

---

# Wireless GPRS-F4 Modems

**MultiModem<sup>®</sup> GPRS (MTCBA-G)**  
**MMCModem<sup>™</sup> GPRS (MTMMC-G)**  
**SocketModem<sup>®</sup> GPRS (MTSMC-G)**

## IP Connectivity AT Commands Reference Guide



## IP Connectivity AT Commands for Wireless GPRS-F4 Modems Reference Guide

Products that use these commands:

- MultiModem® GPRS (MTCBA-G-F4)**
- MMCModem™ GPRS (MTMMC-G-F4)**
- SocketModem® GPRS (MTSMC-G-F4)**

**S000437C**

### Copyright

This publication may not be reproduced, in whole or in part, without prior expressed written permission from Multi-Tech Systems, Inc. All rights reserved. Copyright © 2007-2008, by Multi-Tech Systems, Inc. Multi-Tech Systems, Inc. makes no representations or warranty with respect to the contents hereof and specifically disclaim any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Multi-Tech Systems, Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Multi-Tech Systems, Inc. to notify any person or organization of such revisions or changes.

### Revisions

Revision Level	Date	Description
<b>A</b>	05/17/07	Initial release. Based on Wavecom IP WIPSoft V2.02 commands.
<b>B</b>	08/16/07	Updated the +WIND command in Appendix A.
<b>C</b>	01/16/08	Updated the cover page and updated the product name of the MMCModem.

### Trademarks

SocketModem®, MultiModem®, Multi-Tech®, and the Multi-Tech logo are registered trademarks of Multi-Tech Systems, Inc. MMCModem is a trademark of Multi-Tech Systems, Inc. WAVECOM®, WISMO®, Open AT® and certain other trademarks and logos appearing on this document, are filed or registered trademarks of Wavecom S.A. in France or in other countries. All other company and/or product names mentioned may be filed or registered trademarks of their respective owners.

### 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 1-763-717-5863

### World Headquarters

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

# Table of Contents

<b>Chapter 1 – Introduction</b> .....	<b>4</b>
Acronyms and Abbreviations .....	4
AT Command Syntax .....	5
Command Line .....	5
Information Responses and Result Codes .....	5
Principles .....	5
<b>Chapter 2 – General Configuration AT Commands</b> .....	<b>7</b>
IP Stack Handling +WIPCFG .....	7
Bearers Handling +WIPBR .....	10
<b>Chapter 3 – IP Protocol Services</b> .....	<b>15</b>
Service Creation +WIPCREATE .....	15
Closing a Service +WIPCLOSE .....	19
Service Option Handling +WIPOPT .....	21
<b>Chapter 4 – Data Exchange for Protocol Services</b> .....	<b>23</b>
File Exchange +WIPFILE .....	23
Socket Data Exchange +WIPDATA .....	25
<b>Chapter 5 – Ping Services</b> .....	<b>31</b>
PING Command +WIPPING .....	31
<b>Chapter 6 – Examples of Application</b> .....	<b>33</b>
TCP Socket .....	33
TCP Client Socket .....	35
UDP Socket .....	37
PING .....	38
FTP .....	39
<b>Chapter 7 – Error Codes</b> .....	<b>40</b>
<b>Appendix A – GSM/GPRS +WIND Command</b> .....	<b>41</b>
General Indications +WIND .....	41
<b>Index</b> .....	<b>44</b>

# Chapter 1 – Introduction

## Acronyms and Abbreviations

APN	Access Point Name
ASCII	American Standard Code for Information Interchange
AT	ATtention
CHAP	Challenge Handshake Authentication Protocol
CHV	Card Holder Verification
CID	Context Identifier
CMUX	Converter Multiplexer
CPU	Central Processing Unit
DNS	Domain Name System
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile communicatio006E
IP	Internet Protocol
IPCP	Internet Protocol Control Protocol
M	Mandatory
MS	Mobile Station
N/A	Not Applicable
MSCHAP	Microsoft Challenge Handshake Authentication
MSS	Maximum Segment Size
NU	Not Used
O	Optional
OS	Operating System
PAP	Password Authentication Protocol
PDP	Packet Data Protocol
PIN	Personal Identity Number
PPP	Point-to-Point Protocol
SIM	Subscriber Information Module
TCP	Transmission Control Protocol
TOS	Type Of Service
TTL	Time To Live
UART	Universal Asynchronous Receiver Transmitter
UDP	User Data Protocol
URL	Uniform Resource Locator
WIP	Wavecom Internet Protocol

## AT Command Syntax

### Command Line

Commands always start by the standard prefix **AT+WIP** and end with the **<CR>** character. Optional parameters are shown in brackets [ ].

**Example:**

AT+WIPcmd=<Param1>[,<Param2>]

<Param2> is optional. When the AT+WIP command is executed without <Param2> the default value of <param2> is used.

### Information Responses and Result Codes

Responses start and end with **<CR><LF>**, except for the **ATV0 DCE** response format and the **ATQ1** (result code suppression) commands.

- If the command syntax is incorrect, the **ERROR** string is returned.
- If the command syntax is correct but transmitted with the wrong parameters, the **+CME ERROR: <Err>** string or the **+CMS ERROR: <SmsErr>** string is returned with adequate error codes if CMEE was previously set to 1. By default, CMEE is set to 0, and the error message is only ERROR.
- If the command line has been executed successfully, an OK string is returned.

In some cases, such as **AT+CPIN?** or (unsolicited) incoming events, the product does not return the OK string as a response.

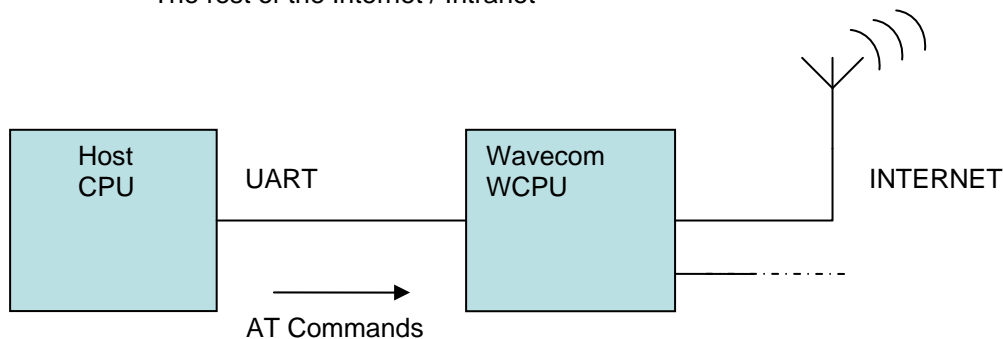
In the following examples **<CR>** and **<CR><LF>** are intentionally omitted.

### Principles

WIPSoft is an Open AT<sup>®</sup> application that implements the TCP/IP protocols using custom AT commands. This Open AT<sup>®</sup> application operates in co-operative mode and must be downloaded to the Wavecom Wireless CPU<sup>®</sup>. The commands are sent from an external application and the corresponding responses are sent back from the Wavecom Wireless CPU<sup>®</sup> to the external application. WIPSoft uses the APIs provided by wipLib and provides custom AT command interface to the external application.

AT+WIP commands involve:

- A host computer, which issues AT+WIP commands
- Wavecom's wireless CPU<sup>®</sup>
- The rest of the Internet / Intranet



**Multiplexing:** Several sockets can be operating at once. The +WIPDATA command allows to temporarily identify the UART in data mode with a given socket. The data written on UART is transferred through the socket. The data which arrives on the socket can be read from the UART. In AT mode, the host receives an unsolicited event when the data arrives on the socket.

**Multiple UARTs:** There can be several UARTs simultaneously active at once, and different UARTs can map a different socket simultaneously. However, you cannot map a single socket on several UARTs simultaneously.

### ***Socket Identification***

Sockets are identified by a pair of numbers: the first one identifies the protocol; the second one identifies a given socket of this protocol.

### ***Possible Protocols***

The possible protocols are:

- 1 = UDP
- 2 = TCP in connect mode (Client)
- 3 = TCP in listen mode (Server)
- 4 = FTP

Two pairs with a different protocol number but the same index identify two distinct sockets.

Example: Both 1,7 and 2,7 are valid identifiers simultaneously; the former identifies a UDP socket and the later, a TCP connected socket.

### ***Number of Sockets***

The number of sockets per protocol is limited.

- UDP: 8 sockets
- TCP Clients: 8 sockets
- TCP Servers: 4 sockets

# Chapter 2 – General Configuration AT Commands

## IP Stack Handling +WIPCFG

**Description:** The +WIPCFG command is used for performing the following operations:

- Starts TCP/IP stack
- Stops TCP/IP stack
- Configures TCP/IP stack
- Displays version information

**Description Notes:**

- This command can be used even if the SIM card is absent.
- The +WIND indication from which this command is allowed is 3, which provides information about the SIM presence after a software reset and also indicates whether the SIM is inserted or removed. See *Appendix A – GSM/GPRS +WIND AT Command*.

