Universal IP (UIP)

AT Commands
Reference Guide

Universal IP (UIP) AT Commands Reference Guide for the following products:

SocketEthernet IP® (MT100SEM)

SocketModem® IP (MT5656SMI-IP) This product uses these commands when it is in IP mode.

SocketModem® GPRS IP (MTSMC-G-F4-IP)

MultiModem® iCell (MTCMR-G/C/H/E) This product uses these commands when it is in IP mode.

PN S000457E, Version E

Revisions

Revision Level	*Date*	*Description*
A	07/31/08	Initial release. This guide replaces S000426F.
B	08/06/08	Added Result Codes to #PERSISTENTSOCKET and updated its Defaults to include the result code default.
C	10/14/08	Added a section on GPIOs to Chapter 10. In General Commands added text that explained differences in functionality of an MT100SEM command and a non-MT100SEM for &F and &W commands. Changed the default values for #PPPMODE, &S, +IPR, and +IFC. Added a Note to #ANSWERMODE command. Added new command: #AUTOCONNECT.
D	10/21/08	Encased the <value> of some commands in double quotes. Changed the description of #POP3PORT to Set POP3 Server Port.
E	02/20/09	Added the iCell modem to the list of products that use this command set. Updated the syntax for the #AUTHENT command. Updated the +IPR command. Added a note after #DNSSERV2.

Note: This product used the FreeRTOS.org real time kernel. The FreeRTOS.org source code can be obtained by visiting http://www.FreeRTOS.org

Trademarks
Multi-Tech Systems, Inc. registered trademarks include ***SocketModem, MultiModem*** and the Multi-Tech logo.

World Headquarters
Multi-Tech Systems, Inc.
2205 Woodale Drive
Mounds View, Minnesota 55112
Phone: 763-785-3500 or 800-328-9717
Fax: 763-785-9874
Internet Address: http://www.multitech.com

Technical Support

Country	**By Email**	**By Phone**
Europe, Middle East, Africa:	support@multitech.co.uk	+(44) 118 959 7774
U.S., Canada, all others:	support@multitech.com	800-972-2439 or 763-717-5863

Contents

Chapter 1 – General AT Commands

General Commands

AT – Command Mode

Description: If the modem responds OK to the command **AT**, the modem is in command mode.

Z – Default Configuration

Description: This command restores the configuration profile from NVM. Settings are recalled on power up.
Syntax: **ATZ**
Response: OK (Command valid)

I – Identification

Description: Displays identification parameters.
Syntax: **ATI<value>**
Values: ATI**3** – Identifies the firmware version. Example: "1.00 to 1.00g" (MT100SEM example)
ATI**5** – Identifies the country. Example: "B5" (MT100SEM example)
Default: **0**

&C – RLSD (DCD) Option Turn On or Off

Description: The modem controls the RLSD output in accordance with the parameter supplied. This command sets the DCD On or Off.
Syntax: **AT&C<value>**
Values: **0** – Sets DCD to remain On at all times.
1 – Sets DCD to follow the physical connection.
2 – DCD follows the state of the TCP/UDP connection.
Default: **1**
Result Codes: OK
Otherwise ERROR

&D – DTR Option

Description: The modem reads the DTR to determine if the Data Terminal is Ready. This command allows the modem to act on the DTR.
Syntax: **AT&D<value>**
Values: **0** – DTR is ignored.
1 – An Active to Inactive transition disconnects an active TCP/UDP connection
2 – (For Non-MT100SEM) An active to inactive transition disconnects the physical connection
Default: **0**
Result Codes: OK
Otherwise ERROR

&F – Reset Factory Defaults

Description: **For MT100SEM:** This command replaces all settings with the factory defaults but does not store them to memory. Use &W to store them to memory.
For NON-MT100SEM: This command replaces the factory defaults of &D, &S, &C, E, and V only.

Syntax: **AT&F**

Values: None

&S – DSR ON or OFF

Description: Set DSR.

Syntax: **AT&Sx** where x=0,1

Values: AT&S0 Sets DSR to remain ON at all times
AT&S1 Sets DSR to follow DCD
AT&S2 DSR follows the state of the TCP/UDP connection

Display: AT&S? Displays the current setting or use AT#VALL to view all settings.

Default: **1**

Result Codes: OK
Otherwise ERROR

V – Terse or Verbose

Description: Sets terse or verbose responses.

Syntax: **ATVx** where x=0,1

Values: ATV0 Sets terse
ATV1 Sets verbose

Display: ATV? Displays the current **V** setting or use AT#VALL to view all settings.

Default: **ATV1**

&W – Store Factory Defaults to NVM

Description: **For MT100SEM:** Stores factory defaults to the NVM.
For NON-MT100SEM: Stores the current settings of &D, &S, &C, E, and V only.

Syntax: **AT&W**

Values: None

+IPR – Set Serial Speed

Description: Sets the serial speed.

Syntax: **AT+IPR=<serial speed value>**

Values: Valid speeds are:
For MT100SEM: **300** to **230400** and **0** (zero stands for autobaud)
For NON-MT100SEM: **300** to **921600**
Note: NON-MT100SEM does not support autobaud.

Display: AT+IPR? Displays the current serial speed.
AT+IPR=? Displays a list of all possible values (serial speeds).

Default: **For MT100SEM:** **0** (zero stands for autobaud)
For NON-MT100SEM: **115200**

+IFC – Set Flow Control

Description: Turns flow control on and off.

Syntax: **AT+IFC=<flow control value>**

Values:
0,0 Flow Control Off
2,2 Flow Control On

Default: **2,2** Flow Control On

+ICF – DTE-DCE Character Framing

Description: This command determines the local serial port start-stop (asynchronous) character framing that the DCE uses.

Syntax: **AT+ICF= <format value>[, <parity value>]**

Values:

<format>

0 Autodetect (not supported)
1 8 Data 2 Stop (supported)
<parity> parameter is ignored
2 8 Data 1 Parity 1 Stop (supported)
If no <parity> provided, 3 is used by default as <parity> value
3 8 Data 1 Stop (supported)
<parity> parameter is ignored
4 7 Data 2 Stop (supported)
<parity> parameter is ignored
5 7 Data 1 Parity 1 Stop (supported)
If no <parity> provided, 3 is used by default as <parity> value
6 7 Data 1 Stop (supported)
<parity> parameter is ignored

<parity>

0 Odd (supported)
1 Even (supported)
2 Mark (supported)
3 Space (supported)
4 None (supported)

Default:
Format: **2**
Parity: **4**
Default Example: AT+ICF= 2,4

Telnet Commands

#TELNET – Start/Stop Telnet Session

Description: Starts and stops a Telnet session.

Syntax: **AT#TELNET=<0,1>**

Values: AT#TELNET=**0** Stop Telnet session.
AT#TELNET=**1** Start Telnet session; wait for remote to Telnet in.

Display: AT#TELNET**?** Indicates whether a Telnet session has been started.

Default: **0**

#TELNETPORT– Set Telnet Port

Description: Sets Telnet port.

Syntax: **AT#TELNETPORT = <port number>**

Values: 1 to 65535

Display: AT#TELNETPORT? or AT#VTELNET or AT#VALL Displays the port number or all Telnet settings or all values.

Default: **23**

#TELNETUSER – Set Telnet User

Description: Sets Telnet user.

Syntax: **AT#TELNETUSER= "<64 char string>"**

Display: AT# TELNETUSER? or AT#VTELNET or AT#VALL Displays the user (64 char string) or all Telnet settings or all values.

Default: **admin**

#TELNETPASSWORD – Set Telnet Password

Description: Sets Telnet password.

Syntax: **AT# TELNETPASSWORD = "<64 char string>"**

Display: AT# TELNETPASSWORD? or AT#VTELNET or AT#VALL Displays the password (64 char string) or all Telnet settings or all values.

Default: **" "**

#VTELNET – View Telnet Settings

Description: Displays all Telnet settings.

Syntax: **AT# VTELNET**

Display: Displays all Telnet settings.

Chapter 2 – SMTP/POP3 Commands

Set Commands

#SENDERNAME – Set Sender Name

Definition: The sender's literal name (different from the SENDERADDR parameter, which is the sender's email address). This parameter will appear in the header of the email sent by the TCP/IP stack software, in the field: "From:"

Syntax: **Set Value: AT#SENDERNAME="<value>"**

View Value: AT#SENDERNAME? or AT#VSMTP, AT#VALL

Values: Alphanumeric ASCII text string up to 120 characters. The address must be provided in literal format (for instance, "machine 245").

Default: None

#SENDERADDR – Set Sender Address

Definition: To send emails, the TCP/IP stack software must know the email address of the sender. The "sender" is the email identification of the hardware platform itself or the optional attached equipment. This email address will appear in the header of the email sent by the TCP/IP stack software, in the field "From:"

Syntax: **Set Value: AT#SENDERADDR="<value>"**

View Value: AT#SENDERADDR? or **AT#VSMTP, AT#VALL**

Values: Alphanumeric ASCII text string up to 60 characters. The address must be provided in literal format (for instance xxxxxxx@web.zyx).

Default: None

#CCREC1 / CCREC2 / CCREC3 – Set Additional Recipient CC

Definition: The software can send email messages to an additional recipient as a "carbon copy". This parameter contains the email address of the additional recipient. This email address will appear in the header of the email sent by the TCP/IP stack software in the field "Cc:"

For a given value n, the CCRECn parameter is directly associated with the RECn parameter.

Syntax: **Set Value: AT#CCRECi="<value>"**

View Values: **AT#CCRECi?** (replace i with 1, 2, or 3) or **AT#VMAILi, AT#VALL**

Values: Alphanumeric ASCII text string up to 120 characters. The address must be provided in literal format (for instance xxxxx@web.com).

Default: None

#DOMAIN – Set Domain Name of the Sender

Definition: When sending an email message, the TCP/IP stack software must provide the SMTP server with the domain name of the sender. In some cases, this domain name may be different from the domain name included in the sender's email address.

Syntax: **Set Value: AT#DOMAIN="<value>"**
View Value: AT#DOMAIN? or AT#VSMTP, AT#VALL

Values: Alphanumeric ASCII text string up to 120 characters.

Default: None

#REC1, REC1ADD / REC2, REC2ADD / REC3, REC3ADD – Set Email Address

Definition: To send email messages, the TCP/IP stack software must know the e-mail address of the recipient. This email address will appear in the email header sent by the TCP/IP stack software, in the field 'To:'
The RECi parameter can hold a maximum of 10 email addresses, each email address being at the most 120 characters long.

Syntax: **Set Value: AT#RECi="<Value>"**
View Value: AT#REC? (replace i with 1, 2, or 3) or **AT#VMAILi, AT#VALL**

Values: RECi (i = 1, 2, or 3)
Alphanumeric ASCII text string up to 120 characters. The address must be provided in literal format (for instance xxxxx@company.com).

Setting One Email Address / Resetting the Parameter / Getting:
Set value / reset the parameter: AT#RECi="<Value>" (replace i with 1, 2, or 3)
View value: AT#REC? (replace i with 1, 2, or 3) or AT#VMAILi, AT#VALL

Setting One to Ten Email Address to the RECi Parameter / Resetting the Parameter:
To set 1 to 10 email addresses to the RECi parameter, enter the AT#RECiADD<CR> overwriting command.
Each email address has to be an alphanumeric ASCII text string, in literal format (for instance, dev12345678@web.xyz). To add another email address, enter the <CR><LF> pair. To end the setting, enter the following character: 1A (in ASCII code), generated in a keyboard by CTRL+Z escape sequence.

Note 1: This command overwrites all previously set email addresses of the RECi parameter.

Note 2: If more than ten addresses are entered, the 11th and subsequent addresses until the end character will be ignored.

Example: AT#REC1ADD<CR>
Email1@domain.fr<CR><LF>
Email2@domain.com<CR><LF>
<CTRL+Z>

Default: None

#SUBJ1 / SUBJ2 / SUBJ3 – Set Email Pre-Defined Subject Text

Definition: These parameters contain the pre-defined subject text that will be used by the TCP/IP stack to compose the email header.

Syntax: **Set Value: AT#SUBJi="<value>"** (replace i with 1, 2, or 3)
View Value: AT#SUBJi? (replace i with 1, 2, or 3) or **AT#VMAILi, AT#VALL**

Values: Alphanumeric ASCII text string up to 120 characters.

Default: None

#BODY1 / BODY2 / BODY3 – Set Pre-Defined Email Combinations

Definition: These parameters store pre-defined message bodies. They allow the host application to send pre-defined email combinations.

