



WIP AT Commands User Guide (WIPSoft V2.00)

Revision: 002
Date: September 2006



wavecom[®]
Make it wireless

WIP AT COMMANDS USER GUIDE (WIPsoft V2.00)

Reference: WM_DEV_OAT_UGD_024

Revision: 002

Date: 25 September 2006

Preliminary

Trademarks

®, 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.

Preliminary

Copyright

This manual is copyrighted by Wavecom with all rights reserved. No part of this manual may be reproduced in any form without the prior written permission of Wavecom.

No patent liability is assumed with respect to the use of the information contained herein.

Preliminary

Overview

The aim of this document is to provide Wavecom customers with a full description of the Wavecom AT commands associated with the Wavecom IP feature.

Preliminary

Document History

Level	Date	History of the evolution	Writer
001	25 August 2006	Creation	Wavecom
002	25 September 2006	Preliminary	Wavecom

Preliminary

Contents

1	INTRODUCTION	10
1.1	Related Documents	10
1.2	Abbreviations and Definitions	11
1.2.1	Abbreviations	11
1.2.2	Glossary	12
1.3	Logos	13
1.4	AT Commands Presentation Rules	14
2	AT COMMAND SYNTAX	15
2.1	Command Line	15
2.2	Information responses and result codes	16
3	INTRODUCTION	17
3.1	Sockets identification	18
3.1.1	Possible Protocols	18
3.1.2	Number of sockets	18
4	IP STACK HANDLING +WIPCFG	19
4.1	Description	19
4.2	Syntax	20
4.3	Parameters and Defined Values	22
4.4	Parameter Storage	25
4.5	Possible errors	26
4.6	Examples	28
5	BEARERS HANDLING +WIPBR	31
5.1	Description	31
5.2	Syntax	32
5.3	Parameters and Defined Values	35
5.4	Parameter Storage	40

5.5	Possible errors.....	41
5.6	Examples	44
6	IP PROTOCOL SERVICES.....	45
6.1	Service creation +WIPCREATE.....	45
6.1.1	Description	45
6.1.2	Syntax.....	45
6.1.3	Parameters and Defined Values.....	47
6.1.4	Parameter Storage	48
6.1.5	Possible errors.....	48
6.1.6	Examples	49
6.2	Closing a Service +WIPCLOSE.....	51
6.2.1	Description	51
6.2.2	Syntax.....	51
6.2.3	Parameters and Defined Values.....	52
6.2.4	Parameter Storage	52
6.2.5	Possible Errors	52
6.2.6	Examples	53
6.3	Service Option handling +WIPOPT.....	54
6.3.1	Description	54
6.3.2	Syntax.....	54
6.3.3	Parameters and Defined Values.....	55
6.3.4	Parameter Storage	56
6.3.5	Possible Errors	56
6.3.6	Examples	56
7	DATA EXCHANGE FOR PROTOCOL SERVICES.....	58
7.1	File Exchange +WIPFILE	58
7.1.1	Description	58
7.1.2	Syntax.....	58
7.1.3	Parameters and Defined Values.....	59

7.1.4	Parameter Storage	60
7.1.5	Possible Errors	60
7.1.6	Examples	60
7.2	Socket data exchange +WIPDATA	61
7.2.1	Description	61
7.2.2	Syntax	62
7.2.3	Parameters and Defined Values	63
7.2.4	Parameter Storage	64
7.2.5	Possible errors	64
7.2.6	Examples	64
8	PING SERVICES +WIPPING	66
8.1	Description	66
8.2	Syntax	67
8.3	Parameters and Defined Values	68
8.4	Parameter Storage	69
8.5	Possible Errors	70
8.6	Examples	71
9	EXAMPLES OF APPLICATION	72
9.1	TCP Socket	72
9.1.1	TCP Server Socket	72
9.1.2	TCP Client Socket	76
9.2	UDP Socket	80
9.3	PING	82
9.4	FTP	83
10	ERROR CODES	85
11	NOTES	88
11.1	Bearer Handling	88
11.2	IP Protocol Services	89
11.3	Data Exchange for Protocol Services	91

