new socket on top of an existing IPv4/v6 link	
Test Command	AT#SOCKETCREATE=?
Response	#SOCKETCREATE:<context_id>,<ip_version>,<socket_type>,[<udp_port>,]<send_timeout>,<receive_timeout>[,<frame_received_urc>,<security_profile_id>] OK
Read Command	AT#SOCKETCREATE?
Response	#SOCKETCREATE: <context_id>,<socket_id>,<socket_status>,<socket_type>,<udp_port>,<send_timeout>,<receive_timeout>,<frame_received_urc>,<security_profile_id> … <context_id>,<socket_id>,<socket_status>,<socket_type>,<udp_port>,<send_timeout>,<receive_timeout>,<frame_received_urc>,<security_profile_id> OK
Set Command	AT#SOCKETCREATE=<context_id>,<ip_version>,<socket_type>,[<udp_port>,]<send_timeout>,<receive_timeout>[,<frame_received_urc>,<security_profile_id>]
Response	#SOCKETCREATE:<socket_id> OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<context_id>,<socket_id>,<ip_version>,<socket_status>,<socket_type>,<udp_port>,<send_timeout>,<receive_timeout>,<frame_received_urc>,<security_profile_id>
<socket_status>	Integer type 0: socket not assigned 1: socket not activated 2: activating socket 3: socket activated 4: UDP socket created 5: raw link 6: TCP socket not connected 7: TCP socket connected
<context_id>	Integer type Context number established on modem side. The IP stack has been configured for this context with the AT#IPCFG command.
<socket_id>	Integer type ID of the new created socket. This number starts with 0 and is incremented each time a new socket is created. The maximum number of sockets supported by the system is 3: one TCP socket and two UDP sockets. This <socket_id> shall be indicated when using the socket for data transmission or to close the socket.
<ip_version>	Integer type The IP version to use for the socket: 0: IPv4

	1: IPv6 This parameter shall be in line with the IP link configuration. If IPv4 version is requested and the IP address of the link is IPv6, or if IPv6 is requested for IPv4 link, an error will be reported.
<socket_type>	String type The type of socket to create: "TCP": TCP communication "UDP": UDP communication "RAW": Data RAW communication
<udp_port>	Integer type The port number to open for UDP reception. This parameter is present only in case of UDP protocol.
<send_timeout>	Integer type Maximum time needed for transmission of packet over the air and reception of acknowledgement.
<receive_timeout>	Integer type Maximum time needed for transmission of packet over the air.
<frame_received_urc>	Integer value (0 or 1) Indicates if an URC (#IPRECV) shall be generated when TCP or UDP data are available and can be read with AT#IPREAD command.
<security_profile_id>	Integer type Optional parameter If specified and if the security profile exists, a TLS link will be used for the socket.
<err>	See Error List
Max Response Time	2 sec
Parameter Saving Mode	NA
Reference	Custom

3.9 AT#TCPCONNECT – Establish a TCP link using a configured IP socket

This command is used to establish a TCP connection using a previously established IP socket. The socket is identified by its context number.

AT#TCPCONNECT – Establish a TCP link using a configured IP socket	
Test Command	AT#TCPCONNECT=?
Response	#TCPCONNECT: <context_id>,<socket_id>,<ip_address>,<port_number> OK
Read Command	AT#TCPCONNECT?
Response	# TCPCONNECT: <context_id>,<socket_id>,<status> OK
Set Command	AT#TCPCONNECT=<context_id>,<socket_id>,<ip_address>,<port_number>
Response	OK
If there is any error	ERROR Or +CME ERROR:<err>
Parameter	<context_id>,<socket_id>,<ip_address>,<port_number>
<context_id>	Integer type Context number established on modem side. The IP stack has been configured for this context with the AT#IPCFG command.
<socket_id>	Integer type Socket number created on this context with the AT#SOCKETCREATE command.
<ip_address>	String type IPv4 or IPv6 address of the remote target.
<port_number>	Integer type Port number
<status>	Integer type Status of the TCP link: 3: TCP not connected 4: TCP connecting 5: TCP connected
<err>	See Error List
Max Response Time	10 sec
Parameter Saving Mode	NA
Reference	Custom

4 IP Configuration

4.1 Preamble

To set an IP connection with TCP or UDP packets exchanges, the ME shall be attached first (PDP context connection established) to a NB-IOT network.
The platform, after boot, should perform this automatically.