Syntax: **Set Value: AT#BODYi="<value>"** (replace i with 1, 2, or 3)
View Value: AT#BODYi? (replace i with 1, 2, or 3) or **AT#VMAILi, AT#VALL**

Values: The body content has to be entered after the AT#BODY1<CR> command. It has to be an alphanumeric ASCII text string up to 120 characters followed this character: 1A (in ASCII code) and generated on a keyboard by CTRL+Z.

Example: AT#BODY1<CR>
Text string
<CTRL+Z>

Default: None

#POP3HEADERMODE – Send/Do Not Send POP3 Header

Definition: When receiving an email message, the TCP/IP stack can be configured to send or not to send the POP3 header over the serial port. The POP3 header contains the From, Cc and Subject fields.

Syntax: **Set Value: AT#POP3HEADERMODE=<value>**
View Value: AT#POP3HEADERMODE? or **AT#VPOP3, AT#VALL**

Values: **0** The email header will not be sent over the serial port while retrieving.
1 The email header will be sent over the serial port while retrieving.

Default: **1**

#POP3PORT – Set POP3 Server Port

Definition: To reach the POP3 server, the TCP/IP stack software must know the port of the POP3 server used for retrieving email.
Note: This parameter should be changed only by your network administrator since it depends on network infrastructure configuration including firewalls, proxy or specific TCP port translation settings.

Syntax: **Set Value: AT#POP3PORT=<value>**
View Value: AT#POP3PPORT? or **AT#VPOP3, AT#VALL**

Values: 5 digits (each digit between 0 and 9 inclusive). Note that numbers above 65,535 are illegal as the port identification fields are 16 bits long in the TCP header.

Default: **110**

#POP3PW – Set POP3 Password

Definition: Password for POP3 account. To retrieve email messages sent to a specified email address, the TCP/IP stack software must know the POP3 password that has been set for that email account.

Syntax: **Set Value: AT#POP3PW="<value>"**
View Value: AT#POP3PW? or AT#VPOP3, AT#VALL

Values: Alphanumeric ASCII text string up to 64 characters.

Default: None

#POP3SERV – Set POP3 Address

Definition: To retrieve email messages, the TCP/IP stack software must know the address of the POP3 server that is to be used. The POP3 server must be the one where the specified email account is hosted (which is not necessarily maintained by the local ISP).

Syntax: **Set Value: AT#POP3SERV="<value>"**
View Value: AT#POP3SERV? or **AT#VPOP3, AT#VALL**

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or
Alphanumeric ASCII text string up to 120 characters if DNS is available.

Default: None

#POP3UN – Set POP3 User Name

Definition: User name for POP3 account. To retrieve email messages sent to a specified email address, the TCP/IP stack software must know the POP3 user name that has been set for that email account.

Syntax: **Set Value: AT#POP3UN="<value>"**
View Value: AT#POP3UN? or **AT#VPOP3, AT#VALL**

Values: Alphanumeric ASCII text string up to 64 characters.

Default: None

#SMTPPORT – Set SMTP Server Port

Definition: To reach the SMTP server, the TCP/IP stack software must know the port of the SMTP server used for sending email.

Note: This parameter should be changed only by your network administrator since it depends on network infrastructure configuration including firewalls, proxy or specific TCP port translation settings.

Syntax: **Set Value: AT#SMTPPORT=<value>**
View Value: AT#SMTPPORT? or **AT#VSMTP, AT#VALL**

Values: From 1 to 5 digits (each digit between 0 and 9 inclusive). Note that numbers above 65,535 are illegal as the port identification fields are 16 bits long in the TCP header.

Default: **25**

#SMTPPW – Set SMTP Password

Definition: To send email messages, some SMTP servers are using an authentication process. In these cases, the TCP/IP stack software will provide the SMTP password (associated to the SMTP user name) for the email sending process.

If this parameter is an empty string, the authentication mode is inactive.

If both this parameter and the SMTPUN parameter are not empty, the authentication mode is active.

Syntax: **Set Value: AT#SMTPPW="<value>"**
View Value: AT#SMTPPW? or **AT#VSMTP, AT#VALL**

Values: Alphanumeric ASCII text string up to 64 characters.

Default: None

#SMTPSERV – Set SMTP Server Address

Definition: To send email messages the TCP/IP stack software must know the address of the SMTP server that is to be used. In most cases, the local ISP maintains the SMTP server.

Syntax: **Set Value: AT#SMTPSERV="<value>"**
View Value: AT#SMTPSERV? or **AT#VSMTP, AT#VALL**

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or
Alphanumeric ASCII text string up to 120 characters if DNS is available.

Default: None

#SMTPUN – Set SMTP User Name

Definition: To send email messages, some SMTP servers use an authentication process. In these cases, the TCP/IP stack software will provide the SMTP user name (associated with a SMTP password) for the email sending process.
If this parameter is an empty string, the authentication mode is inactive. If both this parameter and the SMTPPW parameter are not empty, the authentication mode is active.

Syntax: **Set Value: AT#SMTPUN="<value>"**
View Value: AT#SMTPUN? or **AT#VSMTP, AT#VALL**

Values: Alphanumeric ASCII text string up to 64 characters.

Default: None

#SMTPAUTH – Authentication ON or OFF

Definition: Turns authentication ON or OFF.

Syntax: **Set Value: AT#SMTPAUTH=<value>**
View Value: AT#SMTPAUTH? or **AT#VSMTP, AT#VALL**

Values: 0 OFF
1 ON

Default: **1**

#DNSSERV1 – Set DNS Server IP Address

Definition: In order to translate the server names from literal format into IP addresses, the TCP/IP stack software implements the Domain Name System (DNS) protocol. The DNS Server IP address must be specified for use by the TCP/IP stack software.

Syntax: **Set Value: AT#DNSSERV1="<value>"**
View Value: AT#DNSSERV1? or **AT#VDNS, AT#VALL**

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx).

Default: **0.0.0.0**

#DNSSERV2 – Set Secondary DNS Server

Definition: In order to translate the server names from literal format into IP addresses, the TCP/IP stack software implements the Domain Name System (DNS) protocol. The DNS Server IP address has to be specified for use by the TCP/IP stack software. This secondary DNS server is used in the case where the primary DNS server does not respond to a request.

Syntax: **Set Value: AT#DNSSERV2="<value>"**
View Value: AT#DNSSERV2? or **AT#VDNS, AT#VALL**

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx).

Default: **0.0.0.0**

Note: The UIP resolver (DNS Client) has a list that holds up to 4 DNS servers, the first 2 in the list are populated by the #DNSSERV1 and #DNSSERV2 commands, the 3rd and 4th are populated by DNS information provided by the ISP or cellular carrier. The list is checked in order (1-4). This means that user configured nameservers will be checked before dynamically assigned nameservers.

#GETMAIL – Retrieve Host Mail

Definition: This command allows the attached host to direct the TCP/IP stack to retrieve the first mail present in the POP3 server list. Once an IP link is established, the attached host can retrieve an email message at any time (except when the TCP/IP stack software is already in a process using TCP resources). This command is similar to a “check email box” feature issued by a standard messaging client on a PC.

Syntax: **AT#GETMAIL**

Command	Possible Responses
AT#GETMAIL Note: Retrieve mail	OK_Info_Mail <mail content> **Notes:** This message is issued when one email message is located in the specified POP3 mailbox. Depending on the POP3HEADERMODE parameter, the TCP/IP stack sends the email header over the serial port to the attached host. The (CR)(LF)(.)(CR)(LF) sequence finally indicates the end of the email body.
AT#GETMAIL	OK_Info_NoMail **Note:** There is no email to retrieve in the POP3 mailbox.
AT#GETMAIL	#CME ERROR: 38027 **Notes:** The address of the POP3 server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong POP3 server address has been entered.
AT#GETMAIL	#CME ERROR: <value> **Notes:** An error has occurred during the communication with the remote POP3 server. It may also happen during the data transfer after the MAIL message. In this case, it is preceded by a (CR)(LF)(.)(CR)(LF) sequence. This error can be due to one of the following reasons: - The DNS servers are not able to resolve the POP3 server address - The POP3 server is temporarily out of service - The authentication (POP3UN, POP3PW) is not valid See section ‘Response messages and error codes’.

To view parameters, use #VPOP3

#SENDMAIL1 / #SENDMAIL2 / #SENDMAIL3 – Send Mail

Important Note: MT5656SMI-IP supports only SENDMAIL1 options.

Definition: This command sends one of the 3 pre-defined email combinations. Once an IP link is established, the attached host can direct the TCP/IP stack to send an email message at any time (except when the TCP/IP stack software is already in a process using TCP resources).

The header of this email is built using the REC1/2/3, CCREC1/2/3 and SUBJ1/2/3 parameters while the body is filled in the BODY1/2/3 parameter.

This command is similar to a "send email" operation issued by a standard messaging client on a PC.

Syntax: **AT#SENDMAILi**

Note: In #SENDMAILi, the *i* is used for #SENDMAIL1, #SENDMAIL2, or #SENDMAIL3.

Command	Possible Responses
AT#SENDMAIL1 Note: Send predefined mail #1	**OK** Note: Mail 1 has been successfully sent
AT#SENDMAIL2 Note: Send predefined mail #2	**OK** Note: Mail 2 has been successfully sent
AT#SENDMAIL3 Note: Send predefined mail #3	**OK** Note: Mail 3 has been successfully sent
AT#SENDMAIL2	**#CME ERROR: 38027** Notes: The address of the SMTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary or secondary DNS servers or a wrong SMTP server address has been entered.
AT#SENDMAIL1	**#CME ERROR: <value>** Notes: An error has occurred during the communication with the remote SMTP server. It may also happen during the data transfer (after the OK message). This error can be due to one of the following reasons: - The DNS servers are not able to resolve the SMTP server address - The SMTP server is temporarily out of service - The authentication (SMTPUN, SMTPPW) is not valid - An email address specified in REC1 or CCREC1 is not valid See section 'Response messages and error codes'.

To view parameters, use #VMAIL and #VSMTP. They are also listed in #VMAILi (where i = 1, 2, or 3)

#PUTMAIL – Send Host Mail

Definition: This command allows the attached host to send an email message containing body text passed to the TCP/IP stack over the serial port. Once an IP link is established, the attached host can send an email message at any time (except when the TCP/IP stack software is already in a process using TCP resources).

The header of this email is built using the REC1, CCREC1 and SUBJ1 parameters.

This command is similar to a "send email" operation issued by a standard messaging client on a PC.

Syntax: **AT#PUTMAIL**

Command	Possible Responses
AT#PUTMAIL Note: You have to configure only receiver address1, copy address1, and subject1 before or during the session, but content (body) of the email is typed when the TCP/IP session is established. Content is not echoed.	OK_Info_WaitingForData Notes: An SMTP session has been successfully opened with the remote SMTP server. Once the TCP/IP stack has displayed this message, all the data received on the serial port is sent within the email body. The (CR)(LF)(.)(CR)(LF) sequence sent by the attached host indicates the TCP/IP stack the end of the email body.
<content> Note: Content is not written when typing	
<CR><LF> . <CR><LF> Note: Termination sequence <CR> = <Enter>, <LF> = <Ctrl Enter>	OK Note: The mail has been successfully sent.
AT#PUTMAIL	#CME ERROR: 38027 Notes: The address of the SMTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary or secondary DNS servers or a wrong SMTP server address has been entered.
AT#PUTMAIL	#CME ERROR: <value> Notes: An error has occurred during the communication with the remote SMTP server. It may also happen during the data transfer (after the OK message). This error can be due to one of the following reasons: - DNS servers are not able to resolve the SMTP server address. - SMTP server is temporarily out of service. - Authentication (SMTPUN, SMTPPW) is not valid. - An email address specified in REC1 or CCREC1 is not valid. - *n* = inactivity period of 50 seconds on the serial port. See section 'Response messages and error codes'

To view parameters, use #VMAIL and #VSMTP. They are also listed in #VMAILi (where i = 1, 2, or 3)

Display Commands

#VMAIL1 / #VMAIL2 / #VMAIL3 – Display Email Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the email combinations configuration.

Syntax: **AT#VMAILi** (where **i = 1**, **2**, or **3**)

Command	Possible Responses
AT#VMAIL1 Note: View predefined (nb 1) mail header elements	#Body 1= #REC1= #CCREC1= #SUBJ1= OK

#VPOP3 – Display POP3 Parameters

Definition: Directs the TCP/IP stack to display all the AT# parameters related to the email retriever configuration.

Syntax: **AT#VPOP3**

Command	Possible Responses
AT#VPOP3 Note: View POP3 parameters	#POP3HEADERMODE: 1 #POP3PORT: 110 #POP3PW: “mypop3passwd” #POP3SERV: “pop3server” #POP3UN: “mypop3un” OK

#VSMTP – Display SMTP Parameters

Definition: Directs the TCP/IP stack to display all the AT# parameters related to the email sender configuration.