**Syntax:** For <mode>, see Parameter/Defined Values below.

If <mode>=0, 1    **AT+WIPCFG=<mode>**  
 If <mode>=2      **AT+WIPCFG=<mode>,<opt num>,<value>**  
 If <mode>=3      **AT+WIPCFG=<mode>**  
 If <mode>=4      **AT+WIPCFG=<mode>,<action>**

**Read Syntax:**    AT+WIPCFG?    Displays the current <optnum> and <value>

**Test Syntax:**    AT+WIPCFG=?    Displays OK

**Parameters/Defined Values:**

<b>&lt;mode&gt;</b>	<b>Requested Operation</b>
0	stop TCP/IP stack
1	start TCP/IP stack
2	configure TCP/IP stack
3	display TCP/IP application version
4	TCP/IP stack configuration management

<b>&lt;opt num&gt;</b>	<b>Configuration Option Identifier</b>
0	WIP_NET_OPT_IP_TTL Default TTL of outgoing data grams Range: 0-255 (default value: 64)
1	WIP_NET_OPT_IP_TOS Default TOS of outgoing parameters Range: 0-255 (default value: 0)
2	WIP_NET_OPT_IP_FRAG_TIMEO Time to live in seconds of incomplete fragments Range: 1-65535 (default value: 60)
3	WIP_NET_OPT_TCP_MAXINITWIN Number of segments of initial TCP window Range: 0-65535 (default value: 0)
4	WIP_NET_OPT_TCP_MIN_MSS Default MSS of off-link connections Range: 536-1460 (default value: 536)
5	WIP_NET_OPT_DEBUG_PORT Range: 0-3 (default value: 0)

- 6 WIP\_NET\_OPT\_SOCK\_MAX  
Total number of sockets (TCP and UDP)  
Range: 1-172 (default value: 8)
- 7 WIP\_NET\_OPT\_BUF\_MAX  
Total number of network buffers  
Range: 4-42 (default value: 32)
- 8 WIP\_NET\_OPT\_IP\_MULTI\_MAX  
Total number of multicast group  
**Caution:** The option WIP\_NET\_OPT\_IP\_MULTI\_MAX is a read only parameter.
- 9 WIP\_NET\_OPT\_IP\_ROUTE\_MAX  
Size of IP routing table  
Range: 0-2730 (default value: 0)
- 10 WIP\_NET\_OPT\_RSLV\_QUERY\_MAX  
Maximum number of DNS resolver queries  
Range: 1-511 (default value: 4)
- 11 WIP\_NET\_OPT\_RSLV\_CACHE\_MAX  
Size of DNS resolver cache  
Range: 1-292 (default value: 4)

**<action> Requested operation on TCP/IP stack parameter management**

- 0 Configuration storage (when existing) is freed  
1 Stores the configuration parameters

**<value> Value range for different configuration options****<XX.YY.ZZ> WIP soft release version****<A.B> Open AT® OS release version**

**Parameter Storage:** Only one IP stack configuration set can be saved into the FLASH memory.

- AT+WIPCFG=4,1 is used to store the TCP/IP stack configuration parameters into the FLASH memory
- AT+WIPCFG=4,0 is used to free the TCP/IP stack configuration storage

Executing AT+WIPCFG=1 will apply default parameters when existing. Still, it is possible to change option values at run time using AT+WIPCFG=2,<optnum>,<optvalue>.

**Possible Errors:** The possible error message is displayed only if “AT+CMEE=1” is activated else “ERROR” is displayed.

+CMEE AT error code	Description
800	invalid option
801	invalid option value
802	not enough memory left
820	error writing configuration in FLASH memory
821	error freeing configuration in FLASH memory
844	stack already started
850	initialization failed



## Examples:

Command	Responses
AT+WIPCFG=1 Note: Start IP Stack	OK
AT+WIPCFG?	+WIPCFG: 0,64 +WIPCFG: 1,0 +WIPCFG: 2,60 +WIPCFG: 3,0 +WIPCFG: 4,536 +WIPCFG: 5,0 +WIPCFG: 6,8 +WIPCFG: 7,32 +WIPCFG: 8,0 +WIPCFG: 9,0 +WIPCFG: 10,4 +WIPCFG: 11,4 OK
AT+WIPCFG=2,0,10 Note: Configure TTL of the IP Stack	OK
AT+WIPCFG?	+WIPCFG: 0,10 +WIPCFG: 1,0 +WIPCFG: 2,60 +WIPCFG: 3,0 +WIPCFG: 4,536 +WIPCFG: 5,0 +WIPCFG: 6,8 +WIPCFG: 7,32 +WIPCFG: 8,0 +WIPCFG: 9,0 +WIPCFG: 10,4 +WIPCFG: 11,4 OK
AT+WIPCFG=3 Note: Display software version	WIP soft v202 on Open AT OS v312 Mar 26 2007 11:45:46 WIPLib:v2a07 WIPSoft v1a12 OK
AT+WIPCFG=0 Note: Stop the TCP/IP Stack	OK
AT+WIPCFG=4,1 Note: Store IP configuration parameters into FLASH	OK
AT+WIPCFG=4,0 Note: Free IP configuration parameters stored in FLASH	OK

## Bearers Handling +WIPBR

**Description:** The +WIPBR command can be used to:

- Select the bearer
- Start/close the bearer
- Configure different bearer options such as access point name

**Description Notes:**

- The SIM card must be inserted in order to use this command.
- This command can be used even if the PIN 1/CHV 1 is not entered.
- This command can be used even if the PIN 2/CHV 2 is not entered.

**Syntax:**

For <cmdtype>, see Parameters/Defined Values below.

If <cmdtype>=0,1 or 5 **AT+WIPBR=<cmdtype>,<bid>**

If <cmdtype>=2 **AT+WIPBR=<cmdtype>,<bid>,<opt num>,<value>**

If <cmdtype>=3 **AT+WIPBR=<cmdtype>,<bid>,<opt num>**

If <cmdtype>=4 **AT+WIPBR=<cmdtype>,<bid>,<mode>[,<login>,<password>,<caller identity>]**

If <cmdtype>=6 **AT+WIPBR=<cmdtype>,<bid>,<mode>**

**Read Command:** AT+WIPBR? Reads current values.

**Test Command:** AT+WIPBR=? Lists available values.

**Unsolicited Response:** If <mode>=1 **+WIPBR: <bid>,<status>,<local IP @>,<remote IP @>,<DNS1 @>,<DNS2 @>**

**Parameters/**

**Defined Values:**

**<cmd type>**

0	close bearer
1	open bearer
2	set value of different bearer options
3	get value of different bearer options
4	start bearer
5	stop bearer
6	bearer configuration management

**<bid>**

	<b>Bearer Identifier</b>
1	UART1
2	UART2
3	N/A
4	N/A
5	GSM
6	GPRS
11..14	CMUX port over UART1
21..24	CMUX port over UART2

**<opt num>**

	<b>Bearer Option Identifier</b>
0	WIP_BOPT_LOGIN username (string) max: 64 characters
1	WIP_BOPT_PASSWORD password (string) max: 64 characters
2	WIP_BOPT_DIAL_PHONENB phone number (string) max: 32 characters

5	WIP_BOPT_DIAL_RINGCOUNT Number of rings to wait before sending the WIP_BEV_DIAL_CALL event range: 0-65535
6	WIP_BOPT_DIAL_MSNULLMODEM Enable MS-Windows null-modem protocol ("CLIENT"/"SERVER" handshake) range: 0-1
7	WIP_BOPT_PPP_PAP Allow PAP authentication range: 0-1
8	WIP_BOPT_PPP_CHAP Allow CHAP authentication range: 0-1
9	WIP_BOPT_PPP_MSCHAP1 Allow MSCHAPv1 authentication range: 0-1
10	WIP_BOPT_PPP_MSCHAP2 Allow MSCHAPv2 authentication range: 0-1
11	WIP_BOPT_GPRS_APN Address of GGSN (string) max: 96 characters
12	WIP_BOPT_GPRS_CID CID of the PDP context range: 1-4
13	WIP_BOPT_GPRS_HEADERCOMP Enable PDP header compression range: 0-1
14	WIP_BOPT_GPRS_DATACOMP Enable PDP data compression range: 0-1
15	WIP_BOPT_IP_ADDR Local IP address (IP/string)
16	WIP_BOPT_IP_DST_ADDR Destination IP address (IP/string)
17	WIP_BOPT_IP_DNS1 Address of primary DNS server (IP/string)
18	WIP_BOPT_IP_DNS2 Address of secondary DNS server (IP/string)
19	WIP_BOPT_IP_SETDNS Configure DNS resolver when connection is established range: 0-1
20	WIP_BOPT_IP_SETGW Set interface as default gateway when connection is established range: 0-1
<b>&lt;value&gt;:</b>	<b>range of value for different bearer options</b>
<b>&lt;mode&gt;:</b>	<b>mode of operation</b>
	0 client
	1 server
<b>&lt;state&gt;:</b>	<b>current state of the bearer</b>
	0 stopped
	1 started

**<status>:** **result of the connection process**  
 0 successful  
 any other value: to be matched to error code value (e.g. “814” means PPP authentication failure)

**<local IP @\*>:** **local IP address**  
**<remote IP @\*>:** **remote IP address. (first node in internet)**  
**<DNS1 IP @\*>:** **Domain Name Server address**  
**<DNS2 IP @\*>:** **Domain Name Server address**  
**<login>:** **PPP login**  
**<passwd>:** **PPP password**  
**<caller identity>:** **optional ASCII string (type ascii\*).**  
 If not specified, then target will accept all DATA calls (independently of caller identification). If specified, then target will only accept calls from <caller identity> (which are the GSM data call number of the GSM client).

\*IP @ are displayed in alpha numeric dot format. e.g. 192.168.0.1...When no IP address is known, “0.0.0.0” is displayed.

**Caution:** The options WIP\_BOPT\_IP\_DST\_ADDR, WIP\_BOPT\_IP\_DNS1 and WIP\_BOPT\_IP\_DNS2 are “read only” for GPRS/GSM client

### Parameter Storage

Several bearer configuration set can be saved.

Calling twice AT+WIPBR=6,<bid>,1 with the same <bid> will store the last configuration set.

- **AT+WIPBR=6,<bid>,1** is used to store the bearer configuration parameters set associated with the bearer <bid> into the FLASH memory.
- **AT+WIPBR=6,<bid>,0** is used to free the bearer configuration parameters set associated with the bearer <bid>.

Executing **AT+WIPBR=1,<bid>** will open bearer <bid> with default parameters of the bearer when existing.

### Possible Errors

The possible error message is displayed only if “AT+CMEE=1” is activated else “ERROR” is displayed.