This is notified by message: **+CGEV: ME PDN ACT 5** to the Host.
PS: 5 is the context ID used by default with ST87M01 platform

4.2 IP default parameters command

One AT command is used to read or set IP configuration default parameters.
This AT cmd can manage 4 parameters:

<auto_ip>	Integer type 0: Automatic configuration of IP link not activated. 1: Automatic configuration of IP link activated. When activated, the stack will, by itself, configure the IPv4 and IPv6 addresses and parameters with the information provided by the network.
<preferred_ip_version>	Integer type 0: IPv4 is the preferred stack 1: IPv6 is the preferred stack. The preferred stack will be used by default in case the parameter provided by the user cannot help to decide.
<max_ipstack_periodicity>	Integer type Max Timer in second for the IP stack periodicity. By default, it is set to 0xFFFF (no periodic wake-up is supported unless except it is managed in the Customer Application mode)
<auto_ip_timeout>	Integer type Timer in second between the module startup and attached PS attachment state for entering in sleep

The ME can use this command to retrieve the IP default configuration:

Example of result:

```
at#ipparams?

#IPPARAMS: 1,0,65535,60

OK
```

The same AT command is used also to set preferences of IP connection.
This AT cmd can manage 4 parameters:

Set Command	AT#IPPARAMS=<auto_ip>,<preferred_ip_version>,<max_ipstack_periodicity>,<auto_ip_timeout>

This command is used to configure the IP stack parameters before using it. Parameters configured by this AT command are saved in non-volatile RAM and are retrieved after reset.
If you want to save your configuration, don't forget to send AT#RESET=1 after.

NB: After boot and by default the IP configuration is set to auto and most of time you will have after boot the platform this kind of UART trace:

```
#SIMST: 1    <----- SIM detected

+CEREG: 2      <----- Registration in progress

+CEREG: 0   <----- Registration stopped.

+CEREG: 2    <----- Registration in progress

+CGEV: ME PDN ACT 5     <----- Attached to the network with ID n°5

+CEREG: 1   <----- Registered URC

#IPCFG: 5,0,1    <----- IP configuration set automatically by the platform
```

4.3 Set IPv4 or IPv6 configuration:

To start TCP or UDP communication, The host shall perform first the IP configuration.
Then, if the ME is not configured to use automatic IP (see Paragraph 4.2), the ME shall retrieve its IP address by using:

```
at+cgpaddr=5
```

The answer from network could be something like :

```
+CGPADDR: 5,"100.115.11.113","2A02:8440:5202:65D1::0029:28FD:B001"
OK
```

- first parameter (here: 5) is the default context ID.
- second parameter ("100.115.11.113") is the IPV4 address.
- third parameter ("2A02:8440:5202:65D1::0029:28FD:B001") is the IPV6 address.

Depending on the network, it is possible to only retrieved one of the two addresses.

Then, the ME shall activate his configuration with command **at#ipcfg**

For IPV4:
at#ipcfg=5,0,<UE IPV4 address>

For example, with: ***at#ipcfg=5,0,100.115.11.113***

If everything goes right, The ME will receive this answer:

```
IP Network is Up for context 5
```

Once done, to test the IP connection, the ME can ping Google with following command:

```
at#ipping=5,8.8.8.8

1,0,10,3535

2,0,10,203

3,0,10,193

4,0,10,185

5,0,10,192

#IPPING: 5,5,0,185,3535,861

OK
```

For IPV6:

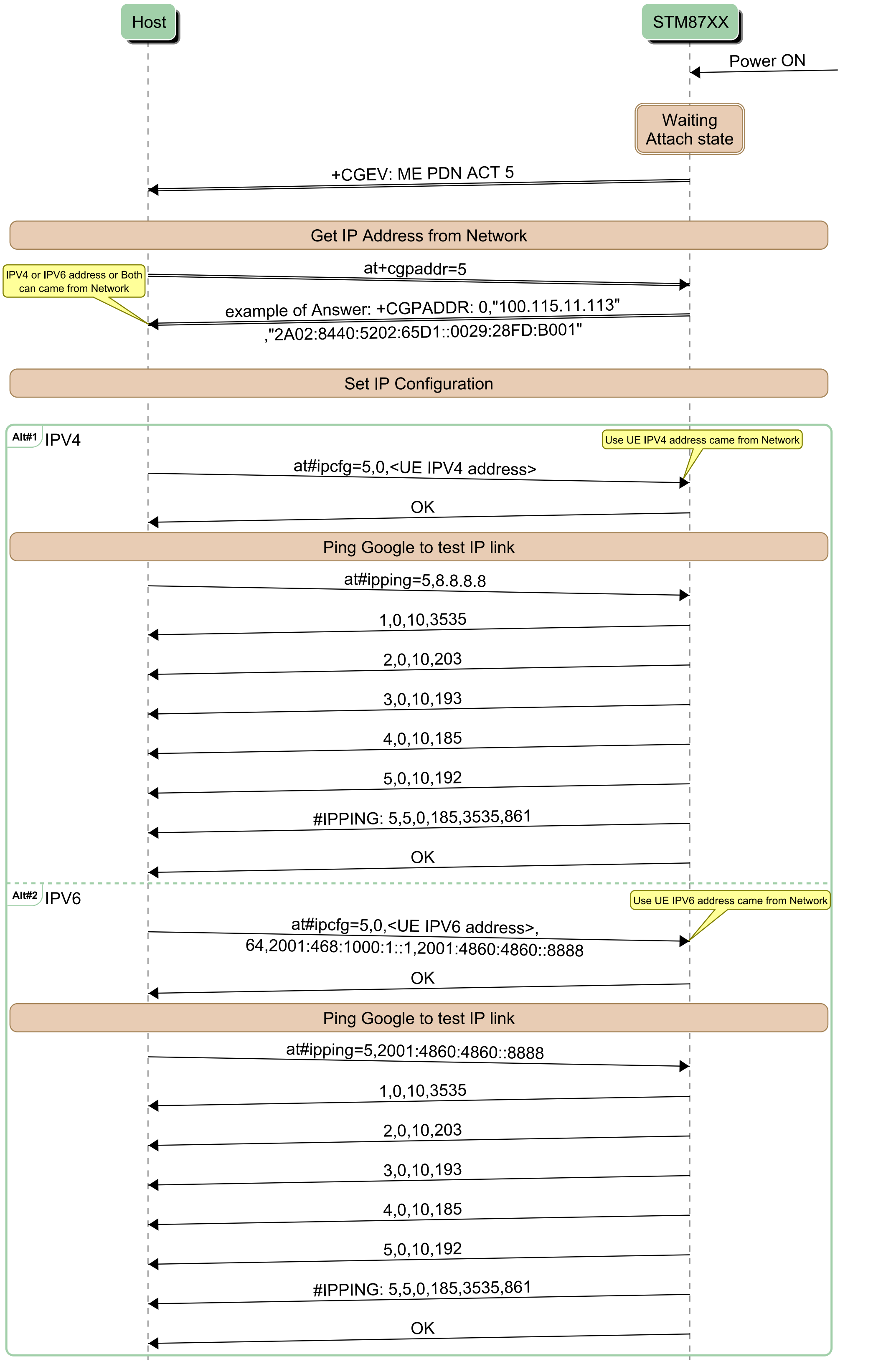
at#ipcfg=5,0,<UE IPV6 address>,64,2001:468:1000:1::1,2001:4860:4860::8888

for example:
at#ipcfg=5,0,2A02:8440:4312:60EF::0045:8437:0901,64,2001:468:1000:1::1,2001:4860:4860::8888

If everything goes right, the ME will receive following answer:
IP Network is Up for context 5

Once done, you can ping Google in IPV6 with following command:
at#ipping=5,2001:4860:4860::8888

4.4 MSC for IP configuration


Host
STM87XX
Power ON
Waiting
Attach state
+CGEV: ME PDN ACT 5
Get IP Address from Network
at+cgpaddr=5
IPV4 or IPV6 address or Both can came from Network
example of Answer: +CGPADDR: 0,"100.115.11.113"
,"2A02:8440:5202:65D1::0029:28FD:B001"
Set IP Configuration
Alt#1 IPV4
Use UE IPV4 address came from Network
at#ipcfg=5,0,<UE IPV4 address>
OK
Ping Google to test IP link
at#ipping=5,8.8.8.8
1,0,10,3535
2,0,10,203
3,0,10,193
4,0,10,185
5,0,10,192
#IPPING: 5,5,0,185,3535,861
OK
Alt#2 IPV6
Use UE IPV6 address came from Network
at#ipcfg=5,0,<UE IPV6 address>,
64,2001:468:1000:1::1,2001:4860:4860::8888
OK
Ping Google to test IP link
at#ipping=5,2001:4860:4860::8888
1,0,10,3535
2,0,10,203
3,0,10,193
4,0,10,185
5,0,10,192
#IPPING: 5,5,0,185,3535,861
OK

https://gitlab.com/msc-generator v8.4

5 TCP Use Cases

5.1 Preamble

For TCP connection and for TCP packets exchange, the ME shall set the IP layer if the AutoIP config is not set by default.

See Chapter 4: IP configuration above.
Now, the ME can start a TCP connection with a TCP server.

We assumed that the customer who wants to test this TCP functionality has a TCP server running correctly inside the Internet network.