Syntax: **AT#VSMTP**

Command	Possible Responses
AT#VSMTP Note: View SMTP parameters	#DOMAIN: “a2myoper.com” #SENDERADDR: “toto@myoper.com” #SENDERNAME: “toto” #SMTPPORT: 25 #SMTPPW: “mysmtppw” #SMTPSERV: “smtp.a2myoper.com” #SMTPUN: “mysmtpun” #SMTPAUTH: 1 OK

#VDNS – Display DNS Parameters

Definition: Directs the TCP/IP stack to display all the AT# parameters related to the DNS servers configuration.

Syntax: **AT#VDNS**

Command	Possible Responses
AT#VDNS	#DNSSERV1: “0.0.0.0” #DNSSERV2: “0.0.0.0” OK

Chapter 3 – FTP Commands

Set Commands

#FTPPORT – Enter FTP Server Port

Definition: To reach the FTP server, the TCP/IP stack software must know the control port of the FTP server used for file transfer.

Note: This parameter should be changed only upon request of your network administrator since it applies to network infrastructure including firewalls, proxy or specific TCP port translation.

Syntax: **Set Value: AT#FTPPORT=<value>**

View Value: AT#FTPPORT? or **AT#VFTP, AT#VALL**

Values: From 1 to 5 digits (each digit between 0 and 9 inclusive).

Numbers above 65,535 are illegal as port identification fields are 16 bits long in the TCP header.

Default: **21**

#FTPMODE – Enter FTP Mode

Definition: Define the FTP behavior for file transfer.

Syntax: **Set Value: AT#FTPMODE=<value>**

View Value: AT#FTPMODE? or **AT#VFTP, AT#VALL**

Values: **0** FTP Active Mode

1 FTP Passive Mode

Default: **0**

#FTPTYPE – Specify Data Type

Definition: Before transferring files from a specified FTP server, the TCP/IP stack software must specify the type of data to be transferred within the FTP session.

Syntax: **Set Value: AT#FTPTYPE="<value>"**

View Value: **AT#FTPTYPE?** or **AT#VFTP, AT#VALL**

Values: **A** FTP ASCII sessions

I FTP Binary sessions.

Note: When this value is set to A, all the data sent by the TCP/IP stack to the FTP server is made of 7 bits characters (NVT-ASCII: the MSB is set to 0). As a consequence binary data containing 8 bits characters will be corrupted during the transfer if the FTPTYPE is set to A.

Default: **I**

#FTPSERV – Specify FTP Server Address

Definition: FTP server address. To connect to an FTP server to download files, the TCP/IP stack software must know the address of the FTP server that is to be used.

Syntax: **Set Value: AT#FTPSERV="<value>"**

View Value: AT#FTPSERV? or **AT#VFTP, AT#VALL**

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or

Alphanumeric ASCII text string up to 120 characters if DNS is available.

Default: None

#FTPUN – Set FTP User Name

Definition: Before transferring files from a specified FTP server, the TCP/IP stack software must open an FTP session using a valid FTP user name.

Syntax: **Set Value: AT#FTPUN="<value>"**
View Value: **AT#FTPUN?** or **AT#VFTP, AT#VALL**

Values: Alphanumeric ASCII text string up to 64 characters.

Default: None

#FTPPW – Set FTP Password

Definition: Before transferring files from a specified FTP server, the TCP/IP stack software must open an FTP session using a valid FTP password.

Syntax: **Set Value: AT#FTPPW="<value>"**
View Value: AT#FTPPW? or **AT#VFTP, AT#VALL**

Values: Alphanumeric ASCII text string up to 64 characters.

Default: None

#FTPGETFILENAME – Set FTP Download File Name

Definition: In order to download a file from the FTP server, the TCP/IP stack software must know the name of the relevant file.

Syntax: **Set Value: AT#FTPGETFILENAME="<value>"**
View Value: AT#FTPGETFILENAME? or **AT#VFTP, AT#VALL**

Values: Alphanumeric ASCII text string up to 120 characters.

Default: None

#FTPGETPATH – Set FTP Path

Definition: In order for the TCP/IP stack software to get a file from the FTP server, the TCP/IP stack software must know the path of the relevant file. For example, it could be: /list

Syntax: **Set Value: AT#FTPGETPATH="<value>"**
View Value: AT#FTPGETPATH? or **AT#VFTP, AT#VALL**

Values: Alpha-numeric ASCII text string up to 120 characters.
Note: Depending on the FTP server, the value can be used for getting a file from the root directory of the FTP server.

Default: None

#FTPPUTFILENAME – Set FTP Upload File Name

Definition: In order for the TCP/IP stack software to upload a file to the FTP server, the TCP/IP stack software must know the name of the relevant file.

Syntax: **Set Value: AT#FTPPUTFILENAME="<value>"**
View Value: AT#FTPPUTFILENAME? or **AT#VFTP, AT#VALL**

Values: Alpha-numeric ASCII text string up to 120 characters.

Default: None

#FTPPUTPATH – Set Path to FTP Server

Definition: In order for the TCP/IP stack software to upload a file to the FTP server, the TCP/IP stack software must know the path of the relevant file. For example, it could be: /list

Syntax: **Set Value: AT#FTPPUTPATH="<value>"**

View Value: AT#FTPPUTPATH? or **AT#VFTP, AT#VALL**

Values: Alpha-numeric ASCII text string up to 120 characters.

Note: Depending on the FTP server, the value can be used for getting a file from the root directory of the FTP server.

Default: None

#FTPGET – Get Data from Server

Definition: This command, sent by the attached host, directs the TCP/IP stack to connect to the specified FTP server and to retrieve the specified file from this server. Once the operation is completed, the TCP/IP stack closes the FTP connection.

Once an IP link is established, the attached host can retrieve a file from an FTP server at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This command is similar to a GET operation (with an automatic connect/disconnect) issued by a standard FTP client on a PC. The TCP/IP stack handles the global FTP process by itself.

Syntax: **AT#FTPGET**

Command	Possible Responses
AT#FTPGET **Note:** Start data reception	Ok_Info_DataBegin The switch from command to data mode. DATA Note: Data transmitted from the FTP server to the TCP/IP stack is sent over the serial port. <ETX> Note: Once the file transfer has finished, the TCP/IP stack sends an ETX character over the serial port to notify the attached host of the end of file transfer: switches from data to command mode. OK Note: The FTP process was successfully completed.
AT#FTPGET	#CME ERROR: 38027 Note: The address of the FTP server has not been resolved by the secondary DNS server. The TCP/IP stack is not able to reach the primary or secondary DNS servers or a wrong FTP server address has been filled in.
AT#FTPGET	#CME ERROR: <value> Note: The connection to the FTP server failed. See section 'Response messages and error codes.' If this error occurs once the data transfer started, it is preceded by an ETX character.

To view the parameters, use the #VFTP command.

#FTPPUT – Put Data to Server

Definition: This command sent by the attached host directs the TCP/IP stack to connect to the specified FTP server and to upload the data received on the serial port to the specified file on this server. Once the operation is completed, the TCP/IP stack closes the FTP connection.

Once an IP link is established, the attached host can send a file to a FTP server at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This command is similar to a PUT operation (with an automatic connect/disconnect) issued by a standard FTP client on a PC. The TCP/IP stack handles the global FTP put process by itself.

Note: The TCP/IP stack will interpret only an <ETX> character as the end of the file to be transferred if it's not preceded by a <DLE> character. As a consequence, the attached host must send <ETX> characters preceded by <DLE> characters, and it must also code <DLE> characters as <DLE><DLE>.

Syntax: **AT#FTPPUT**

Command	Possible Responses
AT#FTPPUT Note: Start sending data	OK_WaitingForData Note: TCP/IP stack is ready to send data from the serial port to the remote FTP server. TCP/IP stack then immediately transfers all the data sent by the attached host to the remote FTP server. To notify the TCP/IP stack that all data has been sent, the attached host must send the <ETX> character. <ETX> Notification from host for end of data: switches from data to command mode. OK Note: The FTP process was successfully completed.
AT#FTPPUT	#CME ERROR: 38027 Note: The address of the FTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary or secondary DNS servers or a wrong FTP server address has been filled in.
AT#FTPPUT	#CME ERROR: <value> Note: The connection to the FTP server failed. See section 'Response messages and error codes'. If an error occurs once the data transfer started, it is preceded by ***ETX***.

To view the parameters, use the #VFTP command.

#FTPAPPEND – Append Data to Server

Definition: This command sent by the attached host directs the TCP/IP stack to connect to the specified FTP server and to append the data received on the serial port to the specified file on this server. Once the operation is completed, the TCP/IP stack closes the FTP connection.

Once an IP link is established, the attached host can send a file to a FTP server at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This command is similar to a PUT operation (with an automatic connect/disconnect) issued by a standard FTP client on a PC. The TCP/IP stack handles the global FTP put process by itself.

Note: The TCP/IP stack will interpret only an <ETX> character as the end of the file to be transferred if it's not preceded by a <DLE> character. As a consequence, the attached host must send <ETX> characters preceded by <DLE> characters, and it must also code <DLE> characters as <DLE><DLE>.

Syntax: **AT#FTPAPPEND**

Command	Possible Responses
AT#FTPAPPEND Note: Start sending data	OK_WaitingForData Note: TCP/IP stack is ready to append data from the serial port to the remote FTP server. TCP/IP stack then immediately appends all the data sent by the attached host to the remote FTP server. To notify the TCP/IP stack that all data has been appended, the attached host must send the <ETX> character. <ETX> Notification from host for end of data: switches from data to command mode. OK Note: The FTP process was successfully completed.
AT#FTPAPPEND	#CME ERROR: 38027 Note: The address of the FTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary or secondary DNS servers or a wrong FTP server address has been filled in.
AT#FTPAPPEND	#CME ERROR: <value> Note: The connection to the FTP server failed. See section 'Response messages and error codes'. If an error occurs once the data transfer started, it is preceded by ***ETX***.

To view the parameters, use the #VFTP command.

Display Command

#VFTP – Display FTP Parameters

Definition: This command directs the TCP/IP stack to display all AT# parameters for the FTP client configuration.

Syntax: **AT#VFTP**

Command	Possible Responses
AT#VFTP Note: View FTP parameters	#FTPGETFILENAME: "" #FTPGETPATH: "." #FTPMODE: 0 #FTPPORT: 21 #FTPPUTFILENAME: "Testseb3.txt" #FTPPUTPATH: "." #FTPPW: "mypass" #FTPSERV: "mytestwebsite.com" #FTPTYPE: = I #FTPUN: "myname" OK

Chapter 4 – TCP Commands

Set Commands

The **id** parameter is used for setting up profiles; Profile 1 and Profile 2. To view parameters including profile settings, use **AT#<cmd>=id** or **AT#VTCP=id** or **AT#VALL**.

#DLEMODE – Set DLE Mode

Definition: When using socket TCP, the attached host has the choice whether or not to code the ETX character.

Syntax: **Set Value: AT#DLEMODE=id,<DLE Mode value>**
View Value: AT#DLEMODE=id or **AT#VTCP=id, AT#VALL**

id: Enter a Profile Setting: 1 or 2 {1,2}

Values: Set the value that the Profile will use:

0 When DLEMODE is set to 0, no specific process is needed on [ETX] characters.
It means that it is not possible for a host to request an end of connection or to receive a clear indication of the end of a connection from the TCP/IP stack.

1 When DLEMODE is set to 1, the [ETX] character means a request or an indication of end of connection.
As a consequence, [ETX] characters that belong to the payload data must be sent by the host on the serial port preceded by a DLE character. Similarly, ETX characters received by the TCP/IP stack from the Internet are sent to the host through the serial port preceded by a DLE character.

Default: **1**

#TCPPORT – Set TCP Port for Remote Peer

Definition: To exchange data over TCP, the TCP/IP stack software must know the port of the remote peer used for the TCP session.

Syntax: **Set Value: AT#TCPPORT=id,<TCP Port number for remote peer value>**
View Value: AT#TCPPORT=id or **AT#VTCP=id, AT#VALL**

id: Enter a Profile Setting: 1 or 2 {1,2}

Values: Set the value that the Profile will use:
From 1 to 5 digits (each digit between 0 and 9 inclusive). Note that numbers above 65,535 are illegal as the port identification fields are 16 bits long in the TCP header.

Default: **0**

#TCPSERV – Set Address of Remote TCP Server

Definition: To exchange data over TCP, the TCP/IP stack software must know the address of the remote TCP server (or host) that is to be used.