1 Introduction

1.1 Related Documents

None

Preliminary

1.2 Abbreviations and Definitions

1.2.1 Abbreviations

Abbreviation	Description
APN	Access Point Name
AT	ATtention
CHV	Card Holder Verification
CID	Context Identifier
CPU	Central Processing Unit
DNS	Domain Name System
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
IN/OUT/GLB	In, out or Global. See glossary
IP	Internet Protocol
M	Mandatory
MS	Mobile Station
NA	Not Applicable
NU	Not Used
O	Optional
PDP	Packet Data Protocol

Abbreviation	Description
PIN	Personal Identity Number
PPP	Point-to-Point Protocol
SIM	Subscriber Information Module
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver Transmitter
UDP	User Data Protocol
WIP	Wavecom Internet Protocol

1.2.2 Glossary

In / out / Glb: used in function parameters:

“In” if the parameter is given to the function,

“Out” if the parameter is a part of the waited results of the function, “Glb” (for Global) if the parameter is used for both.

1.3 Logos



This picture indicates the +WIND indication from which the AT command is allowed. X values can be: 1, 3, 4, 16.



This picture indicates that a SIM card must be inserted to support the AT command.



This picture indicates that an AT command is supported even if the SIM card is absent.



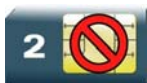
This picture indicates that the PIN 1 /CHV 1 code must be entered to support the AT command.



This picture indicates that an AT command is supported even if the PIN 1 /CHV 1 code is not entered.



This picture indicates that the PIN 2 /CHV 2 code must be entered to support the AT command.



This picture indicates that an AT command is supported even if the PIN 2/CHV 2 code is not entered.

1.4 AT Commands Presentation Rules

The AT commands are presented in the present documentation as follows:

- a "Description" section provides general information on the AT command (or response) behavior
- a "Syntax" section describes the command and response syntaxes and all parameters description
- a "Parameters and Defined Values" section describes all parameters and values
- a "Parameter Storage" presents the command used to store the parameter value and/or the command used to restore the parameter default value
- an "Examples" section presents the real use of the described command
- a "Note" section can also be included indicating some remarks about the command use

Figures are provided wherever necessary.

Preliminary

2 AT Command Syntax

This section describes the AT command format and the default value for their parameters.

2.1 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+WIPcmd is executed without <Param2> the default value of <param2> is used.

Preliminary

2.2 Information responses and result codes

Responses start and end with <CR><LF> characters.

- if the command syntax is incorrect, an <CR><LF>ERROR<CR><LF> message is returned from the Open AT® Firmware.
- if command syntax is correct but issued with incorrect parameters, the response <CR><LF>+CME ERROR: <Err><CR><LF> is returned with an appropriate error code.
- if the command is processed successfully, <CR><LF>OK<CR><LF> response string is returned.

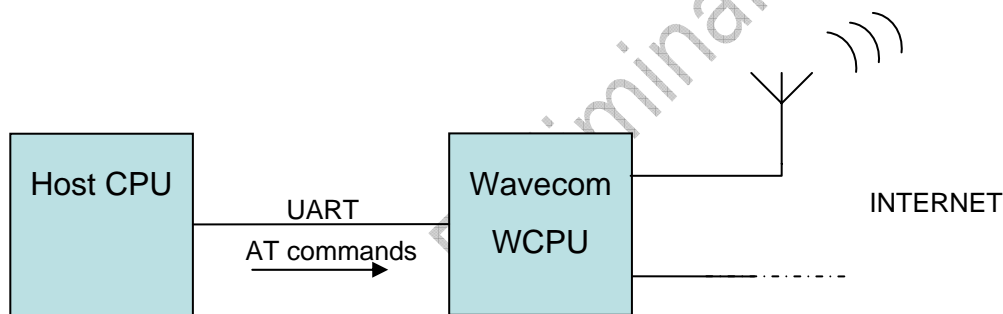
Preliminary

3 Introduction

The wipSoft is an Open AT® application that implements the TCP/IP protocols using custom AT commands. This Open AT® application operates in co-operative mode and must be downloaded to the Wavecom Wireless CPU. The commands are sent from an external application and the corresponding responses are sent back from the Wavecom Wireless CPU to the external application. The 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
- the rest of the Internet / Intranet



Multiplexing: Several sockets can be operating at once. The AT+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 events 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, it is a forbidden to map a single socket on several UARTs simultaneously.

