



# IP EASY USER GUIDE

80000ST10028A Rev. 24 – 2019-07-30

**TELIT**  
**TECHNICAL**  
**DOCUMENTATION**

SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE

## NOTICES LIST

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

## COPYRIGHTS

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

## COMPUTER SOFTWARE COPYRIGHTS

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

## USAGE AND DISCLOSURE RESTRICTIONS

### I. License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

### II. Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit

### III. High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

### IV. Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

### V. Third Party Rights

The software may include Third Party Right software. In this case you agree to comply with all terms and conditions imposed on you in respect of such separate software. In addition to Third Party Terms, the disclaimer of warranty and limitation of liability provisions in this License shall apply to the Third Party Right software.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES EXPRESS OR IMPLIED FROM ANY THIRD PARTIES REGARDING ANY SEPARATE FILES, ANY THIRD PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY "OTHER CODE"), AND THE USE OF ANY OR ALL THE OTHER CODE IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD PARTY LICENSORS OF OTHER CODE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODE OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## Applicability Table

### PRODUCTS

		Platform Version ID	Technology
	GL865 SERIES	10	2G
	GE865-QUAD		
	GE864 V2 SERIES		
	GL868-DUAL		
	GE910 SERIES	13	
	GE910-QUAD-V3	16	
	GE866-QUAD		
	GL865 V3 SERIES		
	GL868-DUAL V3		
	HE910 SERIES	12	
	UE910 SERIES		
	UL865 SERIES		
	UE866 SERIES		
	LE910 Cat1 SERIES	20	4G
	LE910 V2 SERIES		
	LE866 SERIES	23	
	ME866A1 SERIES		
	LE910Cx SERIES		



Platform Version ID is a reference used in the document to identify a series of modules, each module is named with the serial name followed by a suffix (series name-suffix). In addition, Platform Version ID is used in the tag of the different SW versions, e.g. 10 for SW version 10.xx.xxx, 13 for SW version 13.xx.xxx, etc.

## Contents

<b>NOTICES LIST .....</b>	<b>2</b>
<b>COPYRIGHTS .....</b>	<b>2</b>
<b>COMPUTER SOFTWARE COPYRIGHTS.....</b>	<b>2</b>
<b>USAGE AND DISCLOSURE RESTRICTIONS .....</b>	<b>3</b>
<b>APPLICABILITY TABLE .....</b>	<b>4</b>
<b>CONTENTS .....</b>	<b>5</b>
<b>FIGURES LIST .....</b>	<b>12</b>
<b>TABLES LIST.....</b>	<b>13</b>
<b>AT COMMANDS LIST .....</b>	<b>14</b>
<b>1. INTRODUCTION .....</b>	<b>15</b>
1.1. Scope .....	15
1.2. Audience.....	15
1.3. Contact Information, Support .....	15
1.4. Text Conventions.....	16
1.5. Related Documents .....	17
<b>2. PRELIMINARY INFORMATION .....</b>	<b>18</b>
<b>3. IP EASY FEATURES .....</b>	<b>19</b>
<b>4. 2G/3G MODULES .....</b>	<b>22</b>
4.1. Context Types .....	22
4.1.1. PDP Context Setting, +CGDCONT Command.....	22
4.1.2. Platform Version ID 25.....	23
4.1.2.1. Modules: LE910Cx SERIES.....	23
4.2. PDP Context Activation, #SGACT Command .....	23
4.2.1. Max Number of Active PDP Contexts .....	26
4.2.2. Automatic Activation/Reactivation, #SGACTCFG Command .....	28
4.2.3. Abort Context Activation, #SGACTCFGEXT Command.....	28
4.2.4. Authentication Setting, #SGACTAUTH Command.....	28
4.2.5. Examples.....	29
4.2.5.1. PDP Contexts and Sockets Bindings .....	29
4.2.5.2. PDP Context Activation/Reactivation .....	30
4.3. PDP Context Activation, +CGACT Command .....	30