<b>+CMEE AT error code</b>	<b>Description</b>
800	invalid option
801	invalid option value
802	not enough memory left
803	already open
804	not available on this platform
807	bearer connection failure: line busy
808	bearer connection failure: no answer
815	bearer connection failure: PPP authentication failed
816	bearer connection failure: PPP IPCP negotiation failed
820	error writing configuration in FLASH memory
821	error freeing configuration in FLASH memory

**Examples**

<b>Command</b>	<b>Responses</b>
AT+WIPBR?	1,0 6,1 OK Note: Bearer UART1 is open but not started bearer GPRS is open and started
AT+WIPBR?	OK Note: No bearer has been opened yet
AT+WIPBR=1,6 Note: Open GPRS bearer	OK
AT+WIPBR=2,6,11,"APN name" Note: Set APN of GPRS bearer	OK
AT+WIPBR=3,6,11 Note: Get APN of GPRS bearer	+WIPBR: 6,11,"APN name" OK
AT+WIPBR=4,6,0 Note: Start GPRS bearer	OK
AT+WIPBR=5,6 Note: Stop GPRS bearer	OK
AT+WIPBR=0,6 Note: Close GPRS bearer	OK
AT+WIPBR=1,5 Note: Open GSM bearer	OK
AT+WIPBR=2,5,0,"login" Note: Set the login for GSM bearer	OK
AT+WIPBR=2,5,1,"password" Note: Set the password for GSM bearer	OK
AT+WIPBR=2,5,2,"phonenumber" Note: Set the phone number for GSM bearer	OK
AT+WIPBR=2,5,15,"1.1.1.1" Note: Set the local IP address for GSM bearer	OK
AT+WIPBR=2,5,16,"2.2.2.2" Note: Set the destination IP address for GSM bearer	OK
AT+WIPBR=3,5,15 Note: Read the local IP address for GSM bearer	+WIPBR: 5,15,"0.0.0.0" OK Note: Local IP address is not set as GSM bearer and is still not connected
AT+WIPBR=3,5,16 Note: Read the destination IP address for GSM bearer	+WIPBR: 5,16,"0.0.0.0" OK Note: Destination IP address is not set as GSM bearer and is still not connected
AT+WIPBR=4,5,0 Note: Start the GSM bearer as a client	OK
AT+WIPBR=3,5,15 Note: Read the local IP for GSM bearer	+WIPBR: 5,15,"1.1.1.1" OK
AT+WIPBR=3,5,16 Note: Read the destination IP for GSM bearer	+WIPBR: 5,16,"2.2.2.2" OK
AT+WIPBR=5,5 Note: Stop the GSM bearer	OK
AT+WIPBR=0,5 Note: Close the GSM bearer	OK

**Notes:****Starting a Bearer**

The mandatory parameters to start a bearer in

- **server mode:** <cmdtype>, <bid>, <mode>, <login> and <password>
- **client mode:** <cmdtype>, <bid> and <mode>

Depending on the mode and the bearer type, additional parameters are required or forbidden:

Bid	Mode	Other Parameters
1,3,11,14,21,24	0	None
1,3,11,14,21,24	1	<PPP login>, <PPP password>
5	0	None
5	1	<login>,<password>[,<caller identity>]
6	0	None

Starting bearer as a server requires additional parameters as mentioned in the above table.

- For PPP server, only parameters <login> and <password> are required. They will be compared with remote PPP client login and password.
- For GSM server, <login> and <password> will be used for PPP over GSM establishment (same behavior as described for PPP server).

The <caller identity> is an optional ASCII string (type ASCII\*). If not specified, then target will accept all DATA calls (independently of caller identification). If specified, then target will only accept calls from <caller identity> (which is the GSM data call number of the GSM client).

Opening bearer only consists in associating the IP protocol stack with the specified bearer. The corresponding bearer setup has to be done through the adequate already existing AT commands (please refer to +WMFM commands for UART1 and UART2, +CMUX command for CMUX virtual ports and GSM/GPRS AT commands).

Several bearers can be opened at the same time but only one bearer can be started at a time. If both DNS1 and DNS2 are displayed as "0.0.0.0" in the unsolicited message when bearer is opened in server mode, it means that connecting to a remote IP host through an URL will fail.

The options WIP\_BOPT\_DIAL\_REDIALCOUNT and WIP\_BOPT\_DIAL\_REDIALDELAY will not be implemented through AT commands. Nevertheless, for future compatibility reason, Opt num 3 and 4 are kept as reserved.

For GSM bearer, the options WIP\_BOPT\_IP\_ADDR and WIP\_BOPT\_IP\_DST\_ADDR will display valid addresses only when the bearer is started and connected; otherwise, it will display an address "0.0.0.0".

## Chapter 3 – IP Protocol Services

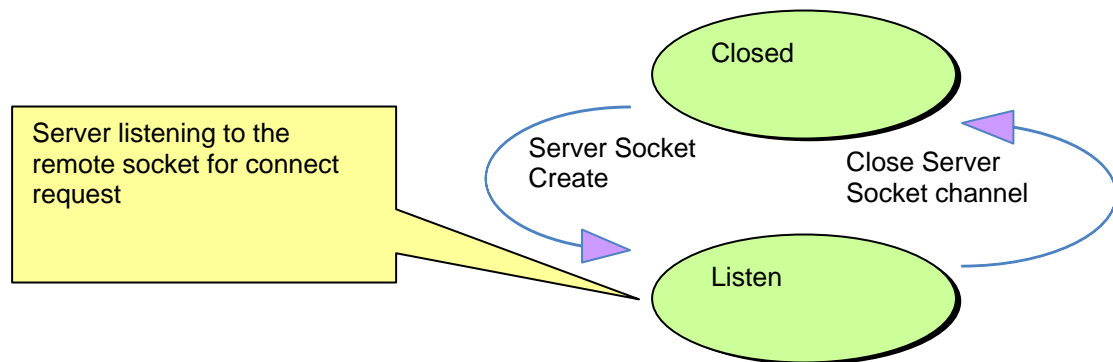
### Service Creation +WIPCREATE

#### Description:

The +WIPCREATE command is used to create UDP, TCP client and TCP server sockets associated with the specified index and FTP service. Only one FTP session at a time is available.

If a local port is specified while creating a socket, the created socket will be assigned to this port; if not, a port will be assigned dynamically by WIP application. If peer IP and peer port is specified, the created socket will be connected to the specified IP and port.

TCP server cannot be used to transfer data. To transfer data, it creates a local TCP client socket. This process of creating local socket is referred as “spawning”. When a server socket is created using, socket passively listens on a specified port for incoming connections. The below mentioned diagram shows different states managed for TCP server.



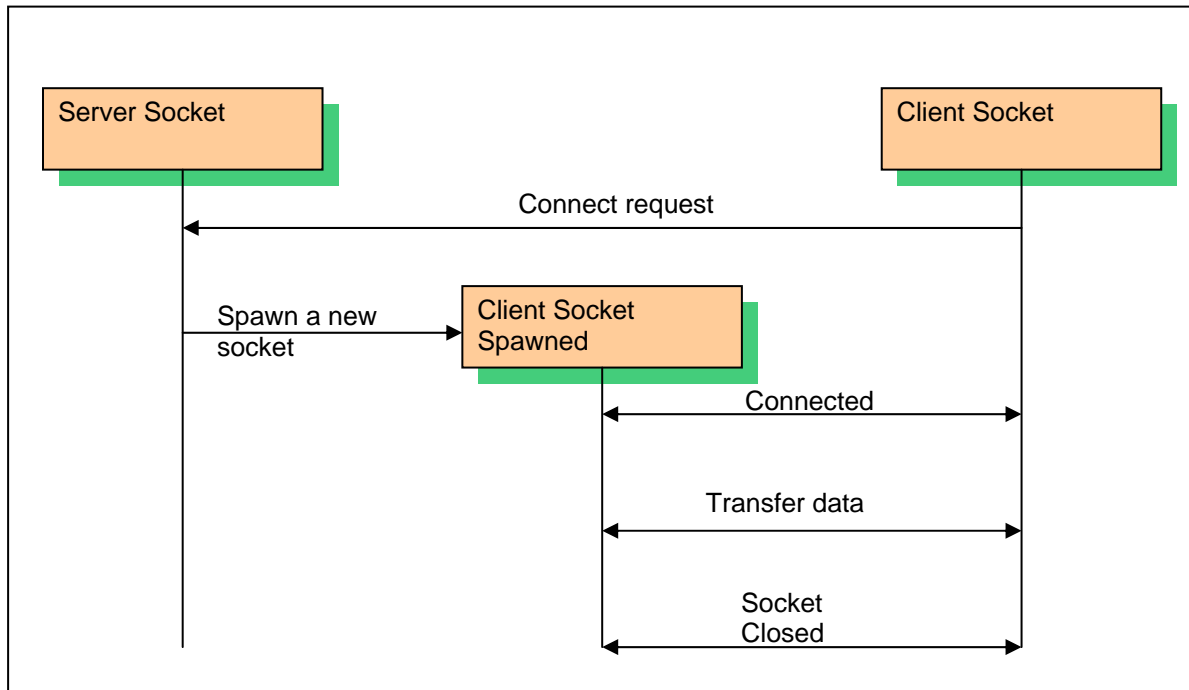
On reception of a connection request from a remote client socket, a server socket does the following:

- Spawns a new socket (client) to connect to the remote socket
- Data transfer is done between the spawned socket and the remote socket
- Server socket remains in the listening mode and is ready to accept the request from other clients

#### Description Notes:

- A SIM card must be inserted in order to use this command.
- The PIN 1/CHV 1 code must be entered to use this command.
- The PIN 2/CHV 2 does not have to be entered to use this command.
- The +WIND general indication command value from which +WIPCREATE is allowed is 4. This value (4) indicates that the product is ready to process AT commands (except phonebooks, AOC, SMS), but is still in emergency mode. See *Appendix A – GSM/GPRS +WIND AT Command*.

This diagram shows how to establish a connection.