3.1 Sockets 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.

3.1.1 Possible Protocols

The possible protocols are,

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

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

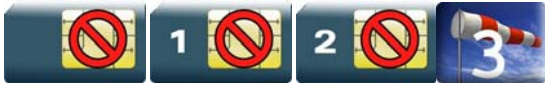
E.g. Both 1,7 and 2,7 are valid identifiers simultaneously; the former identifies a UDP socket and the later, a TCP connected socket.

3.1.2 Number of sockets

The number of sockets per protocol is limited.

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

4 IP Stack Handling +WIPCFG



4.1 Description

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

- start TCP/IP stack
- stop TCP/IP stack
- configuring TCP/IP stack
- displaying version information

Preliminary

4.2 Syntax

- For <mode> = 0,1

Action Command

AT+WIPCFG=<mode>

OK

- For <mode> = 3

Action Command

AT+WIPCFG=<mode>

WIP soft vXX.YY.ZZ on Open AT OS vA.B"

- For <mode> = 2

Action Command

AT+WIPCFG=<mode>,<opt num>,<value>

OK

- For <mode> = 4

Action Command

AT+WIPCFG=<mode>,<action>

OK

Read Command

AT+WIPCFG?

<CR><LF>

+WIPCFG: <optnum>,<value>

[+WIPCFG: <optnum>,<value>[...]]

OK

Test Command

AT+WIPCFG=?

OK

Preliminary

4.3 Parameters and Defined Values

<mode>:	requested operation
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
<opt num>:	configuration option identifier
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: 1-65535 (default value: 0)
4	WIP_NET_OPT_TCP_MIN_MSS - Default MSS of off-link connections range: 0-65535 (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: 0-172 (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 range: NI (default value: NI)
	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

Preliminary

4.4 Parameter Storage

Only one IP stack configuration set can be saved into the FLASH memory.

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

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

Preliminary

4.5 Possible errors

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

- For <mode> = 1

Error	Description
+CME ERROR: 802	Not enough memory left
+CME ERROR: 805	Initialization failed

- For <mode> = 2

Error	Description
+CME ERROR: 800	Invalid option
+CME ERROR: 801	Invalid option value
+CME ERROR: 805	Initialization failed

- For <mode> = 4

Error	Description
+CME ERROR: 800	Invalid option
+CME ERROR: 801	Invalid option value
+CME ERROR: 820	Error writing configuration in FLASH memory
+CME ERROR: 821	Error freeing configuration in FLASH memory

- For AT+WIPCFG?

Error	Description
+CME ERROR: 805	Initialization failed

Preliminary

<p>AT+WIPCFG?</p> <p><i>Note: Read command</i></p>	<pre>+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,NI +WIPCFG: 9,0 +WIPCFG: 10,4 +WIPCFG: 11,4 OK</pre>
<p>AT+WIPCFG=3</p> <p><i>Note: Display software version</i></p>	<pre>WIP soft v201 on Open AT OS v312 OK</pre>
<p>AT+WIPCFG=0</p> <p><i>Note: stop the TCP/IP Stack</i></p>	<pre>OK</pre>
<p>AT+WIPCFG=4,1</p> <p><i>Note: Store IP configuration parameters into FLASH</i></p>	<pre>OK</pre>

<p>AT+WIPCFG=4,0</p> <p><i>Note: Free IP configuration parameters stored in FLASH</i></p>	<p>OK</p>
--	-----------

Preliminary

5 Bearers Handling +WIPBR



5.1 Description

The +WIPBR command can be used to

- select the bearer
- start/close the bearer
- configure different bearer options such as access point name

Preliminary

5.2 Syntax

- For <CmdType> = 0, 1 or 5

Action Command

```
AT+WIPBR=<CmdType>,<bid>
```

```
OK
```

- For <CmdType> = 2

Action Command

```
AT+WIPBR=<CmdType>,<bid>,<opt num>,<value>
```

```
OK
```

- For <CmdType> = 3

Action Command

```
AT+WIPBR=<CmdType>,<bid>,<opt num>
```

```
OK
```


- For <CmdType> = 4

Action Command

```
AT+WIPBR=<CmdType>,<bid>,<mode>[,<login>,<password>,[<caller  
identity>]]
```