Syntax: Set Value: **AT#TCPSERV=id,"<Address of remote TCP server value>"**
View Value: **AT#TCPSERV=id** or **AT#VTCP=id,** or **AT#VALL**

id: Enter a Profile Setting: 1 or 2 {1,2}

Values: Set the value that the Profile will use:
32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or
Alphanumeric ASCII text string up to 120 characters if DNS is integrated.

Default: None

#TCPTXDELAY – Time Delay

Definition: This command determines the time delay introduced before sending a TCP frame that has not been entirely filled with user data. The time is entered in milliseconds, and it should be noted that a value of 0 initiates the sending of a TCP frame as soon as possible after the reception of a single character value from the host.

Syntax: **Set Value: AT#TCPTXDELAY=id,<Time Delay value>**
View Value: AT#TCPTXDELAY=id or **AT#VTCP=id** or **AT#VALL**

id: Enter a Profile Setting: 1 or 2 {1,2}

Values: Set the value that the Profile will use:
Integer multiple of 20 between 0 and 32760 inclusive.

Default: **100**

#LTCPSTART – Open Listening Mode

Definition: This command, sent by the attached host, directs the TCP/IP stack to open a listening TCP connection on the specified TCP port.

Once an IP link is established, the attached host can open a listening TCP socket at any time (except when the TCP/IP stack software is already in a process using TCP resources).

The TCP connection will be active upon reception of a TCP connection request sent by a remote allowed TCP peer (TCPSERV) on the appropriate TCP port (TCPPORT).

Once opened, this TCP connection may be closed by the remote TCP peer or by the attached host by sending an ETX character on the serial port (depending on the DLEMODE parameter).

Notes:
The LTCP command can be aborted before an incoming TCP request has been received by issuing an <ETX> character on the serial port.

If the DLEMODE parameter is set to 1, the TCP/IP stack will interpret an <ETX> character only as a close request if a <DLE> character does not precede it. As a consequence, the attached host must send <ETX> characters preceded by <DLE> characters, and it must also code <DLE> characters in <DLE><DLE>. Similarly, each <ETX> character present in the payload data of the TCP frame will be coded by the TCP/IP stack on the serial port as <DLE><ETX>. Each <DLE> character will be coded as <DLE><DLE>. The attached host must then decode the TCP socket flow to remove these escape characters.

If the DLEMODE parameter is set to 0, the TCP/IP stack will never close the TCP connection (unless an error occurs).

Syntax: **AT#LTCPSTART=id**

Command	Possible Responses
AT#LTCPSTART=1 Note: Active listening mode	OK_Info_WaitingForData Note: This message signals that an allowed remote TCP peer has opened the TCP socket. The TCP connection is now opened. All the data from the attached host / remote TCP server is then immediately transferred by the TCP/IP stack to the remote TCP server / attached host. Depending on the DLEMODE value, the attached host may close this TCP connection by sending an ETX character. If the remote TCP server closes the connection, the TCP/IP stack issues an ETX character on the serial port.
AT#LTCPSTART=2	OK_Info_SocketClosed OK Note: The TCP socket is closed.
AT#LTCPSTART=1	#CME ERROR: <value> Note: An error has occurred during the TCP connection. This connection is being closed. If this error occurs once the TCP connection opened, it is preceded by an ETX character. See 'Response Messages & Error Codes'.

id: Enter a Profile Setting: 1 or 2 {1,2}

To view the parameters, use #VTCP.

#LTCPSTOP – Close Listening Mode

Definition: This command directs the TCP/IP stack to close a TCP listening mode (previously launched by the AT#LTCPSTART command.

Syntax: **AT#LTCPSTOP=id**

Command	Possible Responses
AT#LTCPSTOP=2 Note: Stop TCP listening	OK **Note:** The local listening port is closed in the TCP/IP stack.

id: Enter a Profile Setting: 1 or 2 {1,2}

To view the parameters, use #VTCP.

#OTCP – Open TCP Connection

Definition: This command sent by the attached host directs the TCP/IP stack to open a TCP connection to the specified TCP server. Once an IP link is established, the attached host can open a TCP connection at any time (except when the TCP/IP stack software is already in a process using TCP resources).

This TCP connection may be closed by the remote TCP server or by the attached host via sending an ETX character on the serial port (**depending on the DLEMODE parameter**).

DLEMODE Value Notes:

Depending on the DLEMODE value, the attached host may close this TCP connection by sending an ETX character.

- If the DLEMODE parameter is set to 1, the TCP/IP stack will only interpret an <ETX> character as a close request if it's not preceded by a <DLE> character. As a consequence, the attached host must send <ETX> characters preceded by <DLE> characters, and it must also code <DLE> characters in <DLE><DLE>. Similarly, each <ETX> character present in the payload data of the TCP frame will be coded by the TCP/IP stack on the serial port as <DLE><ETX>. Each <DLE> character will be coded as <DLE><DLE>. The attached host must then decode the TCP socket flow to remove these escape characters.
- If DLEMODE is set to 0, the host cannot close the TCP connection (unless an error occurs).

If the remote TCP server closes the connection, the TCP/IP stack sends an ETX character on the serial port.

Syntax: **AT#OTCP=id**

Command	Possible Responses
AT#OTCP=1 Request TCP socket open	OK_Info_WaitingForData Note: This message signals that the TCP socket has been opened. All the data from the attached host / remote TCP server is then immediately transferred by the TCP/IP stack to the remote TCP server / attached host. Depending on the DLEMODE value, the attached host may close this TCP connection by sending an ETX character. If the remote TCP server closes the connection, the TCP/IP stack issues an ETX character on the serial port.
	OK_Info_SocketClosed OK Note: The TCP socket is closed.
AT#OTCP=2	#CME ERROR: 38027 Note: Address of FTP server has not been resolved by the secondary DNS server. TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong FTP server address has been filled in.
AT#OTCP=2	#CME ERROR: <value> Note: An error has occurred during the TCP connection. This connection is being closed. If this error occurs once the TCP connection opened, it is preceded by an ETX character. See section 'Response messages and error codes'.

id: Enter a Profile Setting: 1 or 2 {1,2}

To view the parameters, use #VTCP.

Display Command

#VTCP – Display TCP Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the TCP socket configuration.

Syntax: **AT#VTCP=id** or **AT#VTCP**

Note: If you use **AT#VTCP=id**, the parameters for profile 1 ***or*** profile 2 will display. If you use **AT#VTCP**, the parameters for both profile 1 and profile 2 will display.

Command	Possible Responses
AT#VTCP Note: View TCP parameters of the TCP socket 2	#DLEMODE: 1 , 1 #TCPSERV: 1 , ”... ” #TCPPORT: 1 , 0 #TCPTXDELAY: 1 , 100 #DLEMODE: 2 , 1 #TCPSERV: 2 , ”...” #TCPPORT: 2 , 0 TCPTXDELAY: 2 , 100 OK

Chapter 5 – UDP Commands

Set Commands

#UDPPORT – UDP Port Number

Definition: If the UDP session is initiated in listen mode, use the local UDP port number.
If the UDP session is initiated in active mode, use the remote UDP port number.

Syntax: **Set Value: AT#UDPPORT=<Port Number value>**
View Values: AT#UDPPORT or **AT#VUDP** or **AT#VALL**

Values: Set the port number that the Profile will use:
From 1 to 5 numeric digits (0 to 9).

Note:
Numbers above 65,535 are illegal since port identification fields are 16-bits long in the IDP header.

Default: **0**

Example: AT#UDPPORT=xxxx

#UDPSERV – UDP Server Address

Definition: If the UDP session is initiated in listen mode, use the IP address filter. This means that the remote must have a defined UDPSERV IP address.
If the UDP session is initiated in active mode, use the remote IP address.
Note: No IP filter is applied if the parameter value is **255.255.255.255**

Syntax: **Set Value: AT#UDPSERV="<UDP Server Addressvalue>"**
View Value: AT#UDPSERV or **AT#VUDP** or **AT#VALL**

Values: Set the UDP Server Address value that the Profile will use:
32-bit number is dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or
Alphanumeric ASCII text string up to 120 characters if DNS is integrated.

Default: None

Example: AT#UDPSERV="xxx.xxx.xxx.xxx"

#UDPTXDELAY – UDP Data Delay

Definition: This parameter determines the delay before sending an UDP datagram that has not been entirely filled with user data. The delay is expressed in milliseconds.
The 0 value initiates the sending an UDP datagram as soon as possible after the reception of a single character value from the host.

Syntax: **Set Value: AT#UDPTXDELAY=<Delay value>**
View Value: AT#UDPTXDELAY or **AT#VUDP** or **AT#VALL**

Values: Set the UDP Data Delay value that the Profile will use:
Integer, multiple of 20 and between 0 and 32760 inclusive.

Default: **100**

Example: AT#UDPTXDELAY=xxx

#LUDPSTART – Open UDP Session in Listen Mode

Definition: Once an IP link is established, and if no other TCP/IP stack resource is active, this command sent by the attached host directs the TCP/IP stack to initiate the UDP session in listen mode on the specified UDP local port UDPPORT. Depending on the UDPSERV parameter content, there are two possible combinations:

- If UDPSERV is 255.255.255.255, the IP address filter feature is turned off. The UDP session will be effective upon reception of the first datagram, supplying the remote port number and the remote IP address of the session. The host can then transmit to the (remote port or remote IP) address.
- If UDPSERV holds a valid IP address (or existing/found alphanumeric IP address), only the remote with the UDPSERV IP address will be authorized to send datagrams to this UDP socket (i.e., the IP address filter feature is activated). The UDP session will be effective upon reception of the first datagram, supplying the remote port number of the session. The host can then transmit to the (remote port, UDPSERV) address. Datagrams can then only be received from this (remote port, UDPSERV) address.

As long as the first datagram is not received, UDPSTOP can be issued to cancel the UDP session. Once the first datagram has been received, the only way to stop the UDP session for the host is to issue an <ETX> character.

Note: The TCP/IP stack will only interpret an <ETX> character as a close request if it is not preceded by a <DLE> character. As a consequence, an <ETX> character must be sent (even in payload data) as <DLE><ETX>, and <DLE> character as <DLE><DLE>. The attached host must then decode the UDP socket flow to remove these escape characters.

Syntax: **AT#LUDPSTART**

Command	Possible Responses
AT#LUDPSTART Note: Active listen mode	OK_Info_WaitingForData Notes: This message signals that the remote allowed UDP peer has sent its first datagram to the local UDP socket. The UDP connection is now effective. All data from the attached host / remote UDP peer is immediately transferred by the TCP/IP stack to the remote UDP peer / attached host. Attached host may close this UDP session by sending an <ETX> character. If an error occurs, the TCP/IP stack issues and <ETX> character on the serial port before sending the appropriate error message.
AT#LUDPSTART	OK_Info_WaitingForData ... ... <ETX> OK_Info_SocketClosed OK Note: The UDP socket is closed.
AT#LUDPSTART	#CME ERROR: <value> Note: An error has occurred during the UDP session creation. If this error occurs once the UDP session is effective, it is preceded by <ETX>. See 'Response Messages & Error Codes'.

Example: AT#LUDPSTART

To view parameters, use #VUDP.

#LUDPSTOP – Close a Listening Mode UDP Session

Definition: This command directs the TCP/IP stack to close a UDP listening mode session (previously launched by the AT#LUDPSTART command).

Syntax: **AT#LUDPSTOP**

Command	Possible Responses
AT#LUDPSTOP Note: Stop UDP listening	OK Note: The local listening port is closed in the TCP/IP stack.

Example: AT#LUDPSTOP

To view parameters, use #VUDP.

#OUDP – Open an Active UDP Session

Definition: Once an IP link is established, and if no other TCP/IP stack resource is active, this command sent by the attached host directs the TCP/IP stack to initiate an UDP session in active mode on the specified UDP remote port UDPPORT to the specified remote IP address UDPSERV.

The host can then transmit to the UDPPORT or UDPSERV address. Datagrams can only be received from this UDPPORT or UDPSERV address.

The host can stop the UDP session by issuing an <ETX> character.

Note: The TCP/IP stack will only interpret an <ETX> character as a close request if it is not preceded by a <DLE> character. As a consequence, an <ETX> character must be sent (even in payload data) as <DLE><ETX>, and <DLE> character as <DLE><DLE>. The attached host must then decode the UDP socket flow to remove these escape characters.