5.2 Create TCP socket

To exchange, TCP packets, we need first to create a socket.
It will be used to exchange data:

For example the ME can create socket with the following at command:

at#socketcreate=5,0,TCP,1000,1,1,1

5: is the context ID used for IP connection: 5 it is the one used by default by the ST87M01 platform.
0: for IPV4, **1**: for IPV6   it should be the same chosen during IP configuration
TCP is the type of socket used to create it.
1 is the timeout set in seconds to send to the server the request.
1 is the timeout set in seconds to send and receive the response from the server.
1 is enabling the URC reception when TCP data are available and can be read with AT#IPREAD

#SOCKETCREATE: 0   ------> socket is created with ID 0
OK

5.3 Connect TCP socket

After the creation of the socket, the ME can connect the TCP layer to the remote server

at#tcpconnect=5,0,< IP Remote Server address> ,< TCP Remote Server port >

5: is the context ID used for IP connection: 5 it is the one used by default by the ST87M01 platform
0: is the Socket ID received during socket creation.

5.4 Send data / Receive Data

The socket is now connected from ME to server,
The data socket can be used.

To send data from UE to TCP server :

at#ipsendtcp=5,0,0,ST87M NBIOT TCP test

5: is the context ID used for IP connection: 5 it is the one used by default by the ST87M01 platform.
0: Is the Socket ID received during socket creation
0: Data type is ASCII

If you have enabled the “URC generation bit” param during the creation of the socket, you will receive Notification URC when data will be available

```
#IPRECV: 5,0
```

To read inside receive data buffer to get data from server :

```
at#ipread=5,0
```

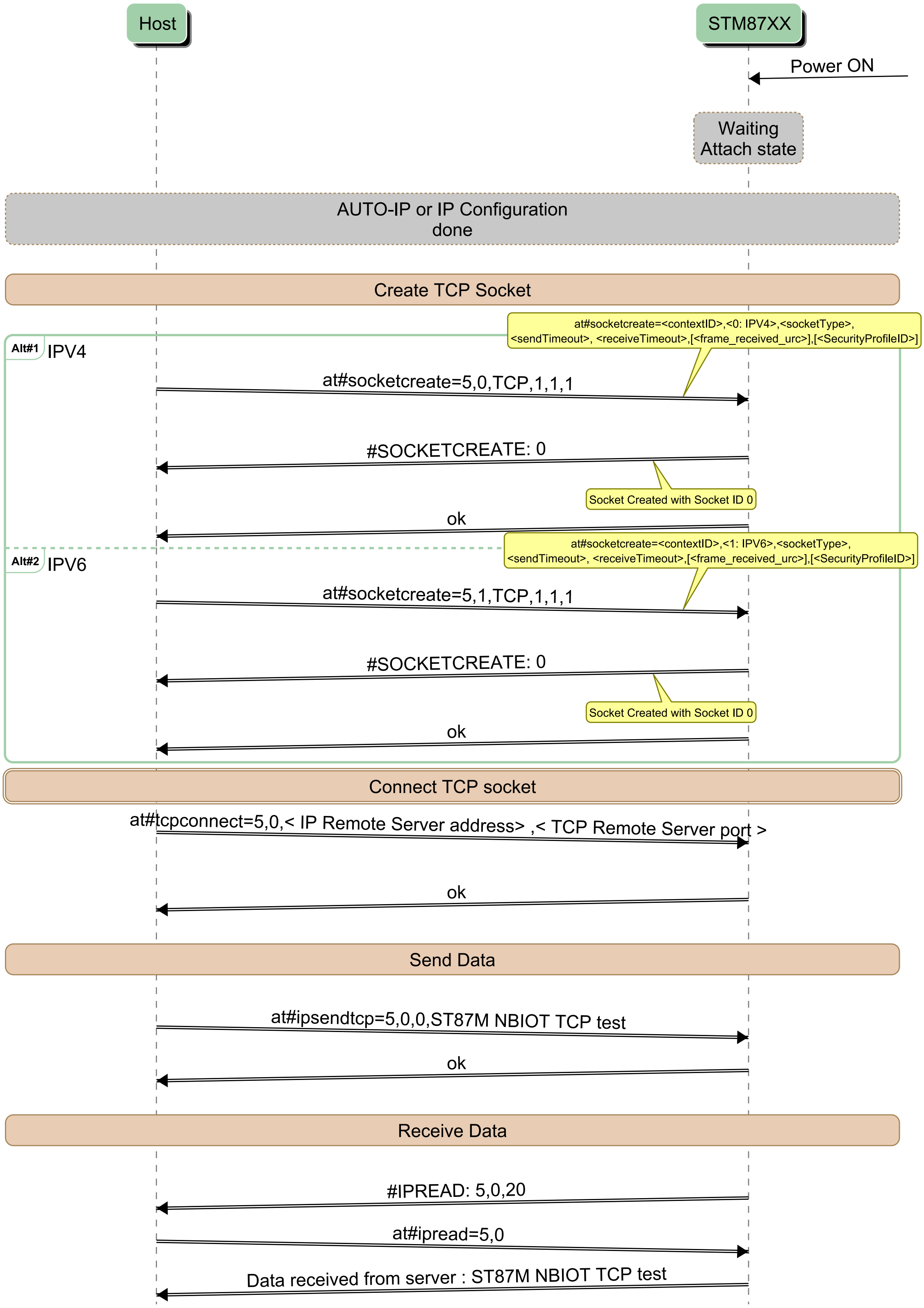
5: is the context ID used for IP connection: 5 it is the one used by default by the ST87M01 platform.
0: Is the Socket ID received during socket creation

```
#IPREAD: 5,0,20
ST87M NBIOT TCP test

OK
```