4.4.	IP Address Information, #CGPADDR Command.....	31
4.5.	QoS Setting (2G/3G) .....	32
4.5.1.	Minimum QoS Setting, +CGQMIN Command .....	32
4.5.2.	Required QoS Setting, +CGQREQ Command .....	34
4.6.	QoS Setting (3G) .....	35
4.6.1.	Minimum QoS Setting, +CGEQMIN Command.....	35
4.6.2.	Required QoS Setting, +CGEQREQ Command.....	35
<b>5.</b>	<b>4G MODULES .....</b>	<b>36</b>
5.1.	Default EPS Bearer at Power ON .....	37
5.2.	PDP Context Setting, +CGDCONT Command.....	38
5.2.1.	Platform Version ID 20.....	38
5.2.1.1.	Modules: LE910 Series.....	38
5.2.1.2.	Modules: LE910-SV1, -SVL, SV V2 .....	39
5.2.2.	Platform Version ID 23.....	40
5.2.2.1.	Modules: LE866-SV1, ME866A1-NV .....	40
5.2.2.2.	Modules: LE866A1-NA, -KK, -JS, -KS, ME866A1-NA.....	41
5.2.3.	Platform Version ID 25.....	41
5.2.3.1.	Modules: LE910Cx SERIES.....	41
5.3.	Establish a Default EPS Bearer .....	42
5.4.	PDP Context Activation, #SGACT Command .....	42
5.4.1.	Max Number of Active PDP Contexts .....	43
5.4.2.	Platform Version ID 20.....	43
5.4.2.1.	Modules: LE910 Series.....	44
5.4.2.2.	Modules: LE910-SV1, -SVL, SV V2 .....	44
5.4.3.	Platform Version ID 23, 25.....	44
5.4.3.1.	Modules: LE866-SV1, ME866A1-NV .....	45
5.4.3.2.	Modules: LE866A1-NA, -KK, -JS, -KS, ME866A1-NA, LE910Cx .....	46
5.5.	Change a Default EPS Bearer .....	46
5.6.	Dedicated EPS Bearer.....	48
5.7.	Multiple PDN, #DUALAPN Command (Platform Version ID 20) ..	48
5.7.1.	Modules: LE910 Series.....	48
5.7.2.	Examples on #DUALAPN=0 Mode .....	50
5.7.2.1.	Defined APN.....	50
5.7.2.2.	Not Defined APN .....	51
5.7.3.	Examples on #DUALAPN=1 Mode .....	53
5.7.3.1.	Defined APN.....	53
5.7.3.2.	Not Defined APN .....	54

5.8.	Authentication Setting, #PDPAUTH Command.....	56
5.8.1.	Platform Version ID 20 and 25.....	56
5.8.1.1.	Modules: LE910 and, LE910Cx SERIES .....	56
<b>6.</b>	<b>OUTGOING SOCKETS IN ONLINE MODE.....</b>	<b>58</b>
6.1.	Sockets Configuration, #SCFG Command.....	58
6.1.1.	2G/3G Modules.....	58
6.1.2.	4G Modules .....	59
6.1.2.1.	Platform Version ID 20.....	59
6.1.2.1.1.	Modules: LE910 Series.....	59
6.1.2.1.2.	Modules: LE910-SV1, -SVL, SV V2 .....	60
6.1.2.2.	Platform Version ID 23.....	62
6.1.2.2.1.	Module: LE866-SV1, ME866A1-NV .....	62
6.2.	Open a Socket, #SD Command.....	63
6.2.1.	Example on Opening & Suspension, (2G/3G).....	63
6.3.	Pending Data, #PADCMD, #PADFWD Commands .....	65
6.4.	Resume a Connection, #SO Command .....	66
6.5.	Close a Connection, #SH, #SLASTCLOSURE Commands .....	67
6.6.	TCP/IP Options Configuration.....	68
6.6.1.	Reassemble TCP Packets, #TCPREASS Command (2G).....	68
6.6.2.	Max TCP Payload Size, #TCPMAXDAT Command (2G).....	68
6.6.3.	#BASE64 Command.....	68
<b>7.</b>	<b>INCOMING SOCKETS IN ONLINE MODE.....</b>	<b>70</b>
7.1.	Accept an Incoming Connection, #SA Command.....	72
7.2.	Check the Socket State, #SS Command .....	72
7.3.	Open UDP Connection, #SLUDP Command .....	73
7.4.	Firewall Setting, #FRWL Command .....	73
7.5.	Examples.....	74
7.5.1.	Modules in Server and Client Configurations .....	74
<b>8.</b>	<b>SOCKET CONNECTION IN COMMAND MODE .....</b>	<b>77</b>
8.1.	Open a Socket, #SD Command.....	77
8.2.	Accept an Incoming Connection, #SA Command.....	77
8.3.	Extended Socket Parameters Configuration .....	77
8.3.1.	#SCFGEXT Command .....	77
8.3.2.	#SCFGEXT2 Command .....	79
8.3.2.1.	2G modules .....	79
8.3.2.2.	3G/4G modules.....	79