**Syntax:** For a definition of <mode>, see Parameter/Defined Values on the next page.

If <mode>=1:

**AT+WIPCREATE=<mode>,<communication index>,[<local port>] [,<peer IP>,<peer port>]**

If <mode>=2

**AT+WIPCREATE=<mode>,<communication index>,<peer IP>,<peer port>**

If <mode>=3

**AT+WIPCREATE=<mode>,<server index>,<local port>,<from idx>,<to idx>**

If <mode>=4

**AT+WIPCREATE=<mode>,<index>,<server>[,<peer\_port>],<username>,<password> [,<account>]**

**Read Command:** AT+WIPCREATE? Displays current values.

**Test Command:** AT+WIPCREATE=? Displays available values.

**Unsolicited Response:** If <mode>=1 or 2  
**+WIPREADY: <mode>,<communication index>**

If <mode>=3

**+WIPACCEPT: <server index>,<communication idx>**



**Parameters/Defined Values:**

<b>&lt;mode&gt;:</b>	<b>Specifies Type of Socket</b>
1	UDP
2	TCP Client
3	TCP Server
4	FTP
<b>&lt;index&gt;:</b>	<b>TCP/UDP/FTP Connection Identifier</b>
<b>&lt;local port&gt;:</b>	<b>Local TCP/UDP Port</b>
<b>&lt;peer IP&gt;:</b>	<b>Peer IP Address; A String Between Quotes</b> Indicating an address either in numeric form (e.g. "85.12.133.10") or as a DNS entry (e.g. "www.wavecom.com")
<b>&lt;peer port&gt;:</b>	<b>Peer TCP/UDP Port in Socket Service or the Server Port in FTP Service</b> Range: 1-65535 (Default Value: 21)
<b>&lt;from idx&gt;:</b>	<b>Minimum Index for Spawned TCP Sockets</b> Range: 1-8
<b>&lt;server index&gt;:</b>	<b>TCP Server Socket Identifier</b>
<b>&lt;to idx&gt;:</b>	<b>Maximum Index for Spawned TCP Sockets</b> Range: 1-8
<b>&lt;communication index&gt;:</b>	<b>Indexes Reserved for Spawned Sockets</b> It cannot be used by other sockets even if the spawned sockets are not created yet.
<b>&lt;server&gt;:</b>	<b>Address of the FTP Server</b> It can either be a 32 bit number in dotted-decimal notation ("xxx.xxx.xxx.xxx") or an alpha numeric string format for hostname.
<b>&lt;user name&gt;:</b>	<b>Login of the User</b> String
<b>&lt;password&gt;:</b>	<b>Password of the User</b> String
<b>&lt;account&gt;:</b>	<b>Account Information of the User</b> This is required by some ftp server during authentication phases. String
<b>Parameter Storage:</b>	None

**Possible Errors:**

<b>+CMEE" AT error code</b>	<b>Description</b>
3	operation not allowed
800	invalid option
803	operation not allowed in the current WIP stack state
830	bad index
832	bad port number
834	not implemented
836	memory allocation error
837	bad protocol
839	error during channel creation
840	FTP session is already active
842	destination host unreachable (whether host unreachable, Network unreachable, response timeout)

## Examples:

Command	Responses
<b>AT+WIPCREATE=1,1,80</b> Note: Create the UDP socket on local port 80 with communication index = 1 ⇔ Wireless CPU <sup>®</sup> acts as an UDP server awaiting for incoming datagram on local port 80	OK Note: An unsolicited event +WIPREADY: 1,1 will be received once the UDP socket is ready for usage
<b>AT+WIPCREATE=1,1,"www.wavecom.com",80</b> Note: Create the UDP socket on arbitrary free local port with peer IP and peer port 80 with communication index = 1 ⇔ Wireless CPU <sup>®</sup> acts as a UDP client that can send datagram towards the remote entity	OK Note: An unsolicited event +WIPREADY: 1,1 will be received once the UDP socket is ready for usage
<b>AT+WIPCREATE=1,1,80,"www.wavecom.com",80</b> Note: Create the UDP socket on local port 80 with peer IP and peer port 80 with communication index = 1 ⇔ Wireless CPU <sup>®</sup> acts as a UDP client and an UDP server : it can send datagram towards the remote entity and receiving datagram on the specified local port.	OK Note: An unsolicited event +WIPREADY: 1,1 will be received once the UDP socket is ready for usage
<b>AT+WIPCREATE=3,1,80,5,9</b> Note: Create the TCP server on port 80 with server index=1 ⇔ Wireless CPU <sup>®</sup> acts as a TCP server : it will from now on spawn TCP client socket from communication index 5 to 9	OK Note: An unsolicited event +WIPACCEPT: 1,5 will be received once the TCP server is ready for usage
<b>AT+WIPCREATE=2,1,"IP ADDR",80</b> Note: Create the TCP client on port 80 with index=1 ⇔ Wireless CPU <sup>®</sup> acts as a TCP client : it can from now on communicate with the remote specified entity through communication index 1	OK Note: An unsolicited event +WIPREADY: 2,1 will be received once the TCP client is ready for usage
<b>AT+WIPCREATE=4,1,"ftp.wavecom.com","admin","123456"</b> Note: Create a FTP session ⇔ towards the remote specified FTP server. Communication index to be used then is 1	OK

## Notes:

- The +WIPCREATE command causes the connection and authentication to the FTP server. If several file uploads and retrievals are required to/from the same server, a single connection with +WIPCREATE is needed. Then, each file operation will be done (one +WIPFILE command per operation), and the FTP connection will be released with +WIPCLOSE.
- SIM card is required only if FTP session is established through GSM or GPRS. An FTP session upon an UART will work without a SIM card.

## Closing a Service +WIPCLOSE

**Description:** The +WIPCLOSE command is used to close a socket or FTP session. When one serial port (UART or CMUX DLCI) is used to map a socket for read/write operations, an [ETX] character can also be used to close the socket.

An unsolicited event is generated, when socket or FTP session is closed.

**Description Notes:**

- The SIM card must be inserted in order to use this command.
- The PIN 1/ CHV 1 code must be entered in order to use this command.
- The PIN 2/CHV 2 code does not have to be entered in order to use this command.
- The +WIND general indication command value from which +WIPCREATE is allowed is 4. This value (4) indicates that the product is ready to process AT commands (except phonebooks, AOC, SMS), but is still in emergency mode. See *Appendix A – GSM/GPRS +WIND AT Command*.

**Syntax:** AT+WIPCLOSE=<protocol>,<idx>

**Read Command:** AT+WIPCLOSE?

**Test Command:** AT+WIPCLOSE=?

**Unsolicited Response:**+WIPPEERCLOSE: <protocol>,<idx>

**Parameters/Defined Values:**

<b>&lt;protocol&gt;:</b>	<b>protocol type</b>
1	UDP
2	TCP client
3	TCP server
4	FTP
<b>&lt;idx&gt;:</b>	<b>socket identifier</b>

**Parameter Storage:** None

**Possible Errors:**

<b>“+CMEE” AT error code</b>	<b>Description</b>
802	not enough memory
803	operation not allowed in the current WIP stack state
830	bad index
831	bad state
834	not implemented
837	bad protocol

**Closing a Service +WIPCLOSE Continued****Examples:**

<b>Command</b>	<b>Responses</b>
<b>AT+WIPCLOSE=1,1</b> Note: Close UDP socket with communication index 1	OK Note: An unsolicited event +WIPPEERCLOSE: 1,1 is received once the UDP socket is closed
<b>AT+WIPCLOSE=2,1</b> Note: Close TCP client with communication index 1	OK Note: An unsolicited event +WIPPEERCLOSE: 2,1 is received once the TCP client is closed
<b>AT+WIPCLOSE=3,1</b> Note: Close TCP server with communication index 1	OK Note: An unsolicited event +WIPPEERCLOSE: 3,1 is received once the TCP server is closed
<b>AT+WIPCLOSE=4,1</b> Note: Close FTP session with index 1	OK Note: An unsolicited event +WIPPEERCLOSE: 4,1 is received once the FTP session is closed

**Notes:**

Sockets will be closed only on issuing +WIPCLOSE command and the closure of the socket is indicated by +WIPPEERCLOSE:

- <protocol>, <idx> unsolicited response.

After issuing +WIPCLOSE command, no more data can be sent and received over the socket.

## Service Option Handling +WISOPT

**Description:** The +WISOPT command is used to read and/or to configure different parameters on sockets and FTP service.

**Description Notes:**

- The SIM card must be inserted in order to use this command
- The PIN 1/CHV 1 code must be entered in order to use this command.
- The PIN 2/CHV 2 code does not have to be entered in order to use this command.
- The +WIND general indication command value from which +WISOCREATE is allowed is 4. This value (4) indicates that the product is ready to process AT commands (except phonebooks, AOC, SMS), but is still in emergency mode. See *Appendix A – GSM/GPRS +WIND AT Command*.

**Syntax:** For a definition of <action>, see Parameter/Defined Values below.

If <action>=1 **AT+WISOPT=<protocol>,<idx>,<action>,<optnum>**

If <action>=2 **AT+WISOPT=<protocol>,<idx>,<action>,<optnum>,<optval>**

**Read Command:** AT+WISOPT? Displays the current values.

**Test Command:** AT+WISOPT=? Displays available values.

**Parameters/Defined Values:**

**<protocol>:** protocol type

1	UDP
2	TCP client
3	TCP server
4	FTP

**<idx>:** socket identifier

**<action>:** requested operation

1	read the value of an option
2	write the value of an option

**<optnum>:** option that can be read/written

**<optval>:** value of an option

**Parameter Storage:** None

**Possible Errors:**

+CME AT error code	Description
800	invalid option
801	invalid option value
803	operation not allowed in the current WIP stack state
830	bad index
834	not implemented
835	option not supported
837	bad protocol
850	unknown reason

**Examples:**

Command	Responses
AT+WISOPT=2,1,2,8,20 Note: Set TTL for TCP client	OK
AT+WISOPT=2,1,1,8 Note: Get TTL for TCP client	+WISOPT: 2,1,8,20 OK
AT+WISOPT=3,1,2,9,10 Note: Set TOS for TCP server	OK
AT+WISOPT=3,1,1,9 Note: Get TOS for TCP server	+WISOPT: 2,1,9,10 OK
AT+WISOPT=1,1,1,1 Note: Get peer port for UDP	+WISOPT: 2,1,1,80 OK
AT+WISOPT=4,1,2,40,1 Note: Set data representation type for FTP	OK
AT+WISOPT=4,1,1,40 Note: Get data representation type for FTP	+WISOPT: 4,1,1,1 OK

**Notes:**

The options numbers, description, read/write permission, on UDP, TCP client and TCP server sockets, is summed up in the following table.