NB : For our Example , the answer of this command will be the same as the TCP sent packet , If you are using an TCP test server with Loopback feature (Received packet sent back to the sender)

5.1 MSC for TCP use cases



at#socketcreate=<contextID>,<0: IPV4>,<socketType>,<sendTimeout>, <receiveTimeout>,[<frame_received_urc>],[<SecurityProfileID>]

at#socketcreate=5,0,TCP,1,1,1

#SOCKETCREATE: 0

Socket Created with Socket ID 0

ok

at#socketcreate=<contextID>,<1: IPV6>,<socketType>,<sendTimeout>, <receiveTimeout>,[<frame_received_urc>],[<SecurityProfileID>]

at#socketcreate=5,1,TCP,1,1,1

#SOCKETCREATE: 0

Socket Created with Socket ID 0

ok

at#tcpconnect=5,0,< IP Remote Server address> ,< TCP Remote Server port >

ok

at#ipsendtcp=5,0,0,ST87M NBIOT TCP test

ok

#IPREAD: 5,0,20

at#ipread=5,0

Data received from server : ST87M NBIOT TCP test

https://gitlab.com/msc-generator v8.4

6 UDP Connection

6.1 Preamble

For UDP connection and for UDP packets exchange, the ME shall set the IP layer if the AutoIP config is not set by default.

See Chapter 4: IP configuration above.

Now, the ME can start an UDP connection with an UDP server.

It is assumed that the customer who wants to test this UDP functionality has a UDP server running correctly inside the Internet network.

6.2 Create UDP socket

UDP transmission does not need creation of a socket if the UE just wants to send some data to server without response.
This is needed only in case of send/receive data of UDP.

<u>For example the ME create socket with the following at command:</u>

at#socketcreate=5,0,UDP,5000,1,1,1

5: is the context ID used for IP connection: 0 it is the fist one. You have to use the same as IP configuration.
0: for IPV4, **1**: for IPV6 (it should be the same chosen during IP configuration)
UDP is the type of socket used to create it.
5000: ss the port number used by UDP server to exchange data
1: is the timeout set in seconds to send to the server the request.
1: is the timeout set in seconds to send and receive the response from the server.
1: is activating if customer needs an URC (#RECV) when TCP or UDP data is received from the network and can be read with AT#IPREAD cmd

#SOCKETCREATE: 0 ------> socket is created with ID 0
OK

6.3 Connect UDP socket

The UDP Transmission is a non-connected protocol.
Connect UDP socket is not needed.

6.4 Send data / Receive Data

To send data from UE to UDP server :

```
AT#IPSENDUDP=5,0,<xxx.xxx.xxx.xxx>,5000,0,0,ST87M NBIOT UDP test
```

5: is the context ID used for IP connection
0: is the Socket number used during socket creation
<xxx.xxx.xxx.xxx>: Ip Address of UDP server
5000: Port number of UDP server
0: Data type set to ASCII
ST87M NBIOT UDP test: String to send to UDP server

If data is coming from Server and if during socket creation, the customer has enabled the URC reception, the host will receive this URC when incoming data will be available:

```
#IPRECV: 5,0
```

5: is the context ID used for IP connection
0: is the socket ID used for socket creation

To read inside receive data buffer to get data from server :

```
at#ipread=5,0
```

5: is the context ID used for IP connection
0: is the socket ID used for socket creation