8.4.	Send Data, #SSEND and #SSENDEXT Commands.....	79
8.5.	Receive Data, #SRECV Command.....	79
8.6.	Socket Information, #SI Command .....	80
8.7.	Examples.....	81
8.7.1.	Switching from COMMAND to ONLINE Mode and Back .....	81
8.7.2.	Accept a Socket Connection in COMMAND Mode.....	81
8.7.3.	SRING Unsolicited Mode: Normal.....	82
8.7.4.	SRING Unsolicited Mode: Data amount.....	83
8.7.5.	SRING Unsolicited Mode: Data view.....	84
8.7.6.	SRING Unsolicited Mode: Data view UDP .....	85
<b>9.</b>	<b>MULTI-SOCKET &amp; PROTOCOL SERVICE SOCKETS.....</b>	<b>87</b>
9.1.	2G/3G Modules.....	87
9.2.	4G Modules .....	88
9.2.1.	#PROTOCOLCFG Command.....	88
9.2.2.	Platform Version ID 20.....	88
9.2.2.1.	Modules: LE910 Series.....	88
9.2.2.2.	Modules: LE910-SV1, -SVL, SV V2 .....	90
9.2.3.	Platform version ID 23 .....	92
9.2.3.1.	Modules: LE866-SV1, ME866A1-NV .....	92
9.2.4.	Platform version ID 25 .....	93
9.2.4.1.	Modules: LE910Cx .....	93
<b>10.</b>	<b>HTTP PROTOCOL .....</b>	<b>95</b>
10.1.	#HTTPCFG Command .....	95
10.1.1.	2G Modules .....	95
10.1.2.	3G Modules .....	96
10.1.3.	4G Modules .....	97
10.1.3.1.	Platform Version ID 20.....	98
10.1.3.1.1.	Modules: LE910 Series.....	98
10.1.3.1.2.	Modules: LE910-SV1, -SVL, SV V2 .....	98
10.1.3.2.	Platform Version ID 23.....	98
10.1.3.2.1.	Modules: LE866-SV1, ME866A1-NV .....	98
10.1.3.3.	Platform Version ID 25.....	98
10.1.3.3.1.	Modules: LE910Cx .....	98
10.2.	#HTTPQRY: GET, HEAD, DELETE.....	99
10.3.	#HTTPSND: POST, PUT .....	99
10.4.	Examples.....	100
10.4.1.	GET Command (2G).....	100