OK

- For <CmdType> = 6

Action Command

```
AT+WIPBR=<CmdType>,<bid>,<mode>
```

OK

Read Command

```
AT+WIPBR?
```

```
<bid>,<state>
```

```
[<bid>,<state>[...]]
```

OK

Test Command

```
AT+WIPBR=?
```

TBD

- For <mode> = 1

Unsolicited response

```
+WIPBR: <bid>,<status>,<local IP @>,<remote IP @>,<DNS1 @>,<DNS2 @>
```

Preliminary

5.3 Parameters and Defined Values

<Cmd Type>:		type of command
	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>:		bearer Identifier
	1	UART1
	2	UART2
	3	USB
	4	N.A.
	5	GSM
	6	GPRS
	11..14	CMUX port over UART1
	21..24	CMUX port over UART2

<opt num>:	bearer option identifier
0	WIP_BOPT_PPP_LOGIN - Username (string)
1	WIP_BOPT_PPP_PASSWORD - Password (string)
2	WIP_BOPT_DIAL_PHONENB - Phone number (string)
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 range: string

	12	WIP_BOPT_GPRS_CID - Cid of the PDP context range: 0-255
	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
<value>:		range of value for different bearer options
<mode>:		mode of operation
	0	client

	1	server
<state>		current state of the bearer
	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>		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).

* 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.

<other params> depends on the mode and the bearer type.

Bid	Mode	other params
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 behaviour as described for PPP server).
 - <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).

5.4 Parameter Storage

Several bearer configuration set can be saved. (Note : calling twice AT+WIPBR=6,<bid>,1 with the same <bid> will store the last configuration set).

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

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

Preliminary

5.5 Possible errors

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

- For <CmdType> = 0

Error	Description
+CME ERROR: 800	Invalid option
+CME ERROR: 803	Already open

- For <CmdType> = 1

Error	Description
+CME ERROR: 800	Invalid option
+CME ERROR: 802	Not enough memory left
+CME ERROR: 803	Already open
+CME ERROR: 804	Not available on this platform

- For <CmdType> = 2

Error	Description
+CME ERROR: 800	Invalid option
+CME ERROR: 801	Invalid option value

- For <CmdType> = 3

Error	Description
+CME ERROR: 800	Invalid option

- For <CmdType> = 4, <mode> = 1

Error	Description
+CME ERROR: 800	Invalid option
+CME ERROR: 801	Invalid option value

- For <CmdType> = 4, <mode> = 0

Error	Description
+CME ERROR: 807	Bearer connection failure : line busy
+CME ERROR: 808	Bearer connection failure : no answer
+CME ERROR: 815	Bearer connection failure : PPP authentication failed
+CME ERROR: 816	Bearer connection failure : PPP IPCP negotiation failed

- For <CmdType> = 5

Error	Description
+CME ERROR: 800	Invalid option
+CME ERROR: 801	Invalid option value

- For <CmdType> = 6

Error	Description
+CME ERROR: 800	Invalid option
+CME ERROR: 801	Invalid option value
+CME ERROR: 820	Error writing configuration in FLASH memory
+CME ERROR: 820	Error freeing configuration in FLASH memory

Preliminary

5.6 Examples

Command	Responses
AT+WIPBR? <i>Note: Read command</i>	1,0 6,1 OK <i>Note: bearer UART1 is open but not started bearer GPRS is open and started</i>
AT+WIPBR? <i>Note: Read command</i>	OK <i>Note: No bearer has been opened yet</i>
AT+WIPBR=1,6 <i>Note: Open GPRS bearer</i>	OK
AT+WIPBR=2,6,11,"APN name" <i>Note: Set APN of GPRS bearer</i>	OK
AT+WIPBR=3,6,11 <i>Note: Get APN of GPRS bearer</i>	+WIPBR: 6,11,"APN name" OK
AT+WIPBR=4,6 <i>Note: Start GPRS bearer</i>	OK
AT+WIPBR=5,6 <i>Note: Stop GPRS bearer</i>	OK
AT+WIPBR=0,6 <i>Note: Close GPRS bearer</i>	OK

6 IP Protocol Services

6.1 Service creation +WIPCREATE



6.1.1 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.