#IPREAD: 5,0,20 ------> URC received to notify incoming read data

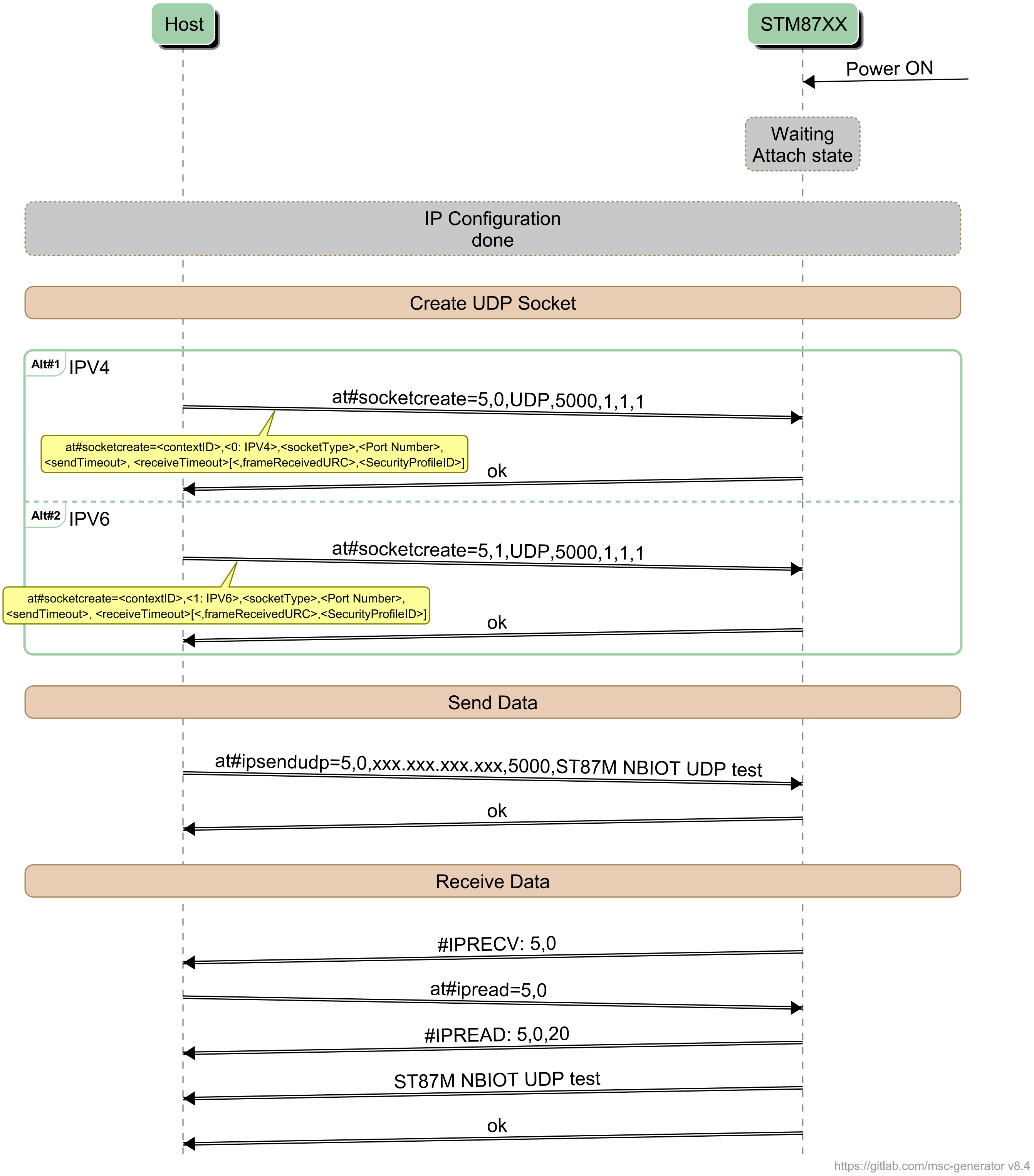
5: is the context ID used for IP connection
0: is the socket ID used for socket creation
20: the size of data received

ST87M NBIOT UDP test ------> ASCII data

OK

NB : For our Example , the answer of this command will be the same as the TCP sent packet , If you are using an TCP test server with Loopback feature (Received packet sent back to the sender)

6.5 MSC for UDP use cases


Host
STM87XX
Power ON
Waiting
Attach state
IP Configuration
done
Create UDP Socket
Alt#1 IPV4
at#socketcreate=5,0,UDP,5000,1,1,1
at#socketcreate=<contextID>,<0: IPV4>,<socketType>,<Port Number>,
<sendTimeout>, <receiveTimeout>[<,frameReceivedURC>,<SecurityProfileID>]
ok
Alt#2 IPV6
at#socketcreate=5,1,UDP,5000,1,1,1
at#socketcreate=<contextID>,<1: IPV6>,<socketType>,<Port Number>,
<sendTimeout>, <receiveTimeout>[<,frameReceivedURC>,<SecurityProfileID>]
ok
Send Data
at#ipsendudp=5,0,xxx.xxx.xxx.xxx,5000,ST87M NBIOT UDP test
ok
Receive Data
#IPRECV: 5,0
at#ipread=5,0
#IPREAD: 5,0,20
ST87M NBIOT UDP test
ok
https://gitlab.com/msc-generator v8.4