10.4.2.	POST Command (2G) .....	101
10.4.3.	PUT Command (2G).....	101
10.4.4.	HTTP Client with #SD.....	102
<b>11.</b>	<b>SMTP PROTOCOL .....</b>	<b>104</b>
11.1.	2G Modules .....	104
11.2.	3G Modules .....	105
11.3.	4G Modules .....	105
11.4.	E-mail Sending, with no Attachment .....	105
11.5.	E-mail Sending with Attachment .....	106
11.6.	Examples.....	106
11.6.1.	With no Attachment .....	106
11.6.2.	With Attachment .....	107
11.6.3.	Sending with #SD .....	108
11.6.4.	Receiving with #SD.....	109
<b>12.</b>	<b>FTP PROTOCOL.....</b>	<b>110</b>
12.1.	2G/3G Modules.....	110
12.2.	4G Modules .....	111
12.3.	Examples.....	111
12.3.1.	Open Six Sockets and the FTP Connection .....	111
12.4.	FTP Time Out Setting, #FTPTO Command .....	113
12.5.	FTP Transfer Type Setting, #FTPTYPE Command.....	113
12.6.	FTP Uploading, #FTPPUT, #FTPAPPEXT Commands.....	114
12.7.	#FTPSIZE, #FTPAPP Commands .....	114
12.8.	Examples.....	114
12.8.1.	Data Uploading in ONLINE Mode .....	114
12.8.2.	Data Uploading in COMMAND Mode .....	116
12.9.	FTP Downloading .....	117
12.9.1.	Downloading in ONLINE mode .....	117
12.9.2.	Downloading in COMMAND mode.....	117
12.10.	FTP Downloading Restart.....	117
12.11.	Examples.....	118
12.11.1.	File Downloading in ONLINE Mode.....	118
12.11.2.	File Downloading in COMMAND Mode .....	119
12.11.3.	FTP and 4G Modules.....	120
<b>13.</b>	<b>PING PROTOCOL.....</b>	<b>124</b>
13.1.	2G/3G Modules.....	124

13.2.	4G Modules .....	125
<b>14.</b>	<b>EASY GSM: IP OVER CSD.....</b>	<b>126</b>
14.1.	GSM Context Setting .....	126
14.2.	GSM Context Activation.....	127
14.3.	Examples.....	127
14.3.1.	GSM/PDP Contexts & FTP Protocol .....	127
14.3.2.	GSM Context & +COPS=? Command .....	130
14.3.3.	HTTP Client with #SD.....	131
14.3.4.	FTP File Uploading .....	132
14.3.5.	E-mail with no Attachment .....	133
14.3.6.	Modules in Server and Client Configuration .....	134
<b>15.</b>	<b>DIALUP CONNECTION .....</b>	<b>137</b>
15.1.	Examples.....	137
15.1.1.	Start PPP Protocol.....	137
15.1.2.	Dial-up Using <cid>=1 by default (2G).....	138
15.1.3.	Dial-up Using <cid>=1 by default (3G).....	139
15.1.4.	4G Modules .....	140
15.1.4.1.	Dial-up & #DUALAPN Command (Platform Version ID 20) .....	140
15.1.4.2.	Dial-up Command, Platform ID 23 .....	142
<b>16.</b>	<b>APPENDIX .....</b>	<b>145</b>
16.1.	IPv6 Protocol .....	145
16.1.1.	2G Modules .....	145
16.1.2.	3G Modules .....	145
16.1.3.	4G Modules .....	145
16.2.	IPV4, IPV6 and Dual Stack .....	146
16.2.1.	2G/3G/4G Modules.....	146
16.2.2.	3G/4G Modules.....	147
16.3.	IPV6 Notation .....	147
16.3.1.	+CGPIAF Command.....	147
16.4.	Examples.....	148
16.4.1.	+CGDCONT Command & IPV6 Notation .....	148
16.4.2.	#SD Command & IPV6 Notation .....	148
16.4.3.	#FTPOPEN Command & IPV6 Notation .....	148
16.4.4.	#PING Command & IPV6 Notation .....	148
16.4.5.	#DNS Command & IPV6 Notation .....	149
16.4.6.	#FRWLIPV6 Command & IPV6 Notation .....	149