6.1.2 Syntax

- For <mode> = 1

Action Command

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

OK

- For <mode> = 2

Action Command

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

OK

- For <mode> = 3

Action Command

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

OK

- For <mode> = 4

Action Command

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

OK

Read Command

AT+WIPCREATE?

TBD

Test Command

AT+WIPCREATE=?

TBD

- For <mode> = 1 or 2

Unsolicited response

+WIPREADY: <mode>, <communication index>

- For <mode> = 3

Unsolicited response

+WIPACCEPT: <mode>, <server idx>, <communication idx>

6.1.3 Parameters and Defined Values

<mode>:	specifies type of socket
1	UDP
2	TCP client
3	TCP server
4	FTP
<index>:	socket or FTP connection identifier
<local port>:	local TCP/UDP port
<peer IP>	peer IP address; a string between quotes, indicating an address either in numeric form (e.g. "85.12.133.10"), or as a DNS entry (e.g. "www.wavecom.com")
<peer port>	peer TCP/UDP port in socket service or the server port in

	FTP service with default value 21. range : 1-65535
<from idx>	minimum index for spawned TCP sockets
<to idx>	maximum index for spawned TCP sockets
<communication idx>	indexes reserved for spawned sockets; cannot be used by other sockets even if the spawned sockets are not created yet
<server>	address of the server. Can be either a 32 bit number in dotted-decimal notation ("xxx.xxx.xxx.xxx") or an alpha numeric string format for hostname
<login>	login of the user
<password>	password of the user
<account>	may be required by some FTP server during authentication Phases

6.1.4 Parameter Storage

None

6.1.5 Possible errors

Error	Description
+CME ERROR: 3	Operation not allowed

6.1.6 Examples

Command	Responses
AT+WIPCREATE=1,1,80 <i>Note: Create the UDP socket on local port 80 with communication index = 1</i>	OK <i>Note: An unsolicited event +WIPREADY: 1,1 will be received once the UDP socket is ready for usage</i>
AT+WIPCREATE=1,1,"www.google.co.in",80 <i>Note: Create the UDP socket on arbitrary free local port with peer IP and peer port 80 with communication index = 1</i>	OK <i>Note: An unsolicited event +WIPREADY: 1,1 will be received once the UDP socket is ready for usage</i>
AT+WIPCREATE=1,1,80,"www.google.co.in",80 <i>Note: Create the UDP socket on local port 80 with peer IP and peer port 80 with communication index = 1</i>	OK <i>Note: An unsolicited event +WIPREADY: 1,1 will be received once the UDP socket is ready for usage</i>
AT+WIPCREATE=3,1,80,5,9 <i>Note: Create the TCP server on port 80 with server index=1</i>	OK <i>Note: An unsolicited event +WIPACCEPT: 3,1,5 will be received once the TCP server is ready for usage</i>
AT+WIPCREATE=2,1,"IP ADDR",80 <i>Note: Create the TCP client on port 80 with index=1</i>	OK <i>Note: An unsolicited event +WIPREADY: 2,1 will be received once the TCP client is ready for usage</i>
AT+WIPCREATE=4,1,"ftp.wavecom.com", "admin", "123456" <i>Note: Create FTP session</i>	OK
AT+WIPCREATE=? <i>Note: Test command</i>	TBD

<p>AT+WIPCREATE?</p> <p><i>Note: Read command</i></p>	<p>TBD</p>
--	------------

Preliminary

6.2 Closing a Service +WIPCLOSE



6.2.1 Description

The +WIPCLOSE command is used to close a socket or FTP session. When UART is in data mode, [ETX] character can also be used to close the socket.