Syntax: **AT#OUDP**

Command	Possible Responses
AT#OUDP Note: Request opening of UDP socket	OK_Info_WaitingForData Notes: This message signals that the UDP socket has been opened. All data from the attached host / remote UDP peer is immediately transferred by the TCP/IP stack to the remote UDP peer / attached host. The attached host may close this UDP session by sending <ETX>. If an error occurs, the TCP/IP stack issues an <ETX> character on the serial port before sending the appropriate error message.
AT#OUDP	OK_Info_WaitingForData ... ... <ETX> OK_Info_SocketClosed OK Note: The UDP socket is closed.
AT#OUDP	#CME ERROR: 38027 Note: The address of the remote UDP peer has not been resolved by the secondary DNS server. The TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong remote UDP peer address has been filled in.
AT#OUDP	#CME ERROR: <value> Note: An error has occurred during the UDP session creation. If this error occurs once the UDP session is effective, it is preceded by <ETX>. See 'Response Messages & Error Codes'.

To view parameters, use #VUDP.

Display Command

#VUDP – Display UDP Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the UDP socket configuration.

Syntax: **AT#VUDP**

Command	Possible Responses
AT#VUDP Note: View UDP parameters	#UDPSERV: "..." #UDPPORT: 0 #UDPTXDELAY: 100 OK

Chapter 6 – PING Commands

Set Commands

#PINGDELAY – PING Delay Time

Definition: This command sets the waiting delay, in seconds, before an echo request is considered as a ***no reply***. It is also the delay between two echo requests (if PINGNUM > 1).

Syntax: **Set Value: AT#PINGDELAY=<Value>**

View Value: AT#PINGDELAY? or **AT#VPING, AT#VALL**

Values: From 1 to 255 inclusive.

Default: **1**

#PINGNUM – Number of PING Requests

Definition: This command sets the number of PING echo requests to issue to PINGREMOTE.

Syntax: **Set Value: AT#PINGNUM=<Value>**

View Value: AT#PINGNUM? or **AT#VPING, AT#VALL**

Values: From 1 to 255 inclusive.

Default: **4**

#PINGREMOTE – IP Address of PING Request

Definition: Sets the IP address or alphanumeric ASCII text string, up to 120 characters long if DNS is available.

Syntax: **Set Value: AT#PINGREMOTE="<value>"**

View Value: AT#PINGREMOTE? or **AT#VPING, AT#VALL**

Values: 32-bit number is dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or

Alphanumeric ASCII text string up to 120 characters if DNS is integrated.

Default: None

#PING – Start PING Request

Definition: Once an IP link is established, and if no other TCP/IP stack resource is active, this command sent by the attached host directs the TCP/IP stack to start PING requests.

Syntax: **AT#PING**

Command	Possible Responses
AT#PING	"multitech.com" is alive : time = 900 ms "multitech.com" is alive : time = 900 ms "multitech.com" is alive : time = 800 ms "multitech.com" is alive : time = 800 ms "multitech.com" is alive : time = 1000 ms NO answer from "wavecom.com" "multitech.com" is alive : time = 900 ms "multitech.com" is alive : time = 800 ms "multitech.com" is alive : time = 900 ms "multitech.com" is alive : time = 900 ms OK Note: TCP/IP stack sends PINGNUM = 10 requests
AT#PING	#CME ERROR: 38027 Note: The address of the remote has not been resolved by the DNS servers. The TCP/IP stack is not able to reach the primary and secondary DNS servers or a wrong remote address has been entered.
AT#PING	#CME ERROR: <value> Note: An error has occurred during the PING requests. See 'Response Messages & Error Codes'.

To view parameters, use #VPING.

Display Command

#VPING – Display PING Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the PING configuration.

Syntax: **AT#VPING**

Command	Possible Responses
AT#VPING Note: View PING parameters	#PINGDELAY: 1 #PINGNUM: 4 #PINGREMOTE: "..." OK

Chapter 7 – Miscellaneous Commands

Set Command

#DELFLASH – Erase Flash Memory Parameter Values

Definition: This command erases the contents of the flash memory and loads with defaults values.

Syntax: **AT#DELFLASH**

Command	Possible Responses
AT#DELFLASH Note: Delete flash memory contents	OK

Values: No parameter values.

Display Commands

#VVERSION – Display Software Version

Definition: This command directs the TCP/IP stack to display the software version.

Syntax: **AT#VVERSION**

Values: No TCP/IP parameters.

Command	Possible Responses
AT#VVERSION Note: Request TCP/IP stack version	#VERSION: “1.00” OK

#VALL – Display All Parameters

Note: The MT100SEM display of Possible Responses differs slightly from the MT5656SMI-IP display. An example of the **MT100SEM Possible Responses** screen is included on this page. See the next page for an example of the **MT5656SMI-IP Possible Responses** screen.

Definition: This command directs the TCP/IP stack to display all AT# parameters. The parameters are displayed by blocks of categories separated by a <CR><LF> sequence, all at the same time.

Syntax: **AT#VALL**

Possible Responses – *MT100SEM*

#FTPGETFILENAME: " "
#FTPGETPATH: " "
#FTPMODE: 0
#FTPPORT: 21
#FTPPUTFILENAME: " "
#FTPPUTPATH: " "
#FTPPW: "ftppassword"
#FTPSERV: " "
#FTPTYPE: 1
#FTPUN: “ftplogin”

#POP3HEADERMODE: 1
#POP3PORT: 110
#POP3PW: "password"
#POP3SERV: "pop3.domain.com"
#POP3UN: "module@domain.com"

#DOMAIN: "domain.com"
#SENDERADDR: "module@domain.com"
#SENDERNAME: "Test module"
#SMTPPORT: 25
#SMTPPW: " "
#SMTPSERV: "smtp.domain.com"
#SMTPUN: " "
#SMTPAUTH: 1

#BODY1: " "
#CCREC1: " "
#REC1: " "
#SUBJ1: "Email Subject"

#BODY2: " "
#CCREC2: " "
#REC2: " "
#SUBJ2: " "

#BODY3: " "
#CCREC3: " "
#REC3: " "
#SUBJ3: " "

#DLEMODE: 1 , 1
#TCPSERV: 1 , " "
#TCPPORT: 1 , 0
#TCPTXDELAY: 1 , 100
#DLEMODE: 2 , 1
#TCPSERV: 2 , " "
#TCPPORT: 2 , 0
#TCPTXDELAY: 2 , 100

#UDPSERV: 1 , " "
#UDPPORT: 1 , 0
#UDPTXDELAY: 1 , 100
#UDPSERV: 2 , " "
#UDPPORT: 2 , 0
#UDPTXDELAY: 2 , 100

#PINGDELAY: 1
#PINGNUM: 4
#PINGREMOTE: " "

#DHCP: 0
#IPADDR: "192.168.2.3"
#IPGATEWAY: "192.168.2.1"
#IPNETMASK: "255.255.255.0"
#EMACSPD: 0
#DNSSERV1: "0.0.0.0"
#DNSSERV2: "0.0.0.0"

#TELNET: 0
#TELNETPORT: 23
#TELNETUSER: "admin"
#TELNETPASSWORD: " "

+IPR: 115200
+ICF: 2,4

+IFC: 2,2

#AUTODISC: 1
#AUTODISCPORT:1020
#AUTODISCTIMER: 10
#AUTODISCHOST: "MT100SEM"
#AUTODISCUSER: "admin"
#AUTODISCPASSWORD: " "

V: 1
&S: 0
&C: 0

OK

Syntax: **AT#VALL**

Possible Responses – Applies to All Except MT100SEM	
#ANSWERMODE: 0	#TCPTXDELAY: 1,100
#CALLBACKTIMER: 2	#UDPPORT: 0
#CALLSCREENNUM: "0"	#UDPSERV: ""
#DIALN1: ""	#UDPTXDELAY: 100
#DIALN2: ""	#PINGDELAY: 1
#DIALSELECT: 1	#PINGNUM: 4
#GPRSMODE: 1	#PINGREMOTE: ""
#PHYTIMEOUT: 15	#AUTHENT: NONE
#REDIALCOUNT: 0	+IPR: 115200
#REDIALDELAY: 5	+ICF: 2,4
#RINGCOUNT: 0	+IFC: 2,2
#AUTOCONNECT: 0	V: 1
#ISPUN: ""	E: 1
#ISPPW: ""	&S: 1
#PPPMODE: 3	&C: 1
#PPPMYIP: 0.0.0.0	&D: 0
#PPPPEERIP: 0.0.0.0	#KEEPALIVEMODE: 0
#PPPSERVPW: ""	#KEEPALIVEPORT: 0
#PPPSERVUN: ""	#KEEPALIVEDELAY: 100
#APNPW: ""	#KEEPALIVESERV: ""
#APNSERV: ""	#LEGACYMODE: 0
#APNUN: ""	#AUTORESET: 0
#GPRSCID: 1	#ATCMD: 1,""
#DNSSERV1: "0.0.0.0"	#ATCMD: 2,""
#DNSSERV2: "0.0.0.0"	#ATCMD: 3,""
#FTPGETFILENAME: ""	#ATCMD: 4,""
#FTPGETPATH: ""	#PERSISTENTSOCKET: 0
#FTPPORT: 21	#OUTPORT: 37500
#FTPPUTFILENAME: ""	#GPIO: 1,0,0,0,0,0
#FTPPUTPATH: ""	#GPIO: 2,0,0,0,0,0
#FTPPW: ""	#GPIO: 3,0,0,0,0,0
#FTPSERV: ""	#GPIO: 4,0,0,0,0,0
#FTPTYPE: I	#GPIO: 5,3,0,0,0,0
#FTPUN: ""	OK
#FTPMODE: 0	
#POP3HEADERMODE: 1	
#POP3PORT: 110	
#POP3PW: ""	
#POP3SERV: ""	
#POP3UN: ""	
#DOMAIN: ""	
#SENDERADDR: ""	
#SENDERNAME: ""	
#SMTPPORT: 25	
#SMTPPW: ""	
#SMTPSERV: ""	
#SMTPUN: ""	
#BODY1: ""	
#CCREC1: ""	
#REC1: ""	
#SUBJ1: ""	
#BODY2: ""	
#CCREC2: ""	
#REC2: ""	
#SUBJ2: ""	
#BODY3: ""	
#CCREC3: ""	
#REC3: ""	
#SUBJ3: ""	
#DLEMODE: 1,1	
#TCPPORT: 1,0	
#TCPSERV: 1,""	

Chapter 8 – IP Commands for MT100SEM Only

Set Commands

#DHCP – DHCP On or Off

Description: Enables or disables DHCP.

Syntax: **AT#DHCP=<0,1>**

Values:
AT#DHCP=**0** Turn off DHCP (sets IP Address, Netmask, and Gateway back to defaults)
AT#DHCP=**1** Turn on DHCP (automatically assigns IP Address, Netmask, and Gateway)
AT#DHCP**?** View current setting.

Default: **0**

#IPADDR – Set IP Address

Description: Set the IP Address.

Syntax: **AT#IPADDR="x.x.x.x"**

Values:
AT#IPADDR="**x.x.x.x**" "x" stands for a number 0-255
AT#IPADDR**?** View the IP Address

Default: **192.168.2.3**

#IPGATEWAY – Set Gateway Address

Description: Set the Gateway Address.

Syntax: **AT#IPGATEWAY="x.x.x.x"**

Values:
AT#IPGATEWAY=**"x.x.x.x**" "x" stands for a number 0-255
AT#GATEWAY**?** View the Gateway Address

Default: **192.168.2.1**

#IPNETMASK – Set Netmask Address

Description: Set the Netmask Address.

Syntax: **AT#IPNETMASK="x.x.x.x"**

Values:
AT#IPNETMASK="**x.x.x.x**" x stands for a number 0-255
AT#NETMASK**?** View the Netmask Address

Default: **255.255.255.0**

#EMACSPD – Set Ethernet Connection

Description: Set the Ethernet connection at 10 or 100 full or half duplex.

Syntax: **AT#EMACSPD=x** where x stands for 0,1,2,3,4

Values:

AT#EMACSPD=0	Auto Sensing
AT#EMACSPD=1	10Mbis half duplex
AT#EMACSPD=2	10Mbis full duplex
AT#EMACSPD=3	100Mbis half duplex
AT#EMACSPD=4	100Mbis full duplex
AT#EMACSPD?	View the current setting. Can view with AT#VALL or AT#VIP also.

Default: **AT#EMACSPD=0**

Display Command

#VIP – View Address Settings

Description: Display the address settings.

Syntax: **AT#VIP**

Values: None

Display Example:

```
#DHCP:0
#IPADDR: "192.168.2.3"
#IPGATEWAY: "192.168.2.1"
#IPNETMASK: "255.255.255.0"
#DNSSERV1: "0.0.0.0."
#DNSSERV2: "0.0.0.0."
```

Chapter 9 – Auto Discovery Commands for MT100SEM Only

A Note About Auto Discovery: Auto Discovery will broadcast information; however, at this time, the information cannot be updated using the Auto Discovery Manager.

Set Commands

#AUTODISC – Auto Discovery On/Off

Definition: This command turns Auto Discovery On or Off.