16.5.	#NCM Command .....	150
16.5.1.	#NCM Command, Mode = 1 .....	150
16.5.2.	#NCM Command, Mode = 2 .....	152
16.6.	#MBIMCFG Command .....	154
16.6.1.	Modules: LE910 Series .....	154
16.6.2.	Modules: LE910-SV1, -SVL, SV V2 .....	154
16.7.	#NWMODE Command .....	155
16.8.	Obsolete Commands .....	157
<b>17.</b>	<b>GLOSSARY AND ACRONYMS .....</b>	<b>158</b>
<b>18.</b>	<b>DOCUMENT HISTORY .....</b>	<b>160</b>

## Figures List

Fig. 1: PSD Connection.....	19
Fig. 2: Example of Ports Config. and Skt Distribution .....	19
Fig. 3: Example of CMUX and Socket Distribution .....	20
Fig. 4: Switching from COMMAND to ONLINE mode .....	21
Fig. 5: Assigned APN and Network Interfaces.....	43
Fig. 6: LE910-EU1, Multi/Protocol/Internal Sockets, and Network Interface .....	90
Fig. 7: Assigned APN does not Support Data Traffic.....	122
Fig. 8: User APN Support Data Traffic .....	123
Fig. 9: GSM CSD Call .....	126
Fig. 10: GPRS Dialup Connection.....	137
Fig. 11: Dial up connection using <cid>=2.....	141
Fig. 12: IPV4 .....	146
Fig. 13: IPV6 .....	146
Fig. 14: Dual IP Stack IPV4V6 .....	147

## Tables List

Tab. 1: 2G/3G Context Types .....	22
Tab. 2: 2G/3G Default Sockets/Contexts Binding Configuration & PDP State .....	24
Tab. 3: Example of PDP Contexts Configuration.....	30
Tab. 4: 2G/3G Default Sockets/Contexts Binding Configuration.....	58
Tab. 5: 2G/3G Sockets/Contexts Binding Configuration.....	59
Tab. 6: LE910-EU1, Network Operator Service & Multi-Socket.....	60
Tab. 7: LE910-SV1, IMS, ADMIN, INTERNET Services & Multi-Socket.....	61
Tab. 8: LE866-SV1, IMS, ADMIN, INTERNET Services & Multi-Socket.....	62
Tab. 9: Sockets/Contexts Binding and PDPs Active.....	63
Tab. 10: Sockets/Contexts Binding and Sockets Opened .....	65
Tab. 11: An Example of Sockets in Listening Mode .....	70
Tab. 12: 2G/3G Multi-Socket & Protocol Sockets.....	87
Tab. 13: LE910-EU1, Network Operator Service, Multi-Socket, and Protocols.....	89
Tab. 14: LE910-SV1, IMS, ADMIN, INTERNET Services, Multi-Socket, and Protocols...	92
Tab. 15: LE866-SV1, IMS, ADMIN, INTERNET Services, Multi-Socket, and Protocols...	93
Tab. 16: LE910Cx, Network Operator Service, Multi-Socket, and Protocols.....	94
Tab. 17: 2G Multi-Socket & HTTP Service Socket .....	95
Tab. 18: 3G Multi-Socket & HTTP Service Socket .....	96
Tab. 19: 2G Multi-Socket & SMTP Service Socket.....	104
Tab. 20: 3G Multi-Socket & SMTP Service Socket.....	105
Tab. 21: 2G/3G Multi-Socket & FTP Service Socket .....	110
Tab. 22: 2G/3G Multi-Socket & FTP Service Socket Opened.....	113
Tab. 23: 2G/3G Multi-Socket & PING Service Socket .....	124
Tab. 24: Platform Version ID & IPV4/IPV6 Addresses.....	145
Tab. 25: Obsolete and New AT Commands .....	157

## AT Commands List

The following list, organized in alphabetical order, shows the AT commands covered by this User Guide. The number close to each command indicates the page of the first AT command occurrence.