Opt Num	Value Format	Meaning	UDP	TCP Client	TCP Server
0	0-65535	WIP_COPT_PORT	R	R	R
1	0-65535	WIP_COPT_PEER_PORT	R	R	-
2	string	WIP_COPT_PEER_STRADDR	R	R	-
3	0-1	WIP_COPT_BOUND	R	-	-
4	0-5839	WIP_COPT_SND_LOWAT	-	RW	RW
5	0-5839	WIP_COPT_RCV_LOWAT	-	RW	RW
6	0-65535	WIP_COPT_NREAD	R	R	-
7	0-1	WIP_COPT_NODELAY	-	RW	RW
8	0-255	WIP_COPT_TTL	RW	RW	RW
9	0-255	WIP_COPT_TOS	RW	RW	RW

The options that can be applied to FTP channel are:

Opt Num	Value Format	Value Type	Meaning
40	0-1	Boolean	data representation type. 0: ASCII 1: binary default: 0
41	0-1	Boolean	FTP mode. 0: active 1: passive default: 1

# Chapter 4 – Data Exchange for Protocol Services

The section deals with the data exchange for the services over TCP/IP. All the commands required for the data exchange through different services are mentioned in succeeding pages.

## File Exchange +WIPFILE

**Description:** The +WIPFILE command define the “file system” services that allow sending a block of data through standard TCP/IP protocols. This command is for file transfer/reception.

**Notes:**

- The SIM card must be inserted in order to use this command
- The PIN 1/ CHV 1 code must be entered in order to use this command.
- The PIN 2/CHV 2 code does not have to be entered in order to use this command.
- The +WIND general indication command value from which +WIPCREATE is allowed is 4. This value (4) indicates that the product is ready to process AT commands (except phonebooks, AOC, SMS), but is still in emergency mode. See *Appendix A – GSM/GPRS +WIND AT Command*.

**Syntax:** **AT+WIPFILE=<protocol>,<index>,<mode>,<filename>**

**Read command:** **AT+WIPFILE?** Displays the current values.

**Test Command:** **AT+WIPFILE=?** Displays available values.

**Parameters/Defined Values:**

<b>&lt;protocol&gt;:</b>	<b>protocol type</b>
4	FTP
<b>&lt;idx&gt;:</b>	<b>channel identifier</b>
<b>&lt;mode&gt;:</b>	<b>file transfer mode</b>
1	File retrieval: Wireless CPU <sup>®</sup> switches to data mode and prints the content of the file on UART. The end of the file is marked by [ETX] character. After this has been sent, the UART switches back to AT mode.
2	File transfer: This command switches the UART in data mode and accepts a stream of data terminated by [ETX] character.
<b>&lt;filename&gt;:</b>	<b>specifies the name of the file to send or retrieve.</b>
	The maximum file length is limited to 128 characters. The actual filename, including path name has to be used.

**Parameter Storage:** None

**Possible Errors:**

“+CMEE” AT error code	Description
800	invalid option
803	operation not allowed in the current WIP stack state
830	bad index
831	bad state
834	not implemented
836	memory allocation error
837	bad protocol
839	error during channel creation

**Examples:**

Command	Responses
AT+WIPFILE=4,1,1,"data.bin" Note: Retrieve the data for the given filename with index 1	CONNECT <data received terminated by [ETX] character> OK
AT+WIPFILE=4,1,2,"report.log" Note: Send data to the given filename	CONNECT <data terminated by [ETX] character> OK

**Note:**

The [ETX] character is considered as an end of data. Hence, in case [ETX] character needs to be transmitted, it should be preceded by [DLE] character.



## Socket Data Exchange +WIPDATA

**Description:** The +WIPDATA command is used to read/write from/to a socket. On successful execution of the command, the UART switches to data mode. The UART can be switched back to AT mode by sending “+++” with 1 second guard time before and after the sequence. If data is not read using +WIPDATA command, further data will be delayed.

An unsolicited event is received when there is a data to read on socket.

Data can be sent on the sockets using two modes

- continuous mode
- continuous transparent mode

### Description Notes:

- The SIM card must be inserted in order to use this command
- The PIN 1/ CHV 1 code must be entered in order to use this command.
- The PIN 2/CHV 2 code does not have to be entered in order to use this command.
- The +WIND general indication command value from which +WIPCREATE is allowed is 4. This value (4) indicates that the product is ready to process AT commands (except phonebooks, AOC, SMS), but is still in emergency mode. See *Appendix A – GSM/GPRS +WIND AT Command*.

### Continuous Mode: TCP Sockets in Continuous Mode

In continuous mode, an [ETX] character is considered as an end of data. When an [ETX] character is sent on the mapped UART, the TCP socket is shutdown and the peer side is informed of this shutdown with the indication “[CR][LF]SHUTDOWN[CR][LF]” on the mapped UART.

In case an [EXT]/[DLE] character needs to be transmitted as data, it should be preceded by a [DLE] character. Similarly, [EXT]/[DLE] characters received by the TXP/IP stack from the Internet are sent to the host through the serial port preceded by a [DLE] character.

To close sockets, switch the UART to AT command mode and use the +WIPCLOSE command.

### UDP Sockets in Continuous Mode

UDP is a connectionless protocol, and, hence, there is no way to detect or cause a shutdown. However, an [ETX] character is used to mark the boundaries of datagrams.

All data written on an UDP socket is collected till an [ETX] character is encountered or the maximum size of the datagram<sup>1</sup> is reached and will be sent as a single datagram. Similarly when reading data, all data will be read till an [ETX] character is encountered which indicates the end of the datagram.

<sup>1</sup> Maximum size of an UDP datagram has been fixed to 5840 Bytes. This limit is an arbitrary one. Nevertheless, note that smaller the datagram is the surer it will reach the aimed destination. Note that UDP is not a reliable transport layer.

In case an [ETX]/[DLE] character needs to be transmitted, it should be preceded by [DLE] character similar to TCP socket.

When the UART leaves DATA mode, either because of “+++” escape sequence or because of an AT+WIPDATA=1, index, 0 on another UART, the currently unsent data are sent as a single datagram.

**[ETX] Escaping Mechanism:**

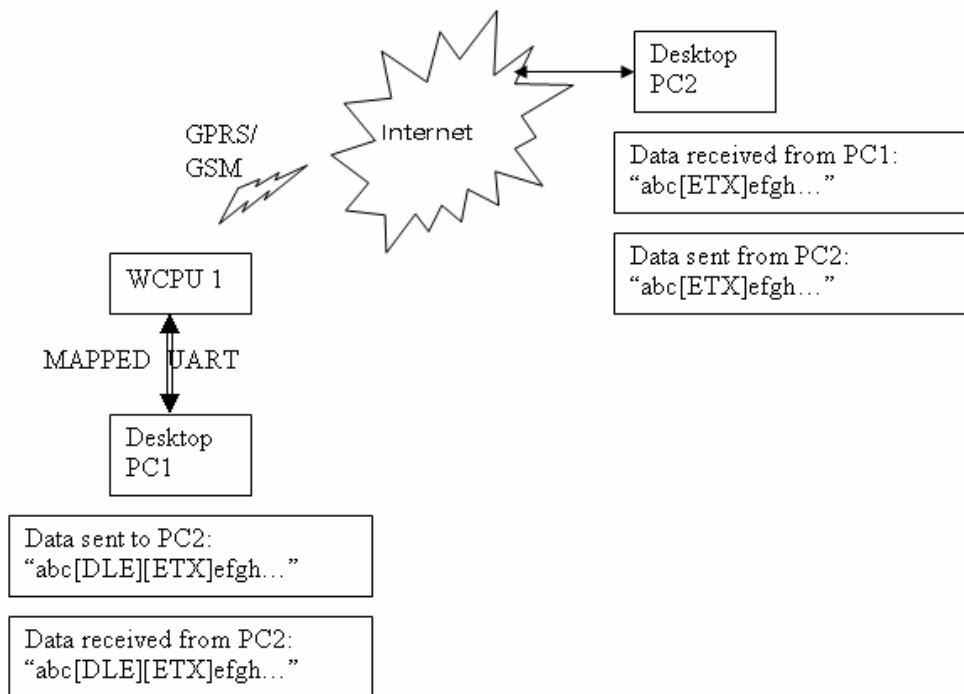
The following schematic explains how [ETX] characters, which have a special meaning in WIPSoft, are handled on the Wavecom Wireless CPU®.

**On transmitting side:** When [ETX] characters are not escaped (use case: Desktop PC1 sends data to the Wireless CPU® and the data contains a non-escaped [ETX] (<=> no [DLE][ETX] sequence), then the [ETX] is not transmitted, but an action is done on the Wavecom Wireless CPU® regarding the concerned socket:

- **UDP socket:** A non-escaped [ETX] character marks the boundary of the current datagram to be sent. The datagram is immediately sent, and the [ETX] is not sent to the desktop PC2.
- **TCP socket:** A non-escaped [ETX] character causes a TCP shutdown operation on the transmitting direction; the peer is informed that the Wavecom Wireless CPU® will not send any more data on that socket. Usually, the peer will shutdown the other way (downlink), and this will result in a “peer close event” on the socket.

**On receiving side:** When [ETX] characters are not escaped (use case: Wavecom Wireless CPU® sends data to the Desktop PC1 and the data contains a non-escaped [ETX] (<=> no [DLE][ETX] sequence), then [ETX] means that a special “IP” event occurred on the Wavecom Wireless CPU® regarding the concerned socket:

- **UDP socket:** A non-escaped [ETX] signals the boundary of the current received datagram.
- **TCP socket:** A non-escaped [ETX] signals that the peer TCP connected to the TCP unit shutdowns the downlink way. Desktop PC1 should then close the uplink socket to totally terminate the TCP “session”.



Protocol	Mapped UART	IP Network (active socket)
UDP	Data containing [DLE][ETX] sequence.	Data containing [ETX].
UDP	[ETX] alone.	Mark the boundary of the UDP Datagram received/to be transmitted.
TCP	Data containing [DLE][ETX] sequence.	Data containing [ETX].
TCP	[ETX] alone.	Causes/signals a shutdown operation on TCP socket.