Syntax: **Set Value: AT#AUTODISC=<value>**

View Value: **AT#AUTODISC?** or **AT#VAUTODISC, AT#VALL**

Values:

0	OFF
1	ON

Default: **1**

#AUTODISCTIMER – Auto Discovery Timer

Definition: This command sets the Auto Discovery timer.

Syntax: **Set Value: AT#AUTODISCTIMER=<value>**

View Value: AT#AUTODISCTIMER? or **AT#VAUTODISC, AT#VALL**

Values: 0 to 60 seconds

Default: **10**

#AUTODISCPORT – Auto Discovery Port

Definition: This command sets the Auto Discovery port.

Syntax: **Set Value: AT#AUTODISCPORT=<value>**

View Value: AT#AUTODISCPORT? or **AT#VAUTODISC, AT#VALL**

Values: 0 to 65535

Default: **1020**

#AUTODISCHOST – Set Auto Discover Host Name

Definition: This command sets the Host Name that will appear in the Auto Discovery Manager.

Syntax: **Set Value: AT#AUTODISCHOST ="<value>"**

View Value: AT#AUTODISCHOST? or **AT#VAUTODISC** or **AT#VALL**

Values: Alphanumeric ASCII text string up to 120 characters.

Default: **MT100SEM**

#AUTODISCUSERNAME – Set Auto Discover User Name

Definition: This command sets the User Name that will appear in the Auto Discovery Manager.

Syntax: **Set Value: AT#AUTODISCUSERNAME =<setting>**

View Value: AT#AUTODISCUSERNAME? or **AT#VAUTODISC** or **AT#VALL**

Values: Text string up to 120 characters.

Default: None

#AUTODISCPASSWORD – Sets Auto Discover Password

Definition: This command sets the Password that will appear in the Auto Discovery Manager.

Syntax: **Set Value: AT#AUTODISCPASSWORD =<setting>**

View Value: AT#AUTODISCPASSWORD? or **AT#VAUTODISC** or **AT#VALL**

Values: Text string up to 120 characters.

Default: None

Display Command

#VAUTODISC – Display Auto Discovery Commands

Definition: This command displays all the AT# parameters related to the Auto Discovery configuration.

Syntax: **AT#VAUTODISC**

Command	Possible Responses
AT#VAUTODISC Note: View Auto Discovery parameters	#AUTODISC: 1 #AUTODISCPORT: 1020 #AUTODISCTIMER: 10 #AUTODISCHOST: MT100SEM #AUTODISCUSERNAME: "admin" #AUTODISCPASSWORD: "" OK

Chapter 10 – AT Commands for Non-MT100SEM Devices

Set Commands

+WOPEN – Set IP Mode or Modem Mode

Definition: This command sets the mode of operation.

Syntax: **Set Value: AT+WOPEN=<value>**
View Value: AT+WOPEN? (Displays the current mode of operation).

Values:
0 Modem Mode
1 IP Mode

Default: **1**

#AUTHENT – Authentication Type

Definition: This command sets the PPP authentication type: PAP, CHAP, or NONE.

Syntax: **Set Value: AT#AUTHENT="<value>"**
View Value: AT#AUTHENT? (Displays current PPP authentication type).

Values:
PAP
CHAP
NONE

Default: NONE

#ANSWERMODE – Set Answering Mode

Definition: The TCP/IP stack can handle incoming calls. This parameter defines how the TCP/IP stack will behave when receiving an incoming call.

Syntax: **Set Value: AT#ANSWERMODE=<value>**
Get Value: AT#ANSWERMODE? or AT#VPHY, AT#VALL

Values:

0 **Ignore.** Ignores the incoming call. In this case, it is the responsibility of the host to accept/not accept the incoming call by issuing the AT#ACCEPT command.

1 **Automatic Answer.** The TCP/IP stack goes off hook and accepts the incoming call. As described below, the calling number must match the one specified in the CALLSCREENNUM parameter. (The RINGCOUNT parameter must be > 0).

2 **Static Callback.** The TCP/IP stack ignores the incoming call and then automatically dials (DIALN1 or DIALN2 determined by the DIALSELECT parameter) by issuing an AT#CONNECTIONSTART command. The calling number must match the one specified in the CALLSCREENNUM parameter. The RINGCOUNT parameter must be > 0.

3 **Dynamic Callback.** The TCP/IP stack ignores the incoming call and then automatically dials the calling number by issuing an AT#CONNECTIONSTART command. For this feature, the Caller ID service is mandatory. As described below, the calling number must match the one specified in the CALLSCREENNUM parameter.

Default: **0**

Note: If #ANSWERMODE is set to a value other than 0, the user must enable caller ID. To enable caller ID, use the command **AT+CLIP=1**.

#CALLBACKTIMER – Set Automatic Call-Back Wait

Definition: This parameter defines the number of seconds the TCP/IP stack will wait before an automatic call-back operation occurs after receiving an incoming call. It applies only when the ANSWERMODE parameter is set to an automatic call-back mode (value>1). This timer starts at the end of the ringing signal.

Syntax: **Set Value: AT#CALLBACKTIMER=<value>**

Get Value: AT#CALLBACKTIMER? or AT#VPHY, AT#VALL

Values: Integer between 2 and 255 inclusive. This timer is set in seconds.

Default: **2**

#CALLSCREENNUM – Set Caller ID Service

Definition: When receiving an incoming call, the caller identification (Caller ID) service allows the TCP/IP stack to identify the phone number of the remote caller. This information is helpful in preventing unauthorized callers to trigger actions on the TCP/IP stack.

This parameter allows the user to filter the incoming calls when the ANSWERMODE parameter is set to an automatic mode (value>0). This filtering doesn't apply when the ANSWERMODE parameter is set to 0. In this case, it is the host's responsibility to accept or reject the incoming call. If an incoming phone number is not authorized, the TCP/IP stack will ignore it.

Syntax: **Set Value: AT#CALLSCREENNUM=<value>**
Get Value: AT#CALLSCREENNUM? or **AT#VPHY, AT#VALL**

Values: **0** Zero. Remote caller not authorized.

***** All. No filtering is applied on incoming calls. All remote phone numbers are authorized. This value must be set to receive incoming calls when Caller ID service is not available.

Not valid for Analog Devices.

Decimal Phone Number. Only the phone number configured here is authorized for incoming calls. Alphanumeric ASCII text string up to 64 characters.

Default: **0**

#KEEPALIVEMODE – Keepalive Mode

Definition: If the keepalivemode is not equal to zero, then the keepalive function will trigger. The function allows a persistent physical connection to be established. If enabled and a remote host is not reached, the physical connection is disconnected (if connected) and reestablished.

Syntax: **Set Value: AT#KEEPALIVEMODE=<Keepalivemode value>**

View Values: AT#KEEPALIVEMODE? or **AT#VKEEPALIVE** or **AT#VALL**

Values: **0** Keepalivemode is disabled
1 Keepalivemode is enabled using a TCP connection
2 Keepalivemode is enabled using ICMP

Default: **0**

Example: AT#KEEPALIVEMODE=1

#KEEPALIVEPORT – KEEPALIVE Port Number

Definition: If the keepalive session is TCP, then keepaliveport controls the remote host port number.

Syntax: **Set Value: AT#KEEPALIVEPORT=<Port Number value>**
View Values: AT#KEEPALIVEPORT? or **AT#VKEEPALIVE** or **AT#VALL**

Values: Set the port number that the Profile will use:
From 1 to 5 numeric digits (0 to 9).
Note:
Numbers above 65535 are illegal since port identification fields are 16-bits long in the IDP header.

Default: **0**

Example: AT#KEEPALIVEPORT=xxxxx

#KEEPALIVESERV – KEEPALIVE Server Address

Definition: The remote address for both TCP and ICMP keepalive.

Syntax: **Set Value: AT#UDPSERV="<Keepalive Server Addressvalue>"**
View Value: AT#KEEPALIVESERV? or **AT#VKEEPALIVE** or **AT#VALL**

Values: Set the Remote Server Address value that the Profile will use:
32-bit number is dotted-decimal notation (i.e., xxx.xxx.xxx.xxx) or
Alphanumeric ASCII text string up to 120 characters if DNS is working.

Default: None

Example: AT#KEEPALIVESERV="xxx.xxx.xxx.xxx" or AT#KEEPALIVESERV="some.address.com"

#KEEPALIVEDELAY – Keepalive Delay

Definition: This parameter determines the delay before the keepalive mechanism will check the remote host for connectivity.

Syntax: **Set Value: AT#KEEPALIVEDELAY=<Data Delay value>**
View Value: AT#KEEPALIVEDELAY? or **AT#VKEEPALIVE** or **AT#VALL**

Values: Set the delay value in minutes
Integer between 0 and 65535

Default: **100**

Example: AT#KEEPALIVEDELAY=xxxxx

#OUTPORT – UDP/TCP Local Port Number

Definition: Controls the local port that the UDP/TCP connection is bound to.

Syntax: **Set Value: AT#OUTPORT=<Port Number value>**
View Values: AT#OUTPORT? or **AT#VKEEPALIVE** or **AT#VALL**

Values: Set the port number local connection will use:
From 1 to 5 numeric digits (0 to 9).

Note:
Numbers above 65,535 are illegal since port identification fields are 16-bits long in the IDP header.

Default: **37500**

Example: AT#OUTPORT=xxxxx

#RESET – Reset Main Processor or Internal Modem

Definition: This command will force a reset on the Stack and internal radio or just the internal radio.

Syntax: **AT#RESET =<value>**

Values: **0** Reset the Stack and internal modem
1 Reset the internal modem only

Example: AT#RESET=0

#AUTORESET - Reset the Processor at a Given Time

Definition: Set an amount of time when the modem will reset after no input has been detected on the serial port.

Syntax: **Set Value: AT#AUTORESET=<time in minutes>**
View Values: AT#AUTORESET? or **AT#VALL**

Values: 0-65535 the time in minutes
From 1 to 5 numeric digits (0 to 9).

Default: **0**

Example: AT#AUTORESET=xxxxx

#PERSISTENTSOCKET - Persistent Socket

Definition: If enabled and the physical connection is present, a socket will be created automatically.

Syntax: **Set Value: AT#PERSISTENTSOCKET=<Persistentsocket value>,<Result Codes>**
View Values: **AT#PERSISTENTSOCKET?** or **AT#VALL**

Values: **0** Persistent socket is disabled
1 The TCP originate is started (#OTCP=1)
2 The TCP listener is started (#LTCPSTART=1)
3 The UDP originate is started (#OUDP=1)
4 The UDP listener is started (#LUDPSTART=1)

Result Codes: **0** All result codes will be suppressed during socket connection/disconnection
1 All result codes will be displayed during socket connection/disconnection

Defaults: **0,1**

Example: AT#PERSISTENTSOCKET=1,0

#ATCMD – Set Default Dial Time AT Commands

Definition: This command allows the user to customize AT commands on dial up.

Syntax: **Set Value:** AT#ATCMD=id,"Valid AT command"
View Value: AT#ATCMD=id or AT#VALL

id: Enter a Profile Setting: 0, 1, 2, 3 {0,1,2,3}

Values: Set the valid AT command (Do not include AT or issue &F, as the command will error)

Defaults: **For Analog Devices:** 0, "-STE=1"
1, "+MS=V34"
2, ".."
3, ".."

For All Other Devices: 0, ".."
1, ".."
2, ".."
3, ".."

#REDIALCOUNT – Set Connection Attempts Allowed

Definition: This parameter indicates how many unsuccessful connection attempts the TCP/IP stack software will make before terminating the connection attempt activity.

Syntax: **Set Value: AT#REDIALCOUNT=<value>**

Get Value: AT#REDIALCOUNT? or **AT#VPHY**, **AT#VALL**

Values: Integer between 0 and 14, inclusive.

If the value is set to 0, the TCP/IP stack software will not make any call retry.

Default: **5**

#REDIALDELAY – Set Call Retry Delay

Definition: This parameter controls the delay (in seconds), if any, that will exist between each call retry.

Syntax: **Set Value: AT#REDIALDELAY=<value>**

Get Value: AT#REDIALDELAY? or **AT#VPHY**, **AT#VALL**

Values: Integers 0 and 5–14 inclusive.

If this parameter is configured to zero, the TCP/IP stack software will attempt another connection immediately after terminating the previous unsuccessful attempt.

Default: **5**

#PHYTIMEOUT – Set Inactivity Time

Definition: This parameter is used by the TCP/IP stack software in order to terminate connections to the telephone line when a long period elapses without activity. "Without activity" is defined as a period when no data is transferred between the Internet and the TCP/IP stack software or between the TCP/IP stack software and the attached equipment. This timer prevents the telephone line from being connected indefinitely for any reason.