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

6.2.2 Syntax

Action command

```
AT+WIPCLOSE=<protocol>,<idx>
```

OK

Read Command

```
AT+WIPCLOSE?
```

TBD

Test Command

```
AT+WIPCLOSE=?
```

TBD

Unsolicited response

+WIPPEERCLOSE: <protocol>,<idx>

6.2.3 Parameters and Defined Values

<protocol>	protocol type
1	UDP
2	TCP client
3	TCP server
4	FTP
<idx>	socket identifier

6.2.4 Parameter Storage

None

6.2.5 Possible Errors

Error	Description
ERROR	

6.2.6 Examples

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

6.3 Service Option handling +WIPOPT



6.3.1 Description

The +WIPOPT command is used to read and/or to configure different parameters on sockets and FTP service.

6.3.2 Syntax

- For <action>=1

Action Command

```
AT+WIPOPT=<protocol>,<idx>,<action>,<optnum>
```

```
+WIPOPT: <proto>,<idx>,<optnum>,<optval>
```

```
OK
```

- For <action>=2

Action Command

```
AT+WIPOPT=<protocol>,<idx>,<action>,<optnum>,<optval>
```

```
OK
```

Read Command

```
AT+WIPOPT?
```

```
TBD
```

<p><i>Test Command</i></p> <p>AT+WIPOPT=?</p> <p>TBD</p>

6.3.3 Parameters and 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

6.3.4 Parameter Storage

None

6.3.5 Possible Errors

Error	Description
ERROR	

6.3.6 Examples

Command	Responses
AT+WIPOPT=2,1,2,8,20 <i>Note: Set TTL for TCP client</i>	OK
AT+WIPOPT=2,1,1,8 <i>Note: Get TTL for TCP client</i>	+WIPOPT: 2,1,8,20 OK
AT+WIPOPT=3,1,2,9,10 <i>Note: Set TOS for TCP server</i>	OK
AT+WIPOPT=3,1,1,9 <i>Note: Get TOS for TCP server</i>	+WIPOPT: 2,1,9,10 OK
AT+WIPOPT=1,1,2,1,80 <i>Note: Set peer port for UDP</i>	OK

AT+WIPOPT=1,1,1,1 <i>Note: Get peer port for UDP</i>	+WIPOPT: 2,1,1,80 OK
AT+WIPOPT=4,1,2,40,1 <i>Note: Set data representation type for FTP</i>	OK
AT+WIPOPT=4,1,1,40 <i>Note: Get data representation type for FTP</i>	+WIPOPT: 4,1,1,1 OK

Preliminary

7 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 below.

7.1 File Exchange +WIPFILE



7.1.1 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.

7.1.2 Syntax

Action Command

AT+WIPFILE=<protocol>,<index>,<mode>,<filename>

CONNECT

OK

Read command

AT+WIPFILE?

OK

Test Command
AT+WIPFILE=?
 OK

7.1.3 Parameters and Defined Values

<protocol>	protocol type
	4 FTP
<idx>	Channel identifier
<mode>	used for file transfer
	1 file retrieval: Wireless CPU 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.
<filename>	specifies the name of the file to send or retrieve. The maximum file length is limited to 128 characters. The actual filename, including path name has to be used.

7.1.4 Parameter Storage

None

7.1.5 Possible Errors

TBD

7.1.6 Examples

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

7.2 Socket data exchange +WIPDATA



7.2.1 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.

7.2.1.1 TCP Sockets in Continuous mode

In continuous mode, an [ETX] character is considered as an end of data. When a TCP socket is shutdown by peer, an [ETX] character will be sent on the UART. Similarly, when the host writes an [ETX] character on a UART, the local socket will be shutdown and the peer socket will be informed of this shutdown. From now on, it is not possible to send data on this shutdown socket.

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

To close all sockets at once, “+++” sequence should be sent followed by AT+WIPCLOSE command.

7.2.1.2 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 a UDP socket is collected till an [ETX] character is encountered 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 end of the datagram.

In case an [ETX] 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.

7.2.1.3 Leaving Continuous 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>,<index>,0 on another UART in AT mode

7.2.2 Syntax

Action Command

AT+WIPDATA=<protocol>,<idx>,<mode>[,<size>]

CONNECT

Read Command

AT+WIPDATA?

TBD

Test Command

AT+WIPDATA=?

TBD

- For <protocol>=1

Unsolicited response

+WIPDATA: <protocol>,<idx>,<datagram size>,<peer IP>,<peer port>

- For <protocol>=2

Unsolicited response

+WIPDATA: <protocol>,<idx>,<number of readable bytes>

7.2.3 Parameters and 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. In this mode, size of the buffer need not be mentioned.
<size>	size of data in bytes. This parameter is only applicable for <mode>= 2 or 3.	