Note that the behavior is symmetrical; i.e., it applies to both the transmitting and receiving side of the mapped UART.

**[DLE] Escaping Mechanism**

A [DLE] character will be sent as data only when it is preceded by another [DLE] character. A single [DLE] character which is not preceded by a [DLE] character will not be transmitted.

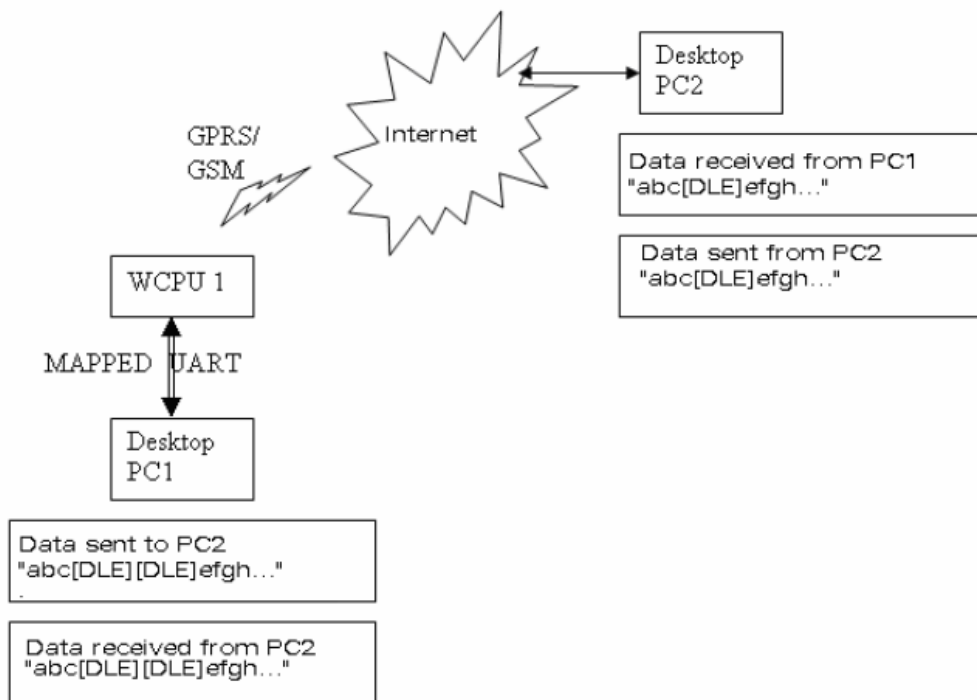
The following schematic explains how [DLE] characters, which have a special meaning in WIPSoft, are handled on the Wavecom Wireless CPU®.

**On the transmitting side:** When a [DLE] character is not escaped (use case: Desktop PC1 sends data to the Wavecom Wireless CPU® and the data contains a non-escaped [DLE] (<=> no [DLE] [DLE] sequence), then the [DLE] is not transmitted.

**On the transmitting side:** When a [DLE] is escaped (use case: Desktop PC1 sends data to the Wavecom Wireless CPU® and the data contains an escaped [DLE] (<=> [DLE] [DLE] sequence), then the [DLE] data is transmitted.

**On the receiving side:** (Use case: When the Desktop PC2 sends data to the Wavecom Wireless CPU® and the data contains a no escaped [DLE], the data sent from Wireless CPU® to Desktop PC1 will contain an escaped [DLE] preceding the [DLE] character (Desktop PC1 receives a [DLE][DLE] character from the Wireless CPU).

This scenario is the same for both TCP and UDP sockets.



Protocol	Mapped UART	IP Network (active socket)
UDP	Data containing [DLE][DLE] sequence.	Data containing [DLE].
UDP	[DLE] alone.	A single [DLE] is ignored.
TCP	Data containing [DLE][DLE] sequence.	Data containing [DLE].
TCP	[DLE] alone.	A single [DLE] is ignored.

**Continuous Transparent Mode**

**TCP Sockets in Continuous Transparent Mode**

In this mode there is no special meaning associated for [DLE]/[ETX] characters. They are considered as normal data and all the data will be transmitted on the mapped UART.

UDP sockets do not support this mode. Attempting to map an UART in this mode will result in a "+CME ERROR: 837".

### Leaving Continuous /Continuous Transparent Mode

The UART can be switched back to AT mode:

- By sending “+++” with 1 second guard time before and after the sequence
- By sending an AT+WIPDATA=<proto>,<idx>,0 on another UART in AT mode

When the UART leaves data mode either because of “+++” escape sequence or because of an unmapping done on another UART, the currently unsent data are sent as a single datagram.

### Resetting TCP Sockets

A TCP socket is reset when the connection is aborted due to an error on the socket. When the socket is reset, an [EXT] character is sent on the mapped UART to indicate the end of communication. The mapped UART switches to AT mode and "CME ERROR:843" is displayed on the UART.

**Syntax:** AT+WIPDATA=<protocol>,<idx>,<mode>

**Read Command:** AT+WIPDATA? Displays the current values.

**Test Command:** AT+WIPDATA=? Displays the available values.

**Unsolicited Response:** If <protocol>=1 **+WIPDATA: <protocol>,<idx>,<datagram size>,<peer IP>,<peer port>**

If <protocol>=2 **+WIPDATA: <protocol>,<idx>,<number of readable bytes>**

**Caution:** The value returned by <number of readable bytes> indicates that there is some TCP data ready to be read but number of bytes returned might not be reliable.

### Parameters/Defined Values:

**<protocol>:** **socket type**  
 1 UDP  
 2 TCP client  
**<idx>:** **socket identifier**  
**<mode>:** **mode of operation**  
 0 unmap: switch the UART (mapped to continuous mode) to AT mode.  
 1 continuous: switch the UART to data mode.  
 2 continuous transparent: switch the UART to data mode. In this mode,[DLE]/[ETX] characters are considered as normal data and not special characters.

**Parameter Storage:** None

### Possible Errors:

"+CMEE" AT error code	Description
851	bad state
837	bad protocol
843	connection reset by peer

**Examples:**

Command	Responses
AT+WIPDATA=2,5,1 Note; TCP Client with index 5 can send/read data in continuous mode	CONNECT <read/write data> +++ OK Note; +++ sequence causes the UART to switch to AT mode
AT+WIPDATA=1,5,1 Note; UDP with index 5 can send/read data in continuous mode	CONNECT <read/write data> +++ OK Note; +++ sequence causes the UART to switch to AT mode
AT+WIPDATA=1,5,1 Note; UDP with index 5 can send/read data in continuous mode	CONNECT <read/write data> <ETX> OK Note; [ETX] character indicates end of data
AT+WIPDATA=2,5,2 Note: TCP with index 5 can send/read data in continuous transparent mode	CONNECT <read/write data> +++ OK Note; +++ sequence causes the UART to switch to AT mode

**Notes:**

**Continuous Mode (Non Transparent) for a TCP Mapped Socket**

If the [ETX] character is sent from the peer, it is considered as an end of data transfer. After sending an [ETX] character, the socket will be shut down and the peer will be informed of this shutdown by a "[CR][LF]SHUTDOWN[CR][LF]" indication on its mapped UART. The UART does not switch to AT mode. This indicates that no more data can be sent from the host socket, but it can receive data.

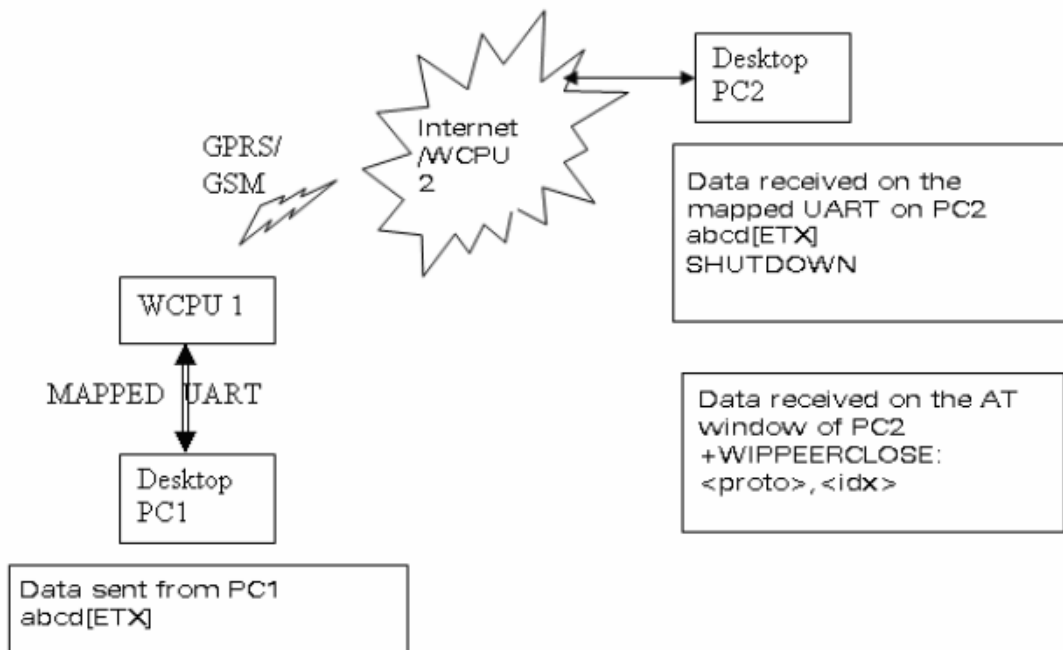
**Shutdown Procedure for a TCP Socket**

In the following schematic, a TCP socket is connected. On the transmitting side, data and an [EXT] character are sent (use case: Desktop PC1 is a Wireless CPU which sends data to PC2 which is either a PC or a Wireless CPU), the data is received on PC2 and an [EXT] character shutdowns the socket on the transmitting side and displays a message "[CR][LF]SHUTDOWN[CR][LF]" on the mapped UART of PC2.

When PC2 is switched back to AT mode, the "+WIPPEERCLOSE:<protocol>,<idx>" indication is received indicating that no more data can be sent by PC1 but can read data sent from PC2.