Note: When the inactivity timer expires, the modem ends the communication.

Syntax: **Set Value: AT#PHYTIMEOUT=<value>**

Get Value: AT#PHYTIMEOUT? or **AT#VPHY**, **AT#VALL**

Values: Integer between 1 and 255 inclusive. This timer is set in minutes.

Default: **15**

Return Codes: TIMEOUT: The inactivity timer is reached; the modem ends the communication.

#RINGCOUNT – Set Number of Rings/Automatic Operation

Definition: This parameter defines the number of rings that will be counted before an automatic operation occurs when receiving an incoming call.

It applies only when the ANSWERMODE parameter is set to an automatic mode (value>0).

If the ANSWERMODE parameter is used, the RINGCOUNT value must also be >0.

Syntax: **Set Value: AT#RINGCOUNT=<value>**

Get Value: AT#RINGCOUNT? or **AT#VPHY**, **AT#VALL**

Values: Integer between 0 and 15 inclusive.

0 = modem will not answer.

1-15 = modem will answer on the ring number selected.

Default: **0**

#DIALN1 – Set Primary Dial-Up Number

Definition: This parameter is the primary dial-up phone number that will connect with the local ISP. Length depends on country/region.

Syntax: **Set Value:** AT#DIALN1="<value>"
Get Value: AT#DIALN1? or AT#VPHY, AT#VALL

Values: Decimal phone numbers.

Default: None

#DIALN2 – Set Secondary Dial-Up Number

Definition: This parameter is the secondary dial-up phone number that will connect with the local ISP. Length depends on country.

Syntax: **Set Value:** **AT#DIALN2="<value>"**
Get Value: **AT#DIALN2?** or **AT#VPHY**, **AT#VALL**

Values: Decimal phone numbers.

Default: None

#DIALSELECT – Set Internet Connection Number

Definition: The value of this command determines the number called to establish an Internet connection. It configures the TCP/IP stack software to use the primary dial-up number or the secondary dial-up number.

Syntax: **Set Value:** **AT#DIALSELECT=<value>**
Get Value: **AT#DIALSELECT?** or **AT#VPHY**, **AT#VALL**

Values: **1** Use primary dial-up number.
2 Use secondary dial-up number

Default: **1**

#ISPPW – Set ISP Password

Definition: This parameter sets the password for the ISP account. When communication is initiated and once the physical (modem) connection has been established with the ISP, the TCP/IP stack software must provide the ISP with the password associated with the account to be used.

Syntax: **Set Value:** **AT#ISPPW="<value>"**
Get Value: **AT#ISPPW?** or **AT#VPPP**, **AT#VALL**

Values: Alpha-numeric ASCII text string up to 64 characters.

Default: None

#ISPUN – Set ISP User Name

Definition: This parameter sets the user name of the ISP account. When communication is initiated and the physical (modem) connection has been established with the ISP, the TCP/IP stack software must provide the ISP with the user name associated with the account to be used.

Syntax: **Set Value:** **AT#ISPUN="<value>"**
Get Value: **AT#ISPUN?** or **AT#VPPP**, **AT#VALL**

Values: Alpha-numeric ASCII text string up to 64 characters.

Default: None

Incoming Call Management Commands

#ACCEPT – Answer Incoming Call

Definition: This command directs the TCP/IP stack to answer an incoming call. When the TCP/IP stack receives an incoming call, it sends over the serial port the "RING" messages. Depending on the value of the ANSWERMODE parameter, the TCP/IP stack may answer automatically or not.

If ANSWERMODE is set to 0, it is the host that is responsible for answering the incoming call.

Once the physical layer is up, the TCP/IP stack runs.

The AT#CONNECTIONSTOP command ends the connection by going on hook.

Values: **Read parameters:**

ANSWERMODE
IPSPW
ISPUN
PPPMYIP

Syntax: **AT#ACCEPT**

Command	Possible Responses
AT#ACCEPT Note: Manual acceptance of an incoming call	OK Note: Beginning of the call setting process CONNECT <speed> Note: Modem speed negotiated between both sides xxx.xxx.xxx.xxx Note: IP address indication attributed to the TCP/IP stack PPP OK Note: The software is ready to run IP applications to send/receive data
AT#ACCEPT Note: Manual acceptance of an incoming call	NO CARRIER Note: The modem handshaking process with the remote host is interrupted or unsuccessful
AT#ACCEPT Note: Manual acceptance of an incoming call	PPP ERROR Note: The PPP negotiation has failed (check ISPUN, ISPPW, PPPMODE, and the configuration of the PPP peer)

#CONNECTIONSTOP – Stop Communication

Definition: This command directs the TCP/IP stack to end a communication previously established with a CONNECTIONSTART command or AT#ACCEPT.

Values: No TCP/IP parameter is used for the execution of this command.

Syntax: **AT#CONNECTIONSTOP**

Command	Possible Responses
AT#CONNECTIONSTOP Note: Disconnect	OK Note: Phone line is released

#CONNECTIONSTART – Start Communication

Definition: This command directs the TCP/IP stack to originate an outgoing call.

Upon receiving this command, the TCP/IP stack attempts to complete a connection session.

The TCP/IP stack will dial the number according to the Dial Option parameter (DIALN1 or DIALN2 depending on DIALSELECT). If an error occurs, the TCP/IP stack automatically re-attempts the call origination attempt, according to the REDIALCOUNT parameter. Once the physical layer is up, the TCP/IP stack runs.

Values: **Read Parameters:**

DIALN1
DIALN2
DIALSELECT
ISPPW
ISPUN
REDIALCOUNT
REDIALDELAY

Syntax: **AT#CONNECTIONSTART**

Command	Possible Responses
AT#CONNECTIONSTART Note: Request connection to network	DIALING Note: Phone line is available 2124560123 Note: Dial DIALN1 or DIALN2 number depending on DIALSELECT CONNECT 9600 Note: Modem speed negotiated between both sides. If the TCP/IP stack is configured for modem only operation (PPPMODE parameter), there are no more return codes 213.192.200.4 Note: IP address attributed to the TCP/IP stack OK_Info_PPP Note: As soon as the TCP/IP stack software displays this message, it is ready to receive commands
AT#CONNECTIONSTART	BUSY Note: A busy signal is detected on the remote site TCP/IP stack will wait REDIALDELAY seconds then dial again. The redialing will continue until success or until the number of call retries defined in parameter REDIALCOUNT has been reached.

#AUTOCONNECT – Automatically Connect the PPP Link

Definition: **#AUTOCONNECT** enables/disables the automatic link connection. The APN or ISP values need to be set correctly.

Values: 0 **AUTOCONNECT** disabled
1 **AUTOCONNECT** enabled

Syntax: **Set Value: AT#AUTOCONNECT=<value>**
Get Value: AT#AUTOCONNECT? or **AT#VALL**

Default: 0

#PPPMODE – Set TCP/IP Behavior

Definition: The TCP/IP stack can manage the access layer in different ways. This parameter selects the way the TCP/IP stack must run once the physical is layer successfully established.

Syntax: **Set Value: AT#PPPMODE=<value>**

Get Value: AT#PPPMODE? or **AT#VPPP**, **AT#VALL**

Values: **3** **Standard PPP.** The TCP/IP stack behaves as a PPP client for outgoing calls and as a PPP client for incoming calls.

Default: **3**

#PPPMYIP – Set IP Address When Using PPP

Definition: When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter setting), it is in charge of the IP address attribution mechanism.

Once the PPP authentication is successfully achieved, the remote PPP peer asks the TCP/IP stack for an IP address. Then the related PPP layer, called IPCP, suggests an IP address to the peer that has been previously stored in the TCP/IP stack parameters. If the remote accepts this address, the IP link is then established.

This parameter defines the IP address to be attributed to the TCP/IP stack when the PPP Server mode is running.

Syntax: **Set Value: AT#PPPMYIP=<value>**

Get Value: AT#PPPMYIP? or **AT#VPPP**, **AT#VALL**

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx).

Default: **0.0.0.0**

#PPPPEERIP – Set IP Address for Remote PPP

Definition: When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter setting), it is in charge of the IP address attribution mechanism.

Once the PPP authentication is successfully achieved, the remote PPP peer asks the TCP/IP stack for an IP address. Then the related PPP layer, called IPCP, suggests an IP address to the peer that has been previously stored in the TCP/IP stack parameters. If the remote accepts this address, the IP link is then established.

This parameter defines the IP address to be attributed to the remote PPP peer when the PPP Server mode is running.

Syntax: **Set Value: AT#PPPPEERIP=<value>**

Get Value: AT#PPPPEERIP? or **AT#VPPP**, **AT#VALL**

Values: 32-bit number in dotted-decimal notation (i.e., xxx.xxx.xxx.xxx).

Default: **0.0.0.0**

#PPPSERVUN – Set User Name for Remote PPP Client

Definition: When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter), it checks the remote PPP client login/password before granting access to the server.

This parameter defines the user name that must be specified by the remote PPP client.

Syntax: **Set Value: AT#PPPSERVUN=<value>**

Get Value: AT#PPPSERVUN? or **AT#VPPP**, **AT#VALL**

Values: Alpha-numeric ASCII text string up to 64 characters.

Default: None

#PPPSERVPW – Set Remote PPP Client Password

Definition: When the TCP/IP stack behaves as a PPP server (according to the PPPMODE parameter setting), it checks the remote PPP client login/password before granting access to the server.

This parameter defines the password that must be specified by the remote PPP client.

Syntax: **Set Value: AT#PPPSERVPW=<value>**

Get Value: AT#PPPSERVPW? or **AT#VPPP**, **AT#VALL**

Values: Alpha-numeric ASCII text string up to 64 characters.

Default: None

#APNPW – Set Access Point Name Password

Definition: #APNPW is the Access Point Name password parameter coming with the APNUN from the GSM operator and provides GPRS access.

Syntax: **Set Value: AT#APNPW=<value>**

Get Value: AT#APNPW? or **AT#VGPRS**, **AT#VALL**

Values: Alphanumeric ASCII text string up to 120 characters.

Default: None

#APNSERV – Set Access Point Name

Definition: #APNSERV is the Access Point Name parameter coming from the GSM operator for providing GPRS access.

Syntax: **Set Value: AT#APNSERV=<value>**

Get Value: AT#APNSERV? or **AT#VGPRS**, **AT#VALL**

Values: Alphanumeric ASCII text string up to 120 characters.

Default: None

#APNUN – Set Access Point User Name

Definition: #APNUN is the Access Point Name User Name parameter coming with the APNPW from the GSM operator for providing GPRS access.

Syntax: **Set Value: AT#APNUN=<value>**

Get Value: AT#APNUN? or **AT#VGPRS**, **AT#VALL**

Values: Alphanumeric ASCII text string up to 120 characters.

Default: None

#GPRSCID – Set PDP Context Identifier

Definition: #GPRSCID is the PDP context identifier which specifies a particular PDP context definition. This parameter is local and may be used in other PDP context-related commands.

Syntax: **Set Value: AT#GPRSCID=<value>**

Get Value: AT#GPRSCID? or **AT#VGPRS**, **AT#VALL**

Values: Numeric between 1 and 4 inclusive.

Default: **1**

#GPRSMODE – Activate Switching Between GSM and GPRS

Definition: #GPRSMODE configures the activation of the software for switching between GSM and GPRS.

Syntax: **Set Value: AT#GPRSMODE=<value>**

Get Value: AT#GPRSMODE? or **AT#VGPRS**, **AT#VALL**

Values: **0** Configured for GSM use.

1 Configured for GPRS use.

Default: **1**

Display Commands

#DISPLAYIP – Display IP Addresses

Definition: This command allows the attached host to view the attributed IP addresses that have been attributed during the IPCP phase of the PPP negotiation. Both local and remote PPP peer IP addresses are displayed. This command should be issued only once the PPP OK message has been received from the TCP/IP stack.

Syntax: **AT#DISPLAYIP**

Response: MY IP: xxx.xxx.xxx.xxx
PEER IP: **xxx.xxx.xxx.xxx**

Values: **Read Parameters:**
PPPMYIP
PPPPEERIP

#VPPP – Display PPP Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the PPP layer configuration.

Values: **Read Parameters:**
ISPPW
ISPUN
PPPMYIP
PPPPEERIP

Syntax: **AT#VPPP**

Command	Possible Responses
AT#VPPP	#ISPUN: “myispun” #ISPPW: “myisppwd: #PPPMODE: 1 #PPPMYIP: “0.0.0.0” #PPPPEERIP: “0.0.0.0” OK

#VPHY – Display Physical Parameters

Definition: This command directs the TCP/IP stack to display all the AT# parameters related to the physical layer configuration.