7 Error codes

7.1 Modem generic error code

Code	Comment
1	no connection to phone
2	phone-adaptor link reserved
3	operation not allowed
4	operation not supported
5	PH-SIM PIN required
6	PH-FSIM PIN required
7	PH-FSIM PUK required
10	SIM not inserted
11	SIM PIN required
12	SIM PUK required
13	SIM failure
14	SIM busy
15	SIM wrong
16	incorrect password
17	SIM PIN2 required
18	SIM PUK2 required
20	memory full
21	invalid index
22	not found
23	memory failure
24	text string too long
25	invalid characters in text string
26	dial string too long
27	invalid characters in dial string
30	no network service
31	network timeout
32	network not allowed - emergency calls only
40	network personalisation PIN required
41	network personalisation PUK required
42	network subset personalisation PIN required
43	network subset personalisation PUK required
44	service provider personalisation PIN required
45	service provider personalisation PUK required
46	corporate personalisation PIN required
47	corporate personalisation PUK required
48	hidden key required
49	EAP method not supported
50	Incorrect parameters

51	command implemented but currently disabled
52	command aborted by user
53	not attached to network due to MT functionality restrictions
54	modem not allowed - MT restricted to emergency calls only
55	operation not allowed because of MT functionality restrictions
56	fixed dial number only allowed - called number is not a fixed dial number
57	temporarily out of service due to other MT usage
58	language/alphabet not supported
59	unexpected data value
60	system failure
61	data missing
62	call barred
63	message waiting indication subscription failure
64	unknown
103	Illegal MS (#3)
106	Illegal ME (#6)
107	GPRS services not allowed (#7)
108	GPRS services and non-GPRS services not allowed (#8)
111	PLMN not allowed (#11)
112	Location area not allowed (#12)
113	Roaming not allowed in this location area (#13)
114	GPRS services not allowed in this PLMN (#14)
115	No Suitable Cells In Location Area (#15)
122	Congestion (#22))
125	Not authorized for this CSG (#25)
126	insufficient resources (#26)
127	missing or unknown APN (#27)
128	unknown PDP address or PDP type (#28)
129	user authentication failed (#29)
130	activation rejected by GGSN Serving GW or PDN GW (#30)
131	activation rejected unspecified (#31)
132	service option not supported (#32)
133	requested service option not subscribed (#33)
134	service option temporarily out of order (#34)
140	feature not supported (#40)
141	semantic error in the TFT operation (#41)
142	syntactical error in the TFT operation (#42)
143	unknown PDP context (#43)
144	semantic errors in packet filter(s) (#44)
145	syntactical errors in packet filter(s) (#45)
146	PDP context without TFT already activated (#46)
149	PDP authentication failure
171	Last PDN disconnection not allowed (#49)
172	Semantically incorrect message (#95)
173	Mandatory information element error (#96)

174	Information element non-existent or not implemented (#97)
175	Conditional IE error (#99)
176	Protocol error unspecified (#111)
177	Operator Determined Barring (#8)
178	maximum number of PDP contexts reached (#65)
179	requested APN not supported in current RAT and PLMN combination (#66)
180	request rejected Bearer Control Mode violation (#48)
181	unsupported QCI value (#83)

7.2 Modem custom error code

Code	Comment
500	Command not available
501	Number of param not correct
502	Exceed parameter number
503	Not enought parameter
504	Command unknown
505	Generic error

7.3 Secure core error code

Code	Comment
1000	Generic Error
1001	String too long
1002	Command analyzing error
1003	Execution error
1004	Bad number of parameter
1005	Error in parameter conversion
1020	EFLASH MCU write error
1021	EFLASH MCU erase, error
1022	NVM read error
1023	NVM write error
1024	NVM save error
1025	NVM erase error
1040	WRITE access error
1041	READ access error
1060	GPIO pin number error
1061	GPIO mode error
1062	GPIO value error
1063	GPIO bank error
1064	GPIO pin error
1065	GPIO mux error
1080	Regulator error
1100	Sleep error
1101	Clock error
1102	Calclock 32khz error
1103	Ccalclock rosc error
1104	Calclock don't exist
1105	Ram test error
1106	Flash test error
1107	Watchdog test error
1120	AT command forward to APP error
1121	AT command forward to Modem error
1140	Storage init error
1200	Assert fault
1205	Load acces fault
1206	Store acces fault
1300	Invalid PSA argument
1301	PSA allready exists
1302	PSA does not exist
1303	Insufficient PSA memory
1304	PSA generic error
1305	PSA not permitted
1306	PSA storage failure
1307	PSA invalid signature
1308	PSA storage cert ID error
1309	PSA invalid handle