There are different indications received for shutdown and reset for a TCP socket. When a TCP socket is reset, an [EXT] character is sent on the mapped UART to indicate the end of communication. The mapped UART switches to AT mode and "+CME ERROR: 843" is displayed on the UART. The reset and shutdown can, therefore, be distinguished by the indications received on the UART.

### Shutdown Procedure for a TCP Socket



### Mapping/Unmapping of a Mapped UDP and TCP Socket

When a TCP socket is unmapped and still active, it is possible to map it again in another mode that is different from the previous one without closing the TCP socket.

The UART switches back to AT mode due to “+++” with a 1 second guard time before and after the sequence or by sending an `AT+WIPDATA=<proto>,<index>,0` on another UART in AT mode. This applies to both UDP and TCP protocols.

When +++ is issued, the Wireless CPU<sup>®</sup> switches from DATA mode to AT mode. If the ATO command is used to switch the Wireless CPU<sup>®</sup> back to DATA mode, then one of two things can happen:

- +CME ERROR:3 will be received when GPRS bearer is used
- no response is received when GSM bearer is used

To switch the Wireless CPU<sup>®</sup> back to DATA mode, `AT+WIPDATA=x,x,x` should be used instead of ATO. After executing `AT+WIPDATA=x,x,x` command, "CONNECT" will be received to indicate that the Wireless CPU<sup>®</sup> is switched back to DATA mode.

## Chapter 5 – Ping Services

### PING Command +WIPPING

**Description:** The +WIPPING command is used to configure different PING parameters and to send PING requests. An unsolicited response is displayed each time a “PING” echo event is received or a timeout expires.

**Description Notes:**

- The SIM card must be inserted in order to use this command
- The PIN 1/ CHV 1 code must be entered in order to use this command.
- The PIN 2/CHV 2 code does not have to be entered in order to use this command.
- The +WIND general indication command value from which +WIPCREATE is allowed is 4. This value (4) indicates that the product is ready to process AT commands (except phonebooks, AOC, SMS), but is still in emergency mode. See *Appendix A – GSM/GPRS +WIND AT Command*.

**Syntax:** `AT+WIPPING=<host>,[<repeat>,<interval>,<timeout>,<nwrite>,<tll>]]]]`

**Read Command:** `AT+WIPPING?` Displays the current values.

**Test Command:** `AT+WIPPING=?` Displays the available values.

**Unsolicited response:** `+WIPPING:<timeout_expired>,<packet_idx>,<response_time>`

**Parameters/Defined Values:**

**<host>:** **host name or IP address**  
string

**<repeat>:** **number of packets to send**  
range: 1-65535 (default value:1)

**<interval>:** **number of milliseconds between packets**  
range: 1-65535 (default value:2000)

**<timeout>:** **number of milliseconds before a packet is considered lost**  
range: 1-65535 (default value:2000)

**<tll>:** **IP packet Time to Live.**  
default value is set by WIP\_NET\_OPT\_IP\_TTL +WIPCFG option  
range: 0-255

**<nwrite>:** **size of packets**  
range: 1-1500 (default value:64)

**<timeout\_expired>:** **PING result**  
0: PING response received before <timeout>  
1: <timeout> expired before the response was received

**<packet\_idx>:** **packet index in the sequence**

**<response\_time>:** **PING response time in millisecond**

**Parameter Storage:** None

**Possible Errors:**

“+CMEE” AT error code	Description
800	invalid option
801	invalid option value
819	error on ping channel

**PING Command +WIPPING Continued****Examples:**

<b>Command</b>	<b>Responses</b>
AT+WIPPING="www.wavecom.com" Note: Ping "www.wavecom.com"	OK +WIPPING: 1,0,0 Note: Ping "www.wavecom.com failed : timeout expired
AT+WIPPING="192.168.0.1" Note: Ping "192.168.0.1"	OK +WIPPING: 0,0,224 Note: Ping "192.168.0.1 succeeded. Ping response received in 224 ms
AT+WIPPING="192.168.0.1",2,2000,1000 Note: Send 2 successive ping requests to "192.168.0.1". Each Ping is every 2000 ms, timeout is set to 2000 ms (if ping responses time is more than 1000 ms then timeout expires)	OK +WIPPING: 0,0,880 +WIPPING: 1,1,xxxx Note: Ping "192.168.0.1 succeeded. First Ping response received in 880 ms. Second one was not received before specified timeout (1000 ms) ⇔ timeout expired



## Chapter 6 – Examples of Application

### TCP Socket

#### TCP Server Socket Using GPRS Bearer

```

AT+WIPCFG=1 //start IP stack
OK
AT+WIPBR=1,6 //open GPRS bearer
OK
AT+WIPBR=2,6,11,"APN name" //set APN name of GPRS bearer
OK
AT+WIPBR=2,6,0,"user name" //set user name (<login>)
OK
AT+WIPBR=2,6,1,"passwd" //set password (<password>)
OK
AT+WIPBR=4,6,0 //start GPRS bearer
OK
AT+WIPCREATE=3,1,80,5,9 // create the server on port 80, idx=1
OK //TCP Server can spawn up to 5 TCP clients.
//Assigned indexes are from 5 to 9.
+WIPACCEPT: 1,5 //unsolicited: the server accepted
//connection; resulting TCP client
//on index 5
AT+WIPDATA=2,5,1 //exchange data on socket index 5
CONNECT
... //read, write
+++ //switch to AT mode
OK
AT+WIPCLOSE=2,5 //close the TCP client socket index 5
OK

```

### TCP Server Socket Using GSM Bearer

```
AT+WIPCFG=1 //start IP stack
OK
AT+WIPBR=1,5 //open GSM bearer
OK
AT+WIPBR=2,5,2,"Phone number" //set phone number for GSM bearer
OK
AT+WIPBR=2,5,0,"user name" //set user name
OK
AT+WIPBR=2,5,1,"passwd" //set password
OK
AT+WIPBR=4,5,0 //start GSM bearer
OK
AT+WIPCREATE=3,1,80,5,9 //create the server on port 80, idx=1
OK //TCP Server can spawn up to 5 TCP clients.
//Assigned indexes are from 5 to 9
+WIPACCEPT: 1,5 //unsolicited: the server accepted
//connection; resulting TCP client
//index 5
AT+WIPDATA=2,5,1 //exchange data on socket idx 5
CONNECT
... //read, write
+++ //switch to AT mode
OK
AT+WIPCLOSE=2,5 //close the TCP client socket index 5
OK
```

## TCP Client Socket

### TCP Client Socket Using GPRS Bearer

```

AT+WIPCFG=1 //start IP stack
OK
AT+WIPBR=1,6 //open GPRS bearer
OK
AT+WIPBR=2,6,11,"APN name" //set APN name of GPRS bearer
OK
AT+WIPBR=2,6,0,"user name" //set user name
OK
AT+WIPBR=2,6,1,"passwd" //set password
OK
AT+WIPBR=4,6,0 //start GPRS bearer
OK
AT+WIPCREATE=2,1,"ip addr",80 //create the TCP client on port 80, idx=1
OK
+WIPREADY: 2,1 //unsolicited: the server accepted
//connection; resulting TCP client index 5

AT+WIPDATA=2,1,1 //exchange data on socket idx 1:
CONNECT
... //read, write
+++ //switch to AT mode
OK
AT+WIPCLOSE=2,1 //close the TCP client socket index 1
OK

```

**TCP Client Socket Using GSM Bearer**

```
AT+WIPCFG=1 //start IP stack
OK
AT+WIPBR=1,5 //open GSM bearer
OK
AT+WIPBR=2,5,2,"Phone number" //set phone number for GSM bearer
OK
AT+WIPBR=2,5,0,"user name" //set user name
OK
AT+WIPBR=2,5,1,"passwd" //set password
OK
AT+WIPBR=4,5,0 //start GSM bearer
OK
AT+WIPCREATE=2,1,"ip addr",80 //create the TCP client on port 80, idx=1
OK
+WIPREADY: 2,1 //unsolicited: the server accepted
//connection; resulting TCP client index 5
AT+WIPDATA=2,1,1 //exchange data on socket idx 1
CONNECT
... //read, write
+++ //switch to AT mode
OK
AT+WIPCLOSE=2,1 //close the TCP client socket index 1
OK
```

## UDP Socket

```
AT+WIPCFG=1 //start IP stack
OK
AT+WIPBR=1,6 //open GPRS bearer
OK
AT+WIPBR=2,6,11,"APN name" //set APN name of GPRS bearer
OK
AT+WIPBR=2,6,0,"user name" //set user name
OK
AT+WIPBR=2,6,1,"passwd" //set password
OK
AT+WIPBR=4,6,0 //start GPRS bearer
OK
AT+WIPCREATE=1,1,80,"www.wavecom.com",80 //start UDP socket
OK
WIPREADY: 1,1
AT+WIPDATA=1,1,1 //exchange data on socket idx 1:
CONNECT
... //read, write
+++ //switch to AT mode
OK
AT+WIPCLOSE=1,1 //close the UDP socket index 1
OK
```

## PING

```
AT+WIPCFG=1 //start IP stack
OK
AT+WIPBR=1,6 //open GPRS bearer
OK
AT+WIPBR=2,6,11,"APN name" //set APN name of GPRS bearer
OK
AT+WIPBR=2,6,0,"user name" //set user name
OK
AT+WIPBR=2,6,1,"passwd" //set password
OK
AT+WIPBR=4,6,0 //start GPRS bearer
OK
AT+WIPPING="192.168.0.1" //start PING session
OK
+WIPPING:0,0,224
```

## FTP

```
AT+WIPCFG=1 //start IP stack
OK
AT+WIPBR=1,6 //open GPRS bearer
OK
AT+WIPBR=2,6,11,"APN name" //set APN name of GPRS bearer
OK
AT+WIPBR=2,6,0,"user name" //set user name
OK
AT+WIPBR=2,6,1,"passwd" //set password
OK
AT+WIPBR=4,6,0 //start GPRS bearer
OK
AT+WIPCREATE=4,1,"FTP server",21,"username","passwd" //create FTP session
OK
AT+WIPFILE=4,1,2,"/filename.txt" //upload file "filename.txt"
CONNECT
<data>
[ETX]
OK
AT+WIPFILE=4,1,1,"/filename.txt" //download file "filename.txt"
CONNECT
<data>
[ETX]
OK
```