*: Applicable for UART from where +WIPDATA command is sent.

7.2.4 Parameter Storage

None

7.2.5 Possible errors

Error	Description
+CME ERROR: 851	Bad state

7.2.6 Examples

Command	Responses
AT+WIPDATA=1,5,0,10 <i>Note: UDP with index 5 sends 10 bytes of data in unmap mode</i>	CONNECT <read/write data> OK
AT+WIPDATA=2,5,0,10 <i>Note: TCP with index 5 sends 10 bytes of data in unmap mode</i>	CONNECT <read/write data> OK
AT+WIPDATA=2,5,1 <i>Note: TCP Client with index 5 can send/read data in continuous mode</i>	CONNECT <read/write data> +++ OK <i>Note: +++ sequence causes the UART to switch to AT mode</i>

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

8 Ping Services +WIPPING



8.1 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.

Preliminary

8.2 Syntax

Action Command

```
AT+WIPPING=<host>,[<repeat>,<interval>,[<timeout>,[<nwrite>,[<ttl>]]  
]]
```

OK

Read Command

```
AT+WIPPING?
```

OK

Test Command

```
AT+WIPPING=?
```

OK

Unsolicited response

```
"+WIPPING:<timeout_expired>,<packet_idx>,<response_time>"
```

8.3 Parameters and Defined Values

<host>	host name or IP address (string)
<repeat>	number of packets to send Range : 1-65535 (Default = 1)
<interval>	number of milliseconds between packets Range : 1-65535 (Default = 2000)
<timeout>	number of milliseconds before a packet is considered lost Range : 1-65535 (Default = 2000)
<ttl>	IP packet Time To Live. Default is set by the WIP_NET_OPT_IP_TTL +WIPCFG option Range : 1-255
<nwrite>	size of packets Range : 1-1500 (Default = 64)
<timeout_expired>	PING result 0: <timeout> expired before the response was received 1: PING response received before <timeout>
<packet_idx>	packet index in the sequence
<response_time>	PING response time in millisecond

8.4 Parameter Storage

None

Preliminary

8.5 Possible Errors

Error	Description
+CME ERROR: 800	Invalid option
+CME ERROR: 801	Invalid option value
+CME ERROR: 819	Error on ping channel

Preliminary

8.6 Examples

Command	Responses
AT+WIPPING=? <i>Note: Test command</i>	OK
AT+WIPPING? <i>Note: Read command</i>	OK
AT+WIPPING="www.wavecom.com" <i>Note: Ping "www.wavecom.com"</i>	+WIPPING: 1,0,0

Preliminary

9 Examples of Application

9.1 TCP Socket

9.1.1 TCP Server Socket

9.1.1.1 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 of PPP login
OK

AT+WIPBR=2,6,1,"passwd" //Set password for PPP login
OK

AT+WIPBR=4,6,0 //Start GPRS bearer
OK
```



```
AT+WIPCREATE=3,1,80,5,9           // Create the server on port 80, idx=0
OK
+WIPACCEPT: 3,5                    // unsolicited: the server accepted
                                    // connection; resulting TCP client
                                    // index 5

AT+WIPDATA=2,5,1                   // exchange data on socket idx 5:
CONNECT
...                                  // read, write...
+++                                  // back to AT mode
AT+WIPCLOSE=2,5                     // close the TCP client socket index 5
OK
```

Preliminary

9.1.1.2 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=0
OK

+WIPACCEPT: 3,5 // unsolicited: the server accepted
// connection; resulting TCP client
```

```
                                // index 5

AT+WIPDATA=2,5,1                // exchange data on socket idx 5:
CONNECT
...                               // read, write...
+++                               // back to AT mode
AT+WIPCLOSE=2,5                // close the TCP client socket index 5
OK
```