7.4 Application core error code

Code	Comment
2000	system memory issue
2001	system queue send issue
2008	adc invalid parameter : channel
2009	adc : no answer
2010	system invalid parameter : time for PSM
2011	system invalid parameter : auto IP success timeout
2012	Invalid AT command
2100	tcpip invalid parameter : context id
2101	tcpip invalid parameter : ip mode
2102	tcpip invalid parameter : ip version
2103	tcpip invalid parameter : ip address
2104	tcpip invalid parameter : socket id
2105	tcpip : no more context available
2106	tcpip : network is down
2110	tcpip invalid parameter : ipv4 address
2111	tcpip invalid parameter : dhcp not supported
2120	tcpip invalid parameter : ipv6 address
2121	tcpip invalid parameter : ipv6 prefix
2122	tcpip invalid parameter : ipv6 prefix length
2123	tcpip invalid parameter : ipv6 dns
2124	tcpip invalid parameter : dhcp6 not supported
2130	tcpip ping not possible
2131	tcpip invalid parameter : nbr of bytes
2132	tcpip invalid parameter : nbr of ping
2133	tcpip invalid parameter : timeout
2140	tcpip invalid parameter : tcp udp
2141	tcpip invalid parameter : port number
2142	tcpip invalid parameter : data
2143	tcpip ip address missing
2144	tcpip invalid parameter : display number of data
2145	tcpip invalid parameter : data type
2146	tcpip invalid parameter : data length
2147	tcpip invalid parameter : RAI
2150	tcpip socket creation error
2151	tcpip socket already bound to different port
2152	tcpip socket error while connecting
2153	tcpip socket not opened
2154	tcpip socket issue during data transfer
2155	tcpip socket bind error
2156	tcpip socket invalid context
2157	tcpip socket already connected
2158	tcpip socket received error
2160	tcpip invalid parameter : tls algo type
2161	tcpip invalid parameter : security profile id
2162	tcpip error while setting tls credentials
2200	mqtt init failed

2201	mqtt protocol stack not initialized
2202	mqtt invalid parameter : client name
2203	mqtt invalid parameter : connexion timeout
2204	mqtt invalid parameter : protocol timeout
2205	mqtt invalid parameter : publish retry
2206	mqtt invalid parameter : keep alive pub msg
2207	mqtt invalid parameter : user
2208	mqtt invalid parameter : passwd
2209	mqtt invalid parameter : topic
2210	mqtt invalid parameter : message
2211	mqtt invalid parameter : subscribe
2212	mqtt invalid parameter : retry number
2213	mqtt invalid parameter : QoS
2214	mqtt connexion failed
2215	mqtt publish failed
2216	mqtt subscribe failed
2217	mqtt unsubscribe failed
2218	mqtt not connected
2219	mqtt disconnexion failed
2220	mqtt protocol stack already connected
2221	mqtt not configured
2300	http init issue
2301	http protocol stack not initialized
2302	http protocol stack already initialized
2303	http invalid parameter : method
2304	http invalid parameter : method path
2305	http request initialization failed
2306	http request send failed
2400	coap invalid parameter: client name
2401	coap invalid parameter: server address
2402	coap invalid parameter: message type
2403	coap invalid parameter: method
2404	coap invalid parameter: uri path
2405	coap invalid parameter: payload
2406	coap invalid parameter: command string
2410	coap invalid status: not configured
2411	coap invalid status: not started
2412	coap invalid status: not connected
2500	lwm2m invalid parameter: client name
2501	lwm2m invalid parameter: server address
2502	lwm2m invalid parameter: server port
2503	lwm2m invalid parameter: server id
2504	lwm2m invalid parameter: server type
2505	lwm2m invalid parameter: binding mode
2506	lwm2m invalid parameter: lifetime
2507	lwm2m invalid parameter: storing
2508	lwm2m invalid parameter: object id
2509	lwm2m invalid parameter: uri
2510	lwm2m invalid parameter: data
2511	lwm2m invalid parameter: command type
2512	lwm2m invalid parameter: command string
2513	lwm2m invalid parameter: register option
2514	lwm2m invalid stack status

2515	lwm2m failed to create device object
2516	lwm2m failed to create security object
2517	lwm2m failed to create server object
2518	lwm2m failed to create firmware object
2519	lwm2m failed to init protocol
2520	lwm2m general failure
2521	lwm2m failed to create custom object
2522	lwm2m max object num
2523	lwm2m failed to remove custom object
2524	lwm2m invalid resource
2525	lwm2m notify not initialized
2526	lwm2m invalid notify
2527	lwm2m notity timed out
2528	lwm2m notify mismatch
2600	fota invalid option
2601	fota invalid uri
2602	fota failed to download
2603	fota failed to abort download
2604	fota failed to update
2605	fota failed to cancel update
2700	gnss invalid mode
2701	gnss invalid const
2702	gnss invalid IP
2703	gnss invalid port
2704	gnss invalid timeout
2705	gnss invalid sate
2706	gnss invalid data
2707	gnss error start
2708	gnss invalid event
2709	gnss invalid format
2710	gnss invalid period
2711	gnss sv status fail
2712	gnss busy

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of
Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.