<b>AT#BASE64</b> .....	67	<b>AT#HTTPCFG</b> .....	93	<b>AT#SL</b> .....	69
<b>AT#CEERNET</b> .....	50	<b>AT#HTTPQRY</b> .....	97	<b>AT#SLASTCLOSURE</b>	
<b>AT#CGDATA</b> .....	149	<b>AT#HTTPCRV</b> .....	98	.....	66
<b>AT#CGPADDR</b> .....	31	<b>AT#HTTPSND</b> .....	97	<b>AT#SLUDDP</b> .....	71
<b>AT#DNS</b> .....	147	<b>AT#ICMP</b> .....	122	<b>AT#SMTPCFG</b> .....	102
<b>AT#DUALAPN</b> .....	48	<b>AT#NBIMCFG</b> .....	42	<b>AT#SMTPCL</b> .....	104
<b>AT#EADDR</b> .....	104	<b>AT#NCM</b> .....	42	<b>AT#SO</b> .....	65
<b>AT#EMAILD</b> .....	103	<b>AT#NWMODE</b> .....	153	<b>AT#SRECV</b> .....	77
<b>AT#EPASSW</b> .....	104	<b>AT#PADCMD</b> .....	64	<b>AT#SS</b> .....	70
<b>AT#ESMTP</b> .....	104	<b>AT#PADFWD</b> .....	64	<b>AT#SSEND</b> .....	77
<b>AT#EUSER</b> .....	104	<b>AT#PDPAUTH</b> .....	56	<b>AT#SENDEXT</b> .....	77
<b>AT#FRWL</b> .....	71	<b>AT#PING</b> .....	122	<b>AT#TCPMAXDAT</b> .....	67
<b>AT#FTPAPP</b> .....	112	<b>AT#PROTOCOLCFG</b> .....	86	<b>AT#TCPREASS</b> .....	67
<b>AT#FTPAPPEXT</b> .....	114	<b>AT#SA</b> .....	70	<b>AT+CGCONTRDP</b> ...	37
<b>AT#FTPCLOSE</b> .....	108	<b>AT#SCFG</b> .....	58	<b>AT+CGACT</b> .....	31
<b>AT#FTPGET</b> .....	115	<b>AT#SCFGEXT</b> .....	75	<b>AT+CGATT</b> .....	29
<b>AT#FTPGETPKT</b> .....	115	<b>AT#SCFGEXT2</b> .....	77	<b>AT+CGDCONT</b> .....	22
<b>AT#FTPOPEN</b> .....	108	<b>AT#SD</b> .....	31	<b>AT+CGPADDR</b> .....	51
<b>AT#FTPPUT</b> .....	112	<b>AT#SGACT</b> .....	23	<b>AT+CGPIAF</b> .....	145
<b>AT#FTPPWD</b> .....	116	<b>AT#SGACTAUTH</b> .....	28	<b>AT+CGQMIN</b> .....	32
<b>AT#FTPREST</b> .....	115	<b>AT#SGACTCFG</b> .....	28	<b>AT+CGQREQ</b> .....	34
<b>AT#FTPSIZE</b> .....	112	<b>AT#SGACTCFGEXT</b> .....	28	<b>AT+CMEE</b> .....	30
<b>AT#FTPTO</b> .....	111	<b>AT#SH</b> .....	66	<b>AT+COPS</b> .....	50
<b>AT#FTPTYPE</b> .....	111	<b>AT#SI</b> .....	78	<b>AT+WS46</b> .....	136
<b>AT#GSMCONT</b> .....	31	<b>AT#SKIPESC</b> .....	67	<b>ATD</b> .....	42

## 1. INTRODUCTION

### 1.1. Scope

This document illustrates the set of AT commands used to manage IP Easy features. Reading these pages, it is suggested to refer to the AT Command Reference Guide according to the product you are using (2G/3G/4G). The present guide describes the use of the AT commands through examples.