## Chapter 7 – Error Codes

“+CMEE” AT error code	Description
800	invalid option
801	invalid option value
802	not enough memory
803	operation not allowed in the current WIP stack state
804	device already open
805	network interface not available
806	operation not allowed on the considered bearer
807	bearer connection failure : line busy
808	bearer connection failure : no answer
809	bearer connection failure : no carrier
810	bearer connection failure : no SIM card present
811	bearer connection failure : SIM not ready (no pin code entered, ...)
812	bearer connection failure : GPRS network failure
813	bearer connection failure : PPP LCP negotiation failed
814	bearer connection failure : PPP authentication failed
815	bearer connection failure : PPP IPCP negotiation failed
816	bearer connection failure : PPP peer terminates session
817	bearer connection failure : PPP peer does not answer to echo request
818	incoming call refused
819	error on Ping channel
820	error writing configuration in FLASH memory
821	error reading configuration in FLASH memory
822-829	reserved for future use
830	bad index
831	bad state
832	bad port number
833	bad port state
834	not implemented
835	option not supported
836	memory allocation error
837	bad protocol
838	no more free socket
839	error during channel creation
840	FTP session is already active
841	peer closed
842	destination host unreachable ( whether host unreachable, Network unreachable, response timeout)
843	connection reset by peer
844	stack already in use
845-849	reserved for future use
850	unknown reason
851	bad state



# Appendix A – GSM/GPRS +WIND Command

## General Indications +WIND

**Description:** This command provides a general mechanism to send unsolicited non-standardized indications to the application. The indicators are:

- Indication of a physical change on the SIM detect pin from the connector (meaning SIM inserted, SIM removed)
- Indication during mobile originated call setup that the calling party is ringing.
- Indication of the availability of the product to receive AT commands after boot.
- NITZ indication (Network Information and Time Zone)

For each of these indications, a “bit flow” has to be indicated.

**Values:**

**<IndLevel>**

<b>0</b>	No unsolicited “+WIND: <IndNb>” will occur. <b>Default.</b>
<b>1 (bit-0)</b>	Hardware SIM Insert/Remove indications or SIM presence after software reset.
<b>2 (bit-1)</b>	Calling party alert indication.
<b>4 (bit-2)</b>	Product is ready to process AT commands (except phonebooks, AOC, SMS), but still in emergency mode.
<b>8 (bit-3)</b>	The product is ready to process all AT commands at the end of init or after swapping to ADN in case of FDN configuration
<b>16 (bit-4)</b>	A new call identifier has been created (after an ATD command, +CCWA indication)
<b>32 (bit-5)</b>	An active, held or waiting call has been released by network or other party
<b>64 (bit-6)</b>	Network service available indication
<b>128 (bit-7)</b>	Network lost indication
<b>256 (bit-8)</b>	Audio ON indication
<b>512 (bit-9)</b>	SIM phonebooks reload status
<b>1024 (bit-10)</b>	SIM phonebooks checksum indication
<b>2048 (bit-11)</b>	Interruption indication (only if FTR_INT is activated)
<b>4096 (bit-12)</b>	Hardware rack open/closed indication
<b>8192 (bit-13)</b>	NITZ indication
<b>16384 (bit-14)</b>	SMS service ready indication

Combination (addition of the values) is used to allow more than one indication flow:  **$0 \leq \text{IndLevel} \leq 32767$**

- To activate a specific WIND indication, <IndLevel> must have a value described above.  
AT+WIND=16384 only activates SMS service indication.
- To activate several WIND indications, <IndLevel> must have a value just before the last indication required.  
AT+WIND=32767 all unsolicited indications.

**<event>**

<b>0</b>	The SIM presence pin has been detected as “SIM removed”
<b>1</b>	The SIM presence pin has been detected as “SIM inserted”
<b>2</b>	Calling party is alerting
<b>3</b>	Product is ready to process AT commands (except phonebooks, AOC, SMS), at init or after AT+CFUN=1
<b>4</b>	Product is ready to process all AT commands, end of phonebook init or swap (FDN to ADN)
<b>5</b>	Call <idx> has been created (after ATD or +CCWA...)

- 6 Call <idx> has been released, after a NO CARRIER, a +CSSU: 5 indication, or after the release of a call waiting.
- 7: The network service is available for an emergency call.
- 8 The network is lost.
- 9 Audio ON.
- 10 Show reload status of each SIM phonebook after init phase (after Power-ON or SIM insertion).
- 11 Show the checksum of SIM phonebooks after loading.
- 12 An interruption has occurred.
- 13 The rack has been detected as Closed.
- 14 The rack has been detected as Open.
- 15 The modem received a NITZ information message from the network.
- 16 SMS and SMS CB services are ready.

**Event 10:**

<phonebook>: SIM phonebook

"SM"  
 "FD"  
 "ON"  
 "SN"  
 "EN"

<status>:

- 0 Not Reloaded from SIM (no change since last init or SIM remove)
- 1 Reloaded from SIM to internal memory (at least one entry has changed)

**Event 11:**

<checksum>: 128-bit "fingerprint" of the phonebook.

**Note:** If the service of the phonebook is not loaded or not present, the checksum is not displayed and two commas without checksum are displayed (,,).

**Event 15:**

<Full name>: String. Updated long name for current network.

<Short name>: String. updated short name for current network.

<Local time zone>: Signed integer. Time Zone indicates the difference, expressed in quarters of an hour, between the local time and GMT.

<Universal time and local time zone>: String, Universal Time and Time Zone, in format "yy/MM/dd,hh:mm:ss±zzz"

(Year/Month/Day,Hour:Min:Seconds± Time Zone).

The Time Zone indicates the difference, expressed in quarters of an hour, between the local time and GMT.

<LSA Identity>: Hexa string. LSA identity of the current cell in hexa format (3 bytes).

<Daylight Saving Time>: Integer (0-2). When the LTZ is compensated for DST (Daylight Saving Time or summertime), the serving PLMN shall provide a DST parameter to indicate it. The adjustment for DST can be + 1h or +2h.

**Note:** For the NITZ indication, all the fields indicated here are optional. That is why there is an index related to each of the following:

- 1: Full name for network
- 2: Short name for network
- 3: Local time zone
- 4: Universal time and local time zone
- 5: LSA identity
- 6: Network Daylight Saving Time

**Command syntax:** AT+WIND= <IndLevel >

Command	Possible Responses
AT+WIND?	+WIND: 0 OK
AT+WIND=255	OK
Note: The SIM has been removed.	+WIND: 0 Note :The SIM presence pin has been detected as "SIM removed"
Note: The SIM has been inserted.	+WIND: 1 Note :The SIM presence pin has been detected as "SIM inserted"
Note: The network service is available for an emergency call	+WIND: 7
Note: The initialization has been completed	+WIND: 4
Note: The modem received a NITZ information message	+WIND: 15,1,"Cingular Extended",2,"Cingular",3,"+08",4,"03/14/27,16:59:48+08",5,"123456",6,"2"

#### Additional Notes:

- The AT+WIND? command is supported and indicates the <allowed bit flows>.
- AT+WIND settings are automatically stored in non volatile memory (EEPROM). This means the &W command does not need to be used and the selected flows are always activated after boot.
- Default value is 0: no flow activated, no indication.
- AT+WIND=? gives the possible value range (0-4095)
- The unsolicited response will then be:  
+WIND: <event> [,<idx>]  
<idx>: Call identifier, defined in +CLCC command.
- Or for event 10:  
+WIND: <event>,<phonebook>,<status>,...,<phonebook>,<status>
- Or for event 11:  
+WIND: <event>,[["<checksum of SM>"],["<checksum of FD>"],["<checksum of ON>"],["<checksum of SN>"],["<checksum of EN>"],["<checksum of LD>"]]
- Or for event 15 (NITZ indication):  
+WIND: <event>[,1,"Full name>"][,2,"<Short name>"][,3,"<Local time zone>"][,4,"<Universal time and local time zone>"][,5,"<LSA identity>"][,6,"<Daylight Saving Time>"]]

# Index

<b>+</b>		
+WIND General Indications .....	41	
+WIPBR Bearers Handling .....	10	
+WIPCFG IP Stack Handling .....	7	
+WIPCLOSE Closing a Service .....	19	
+WIPCREATE Service Creation .....	15	
+WIPDATA Socket Data Exchange .....	25	
+WIPFILE File Exchange .....	23	
+WIPOPT Service Option Handling .....	21	
+WIPPING Ping Command .....	31	
 <b>A</b>		
Acronyms and Abbreviations .....	4	
 <b>C</b>		
Command Line .....	5	
Commands, Specific		
General Indications +WIND .....	41	
 <b>D</b>		
Data Exchange for Protocol Services		
File Exchange +WIPFILE .....	23	
Socket Data Exchange +WIPDATA .....	25	
 <b>E</b>		
Error Codes .....	40	
Examples of Application		
FTP .....	39	
PING .....	38	
TCP Client Socket Using GPRS Bearer .....	35	
TCP Client Socket Using GSM Bearer .....	36	
 TCP Server Socket Using GPRS Bearer .....		33
TCP Server Socket Using GSM Bearer .....		34
UDP Socket .....		37
 <b>G</b>		
General AT Commands		
Bearers Handling +WIPBR .....	10	
IP Stack Handling +WIPCFG .....	7	
General Indications +WIND .....	41	
 <b>I</b>		
IP Protocol Services		
Closing a Service +WIPCLOSE .....	19	
Service Creation +WIPCREATE .....	15	
Service Option Handling +WIPOPT .....	21	
 <b>M</b>		
Multiple UARTs .....	6	
Multiplexing .....	6	
 <b>N</b>		
Number of Sockets .....	6	
 <b>P</b>		
Ping Services		
Ping Command WIPPING .....	31	
Possible Protocols .....	6	
 <b>S</b>		
Socket Identification .....	6	