Values:
ANSWERMODE
CALLBACKTIMER
CALLSCREENNUM
DIALN1
DIALN2
DIALSELECT
PHYTIMEOUT
REDIALCOUNT
REDIALDELAY
RINGCOUNT

Syntax: **AT#VPHY**

Command	Possible Responses
AT#VPHY	#ANSWERMODE: 0 #CALLBACKTIMER: 2 #CALLSCREENNUM: “0” #DIALN1: “” #DIALN2: “” #DIALSELECT: 1 #PHYTIMEOUT: 15 #REDIALCOUNT: 5 #REDIALDELAY: 5 #RINGCOUNT: 0 OK

GPIOs

About GPIO

If the specific hardware allows GPIO (check availablity with AT#GPIO=1), then you may specify certain external events from the GPIOs to trigger specific stack functions; i.e., email, upload files, download files. **In order for this to work, you must configure the specific function first.** (Original sentence was: The requirements are the configuration of the specific function. Check to see that this meaning has not been lost.) For instance, if SENDMAIL1 will be called, then all settings pertaining to SENDMAIL1 must be configured before the #GPIO command is configured.

#GPIO – Enable/Disable and Configure External GPIO Driven Events

Definition: If GPIO is enabled and configured correctly, a specific external trigger (Digital HI/LOW or Analog value from 0-3.3V) will trigger a specified function from the stack.

Syntax:
Set Value: AT#GPIO=ID, Pin Function, Trigger Level, Debounce, Stack Function, Hangup
View Values: AT#GPIO=<n> where n is the Pin number or AT#VALL

ID: Pin number as described in the hardware documentation
Values: 1 - 5

Pin Function: Allows the pin to be a Digital Input, Digital Output or ADC input
Values:
0 - Digital Input with internal pullup
1 - Digital Input without internal pullup
2 - Digital Output
3 - ADC input

Trigger Level: The Voltage level at which GPIO will trigger.
For Digital input, a value greater than 0 indicates high. 0 indicates low.
For Digital output, a positive value will drive the line high. A zero value will drive the line low.
For ADC input, the trigger value will be in the range of 0 to 3300 millivolts.
Values: 0 - 3300 in milliVolts

Debounce: The time between successive reads on the pin.
Values: 0 - 65535 milliseconds

Stack Function: The desired stack function to be **specified** on a GPIO trigger. The stack function must be properly configured; i.e., for a sendmail function, all proper sendmail items must be configured.
Values:
0 - Disabled (The GPIO will not trigger)
1 - #SENDMAIL1
2 - #SENDMAIL2
3 - #SENDMAIL3
4 - #PUTMAIL
5 - #GETMAIL
6 - #LTCPSTART=1
7 - #OTCP=1
8 - #OUDP=1
9 - #LUDPSTART=1

Hangup: Determines whether to disconnect from the network after GPIOs have been processed.
Values:
0 - Physical connection will disconnect when all GPIOs have been processed
1 - Physical connection will stay connected.
2 - Physical connection will disconnect when all GPIOs have been processed; suppress responses
3 - Physical connection will stay connected; suppress responses

Default: 1,0,0,0,0,0
2,0,0,0,0,0
3,0,0,0,0,0
4,0,0,0,0,0
5,3,0,0,0,0

Example: AT#GPIO=1,0,1,180,1,0

PIN 1 would be configured as a Digital Input with internal pullup (0).
A positive voltage will trigger the pin (1).
A Debounce of time of 180 ms (180) will be used.
#SENDMAIL1 will be called if the GPIO triggers (1).
After processing the GPIO, the physical connection will be disconnected (0) and all responses will be printed to the user.

Legacy Mode

#LEGACYMODE - Legacymode Enable/Disable

About Legacymode:

Legacy mode allows for smoother transition to TCP/IP based connectivity from standard analog interface connections.

When #LEGACYMODE=1, the stack will be in pseudo legacy mode where the commands &S, &D, &C, A, D, S0, E, V operate independently of the stack. The commands will be saved to a different NVRAM location and will not affect the IP stack once legacy mode has been exited.

Once in the Legacymode, the user can configure the #TCP values and issue an ATD command to make a connection to the remote host. If necessary, the user can also issue ATDxxxxxxxxxxxxppppp where x is the decimal address and ppppp is the remote port. The stack will perform the physical connection and present the user with the proper messages:

NO CARRIER
NO ANSWER
CONNECT
OK
ERROR

Definition: If Legacymode is enabled, the AT interface tries to behave like a legacy analog modem. ATD and ATA will function on the TCP stack instead of the physical connection. Responses will behave like that of an analog modem.

Syntax: **Set Value: AT#LEGACYMODE=1**
View Values: AT#LEGACYMODE?

Values: **0** Disable legacymode
1 Enable legacymode

Default: **0**

Example: AT#LEGACYMODE=1

Chapter 11 – Response Messages & Error Codes

Response Messages

Standard AT Messages		
Numeric	**Verbose**	**Description**
0	OK	Operation or command success
3	NO CARRIER	No physical layer connection
7	BUSY	Destination busy
8	NO ANSWER	No answer from destination
4	ERROR	Operation or command unsuccessful
2	RING	Incoming call indication

Information Messages		
Numeric	**Verbose**	**Description**
1025	OK_Info_DataBegin	Start of data
1028	OK_Info_WaitingForData	Send data
3074	OK_Info_SocketClosed	Socket connection closed successfully
3072	OK_Info_NoMail	No mail to retrieve on server
3073	OK_Info_Mail	Mail ready to be retrieved on server
3077	OK_Info_PPP	PPP connection successful

Error Codes

Error Codes	
Numeric	**Description**
34817	Bad command : Unknown command
34819	Bad command : Syntax error
34824	Bad command : EEPROM write failed
34881	Bad command : Command to long
34882	Bad command : Bad command argument value
34883	Bad command : High level Internet configuration only command
35840	Physical layer : Modem is already running
35862	Physical layer : Timeout, no activity on network connection
35865	Physical layer : Module is not attached to the network
35866	Physical layer : Invalid event during activation process
35867	Physical layer : Physical layer connection is currently not active
35869	Physical layer : Invalid incoming call type
35870	Physical layer : Incoming call CLI not provided
36872	IP Connectivity library internal error : internal resource unavailable.
36929	IP Connectivity library : Bad parameter configuration attempt
37122	IP Connectivity library : Another internal application is already running
37123	IP Connectivity library : Service is running. Unable to set parameter
37124	IP Connectivity library : Data buffer oversized
37125	IP Connectivity library : No UDP datagram received
37952	Distant : TCP session closed (TCP Context cancelled)
37964	Distant : No response from server
37966	Distant : TCP session closed by peer (FIN received from peer)
38016	Distant : Open session attempt failed
38017	Distant : Data send attempt failed
38018	Distant : Close session attempt failed
38023	Distant : File deletion attempt failed
38024	Distant : Data retrieve attempt failed
38025	Distant : Email retrieve attempt failed
38026	Distant : Email header receive failed
38027	Distant : No answer from DNS servers or domain name resolution could not be completed by the server.
38028	Distant : Sender email address rejected by server
38029	Distant : Recipient email address rejected by server
38030	Distant : CC Recipient email address rejected by server
38031	Distant : Email body send request rejected by server
38080	Distant : Username rejected by server
38081	Distant : Password rejected by server

Chapter 12 – Response Message Examples

Response Messages Examples

Sending/Retrieving Email Response Messages

Sending an Email: AT#PUTMAIL

Commands	Responses
AT#SMTPSERV="smtp.domain.com" *Note: SMTP server used*	OK
AT#DOMAIN="domain.com" *Note: Domain name*	OK
AT#SENDERNAME="Test module" *Note: Sender name*	OK
AT#SENDERADDR="module@domain.com" *Note: Sender email address*	OK
AT#SMTPUN="Name" *Note: SMTP user name*	OK
AT#SMTPPW="Password" *Note: SMTP user password*	OK
AT#SMTPORT="Port" *Note: Port used by SMTP server*	OK
AT#REC1=recipient@domain.com or AT#REC2 or AT#REC3 *Note: Recipient email address*	OK
AT#CCREC1=ccrecipient@domain.com or AT#CCREC2 or AT#CCREC3 *Note: Carbon Copy recipient*	OK
AT#SUBJ1="Email Subject" *Note: Email Subject*	OK
1. AT#PUTMAIL *Note: Send an email (type the email text and then the end sequence)* **OR** 1. AT#BODY1 (or AT#BODY2 or AT#BODY3) *Note: Write an email for a predefined message.* AND 2. AT#SENDMAIL1 (or AT#SENDMAIL2, AT#SENDMAIL3) *Note: Send the predefined message.*	OK_Info_WaitingForData *Note: The software is ready to receive incoming data (not echoed)* *At the end of date, the* [CR][LF] . [CR][LF] sequence ends the email. This sequence can be sent by a keyboard using: ENTER CTRL+ENTER . ENTER CTRL+ ENTER OK OK

Retrieving an Email: AT#GETMAIL

Commands	Responses
AT#POP3SERV="pop3.domain.com" Note: POP3 server used	OK
AT#POP3UN="module@domain.com" Note: POP3 username (not always the complete email address). It is the POP3 login	OK
AT#POP3PW="password" Note: POP3 password	OK
AT#POP3PORT="110" Note: POP3 port	OK
AT#GETMAIL Note: Retrieve an email	OK_Info_Mail Note: The software switches from command mode to data mode for receiving the email content. Data Data . Note: At the end of data, the [CR][LF] . [CR][LF] sequence notifies the end of the data mode OK Note: The email is successfully retrieved

FTP: Download / Upload Files Response Messages

Upload a file to an FTP Server: AT#FTPPUT

Commands	Responses
AT#FTPSERV="ftp.domain.com" Note: FTP server used	OK
AT#FTPUN="ftplogin" Note: FTP username	OK
AT#FTPPW="ftppassword" Note: FTP password	OK
AT#FTPPUTFILENAME="upload.text" Note: Name of the file that will be written in the FTP server	OK
AT#FTPPUTPATH="." Note: Path in the server where the file will be written	OK
AT#FTPPUT Note: FTP put	OK_Info_WaitingForData Note: Switch from command to data mode. The host can send the data that will compose the file. (Data not echoed). To notify the end of data, the host has to send [ETX] (CTRL+C on the keyboard). This character is echoed. OK
Note: End of data notified	OK

Download a file from an FTP Server: AT#FTPGET

Commands	Responses
AT#FTPSERV="ftp.domain.com" Note: FTP server used	OK
AT#FTPUN="ftplogin" Note: FTP username	OK
AT#FTPPW="ftppassword" Note: FTP password	OK
AT#FTPGETFILENAME="upload.text" Note: Name of the file stored in the FTP server	OK
AT#FTPGET Note: FTP get	OK_Info_DataBegin Note: Switch from command to data mode. The data is sent over the serial port. To end of data is notified by [ETX] sent over the serial port. It switches from data to command mode. Character is echoed. OK

TCP Socket Response Messages

Open a TCP socket between machines. One machine acts as a caller (TCP client); the other acts as a listener (TCP server). Both machines have to be connected to the Internet and set to the same TCP port.

Act as a TCP Server: AT#LTCPSTART

Commands	Responses
AT#TCPSERV=1,"255.255.255.255" Note: No filter of the incoming TCP client	OK
AT#TCPPORT=1,"23" Note: TCP port between the TCP client and the TCP server must be the same	OK
AT#LTCPSTART=1 Note: Launch the listening mode; waiting for an incoming TCP connection from a TCP client	OK OK_Info_WaitingForData Note: Message sent over the serial port in case of successful TCP socket opening (Telnet for example in the IP address)
Data Note: Data flow is bidirectional	Data Note: Data flow is bidirectional
Note: The socket can be closed locally by the attached host sending an [ETX] character (CTRL+C on the keyboard)	OK
	Note: The socket can be closed by the remote OK_Info_SocketClosed OK

Act as a TCP Client: AT#OTCP

Commands	Responses
AT#TCPSERV=1,"xxx.xxx.xxx.xxx" Note: No filter of the incoming TCP client	OK
AT#TCPPORT=1,"23" Note: TCP port between the TCP client and the TCP server must be the same	OK
AT#OTCP=1 Note: Open as a TCP client, a socket TCP with remote TCP server	OK_Info_WaitingForData Note: Message notifying the socket opening and the switch in data mode
Data Note: Data flow is bidirectional	Data Note: Data flow is bidirectional
Note: The socket can be closed locally by the attached host sending an [ETX] character (CTRL+C on the keyboard)	OK
	Note: The socket can be closed by the remote OK_Info_SocketClosed OK

Note: The closing of the socket can be performed either locally or remotely.

Index