Preliminary

9.1.2 TCP Client Socket

9.1.2.1 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 of PPP login
OK

AT+WIPBR=2,6,1,"passwd" //Set password for PPP login
OK

AT+WIPBR=4,6,0 //Start GPRS bearer
OK

AT+WIPCREATE=3,1,80,5,9 //Create the server on port 80, idx=0
OK

+WIPREADY: 2,1 // unsolicited: the server accepted
//connection;resulting TCP clientindex 5
```

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

Preliminary

9.1.2.2 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=0
OK

+WIPREADY: 2,1 // unsolicited: the server accepted
//connection;resulting TCP clientindex 5
```

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

Preliminary

9.2 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 of PPP login
OK

AT+WIPBR=2,6,1,"passwd" //Set password for PPP login
OK

AT+WIPBR=4,6,0 //Start GPRS bearer
OK

AT+WIPCREATE =1,1,80,"www.google.co.in",80 //Start UDP socket
OK

WIPREADY: 1,1

AT+WIPDATA=1,1,1 // exchange data on socket idx 1:
```



```
CONNECT
... // read, write...
+++ // back to AT mode
AT+WIPCLOSE=1,1 // close the UDP socket index 1
OK
```

Preliminary

9.3 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 of PPP login
OK

AT+WIPBR=2,6,1,"passwd" //Set password for PPP login
OK

AT+WIPBR=4,6,0 //Start GPRS bearer
OK

AT+WIPPING="www.wavecom.com" //Start PING session
OK

+WIPPING:1,0,0
```

9.4 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 of PPP login
OK

AT+WIPBR=2,6,1,"passwd" //Set password for PPP login
OK

AT+WIPBR=4,6,0 //Start GPRS bearer
OK

AT+WIPCREATE=4,1,"FTP server",21,"username","passwd"
OK //Create FTP session
```

```
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
```

Preliminary

10 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
837	Bad proto
838	No more free socket
839-849	Reserved for future use
850	Unknown reason

851	Bad state
-----	-----------

Preliminary

11 Notes

11.1 Bearers Handling

- 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 2, +CMUX command for CMUX virtual ports and GSM/GPRS AT commands).
- several bearer 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
- WIP_BOPT_DIAL_REDIALCOUNT and WIP_BOPT_DIAL_REDIALDELAY options will not be implemented through AT commands. Nevertheless, for future compatibility reason, Opt num 3 and 4 are kept as reserved

Preliminary

11.2 IP Protocol Services

- +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 AT+WIPCREATE is needed. Then, each file operation will be done (one AT+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
- The event +WIPPEERCLOSE: <protocol>,<idx> will be generated when the socket is closed by the peer
- 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	RW	R	-
2	string	WIP_COPT_PEER_STRADDR	RW	R	-
3	0..1	WIP_COPT_BOUND	R	-	-
4	0..65535	WIP_COPT_SND_LOWAT	-	RW	RW
5	0..65535	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 type	Meaning
0-39	-	used by and for all protocols types (including FTP).
40	boolean	"TYPE" ⇔ Data representation type. This can be Binary (value is "1") or ASCII(value is "0"). Default type is ASCII.
41	boolean	"PASSIVE" ⇔ 0/1 for des/activating Passive mode. Default configuration is passive mode.
42-49	-	Reserved for future use
	-	Reserved for other protocols.

- the opt num that can be applied only to FTP protocol is [40-41]

11.3 Data Exchange for Protocol Services

- [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.
- The “+++” escape sequence is interpreted as an [ETX] character if it is sent over serial port during data transmission.
- UART switches back to AT mode due to “+++” with 1 second guard time before and after the sequence or by sending an AT+WIPDATA=<proto>,<index>,0 on another UART in AT mode

Preliminary



wavecom[®]
Make it wireless

WAVECOM S.A. - 3 esplanade du Foncet - 92442 Issy-les-Moulineaux Cedex - France - Tel: +33(0)1 46 29 08 00 - Fax: +33(0)1 46 29 08 08
Wavecom, Inc. - 4810 Eastgate Mall - Second Floor - San Diego, CA 92121 - USA - Tel: +1 858 362 0101 - Fax: +1 858 558 5485
WAVECOM Asia Pacific Ltd. - 4/F, Shui On Centre - 6/8 Harbour Road - Hong Kong - Tel: +852 2824 0254 - Fax: +852 2824 025

www.wavecom.com