### 1.2. Audience

The reader is expected to have experience in 2G/3G/4G technologies as well as in Telit's AT Commands interface.

### 1.3. Contact Information, Support

The reader is expected to have experience in 2G/3G/4G technologies as well as in Telit's AT Commands interface.

- TS-EMEA@telit.com
- TS-AMERICAS@telit.com
- TS-APAC@telit.com

Alternatively, use:

<http://www.telit.com/support>

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

<http://www.telit.com>

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

## 1.4. Text Conventions

---



Danger – This information **MUST** be followed or catastrophic equipment failure or bodily injury may occur.

---



Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

---



Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

---

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

## 1.5. Related Documents

- [1] AT Commands Reference Guide, 80000ST10025a
- [2] Telit 3G Modules AT Commands Reference Guide, 80378ST10091A
- [3] Telit's Modules Software User Guide, 1vv0300784
- [4] LE910 V2 Series AT Commands Reference Guide, 80446ST10707A
- [5] SSL/TLS Users Guide, 1vv0300989
- [6] Telit CMUX Implementation User Guide, 1vv0300994
- [7] Telit 3G Modules Ports Arrangements User Guide, 1vv0300971
- [8] GE910 Series Ports Arrangements User Guide, 1vv0301049
- [9] LE910 V2, LE910 Cat1 NCM Protocol User Guide, 1vv0301246
- [10] LE910 V2, LE910 Cat1 Ports Arrangements User Guide, 1vv0301252
- [11] LE866 Series AT Commands Reference Guide, 80471ST10691A
- [12] LE866, ME866A1 Ports Arrangements User Guide, 1vv0301469
- [13] LE910Cx AT Commands Reference Guide, 80502ST10950A

## 2. PRELIMINARY INFORMATION

The AT commands covered by this guide are not described with an exhaustive list of their parameters and their related meanings, it is given more space to examples describing the major concepts about the IP Easy features. To get information on AT commands syntax and related parameters, refer to the AT Command Reference Guide according to the module you are using, see chapter 1.5.

### 3. IP EASY FEATURES

Telit modules provide a set of communication features accessible to the user through AT commands. This environment is a software layer based on the TCP/IP protocol stack and a set of AT commands, it is called IP Easy. By means of these AT commands, the user device, connected through a serial line to the Telit module, exchanges data with a remote device on Internet network, refer to Fig. 1. Telit module establishes a connection over a radio interface using its internal TCP/IP protocol stack, and exchanges TCP/IP packets with the remote device. In this scenario, the user device does not need an own TCP/IP protocol stack.

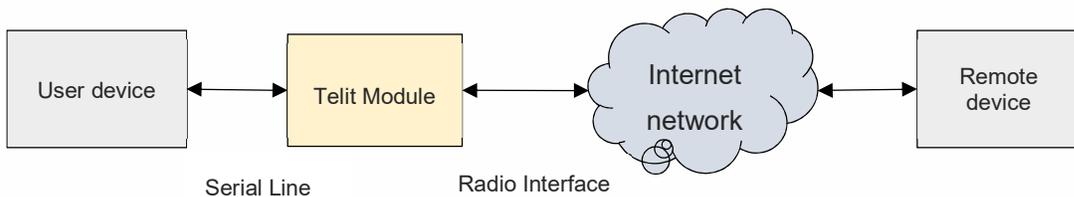


Fig. 1: PSD Connection

IP Easy environment provides the following features:

- I. Multi-socket environment. It supports up to N sockets, the N value depends on the module you are using. To know the N value, use the AT#SCFG=? Test command. The N socket connections can be distributed over the available physical ports, as shown in Fig. 2, and Fig. 3. The AT Commands Interfaces (AT Parsers: AT0, AT1, AT2) can work in COMMAND mode or in ONLINE mode. In COMMAND mode, the user can send AT commands which are parsed by the AT parser, and, on the same physical line, the user receives the AT commands responses.

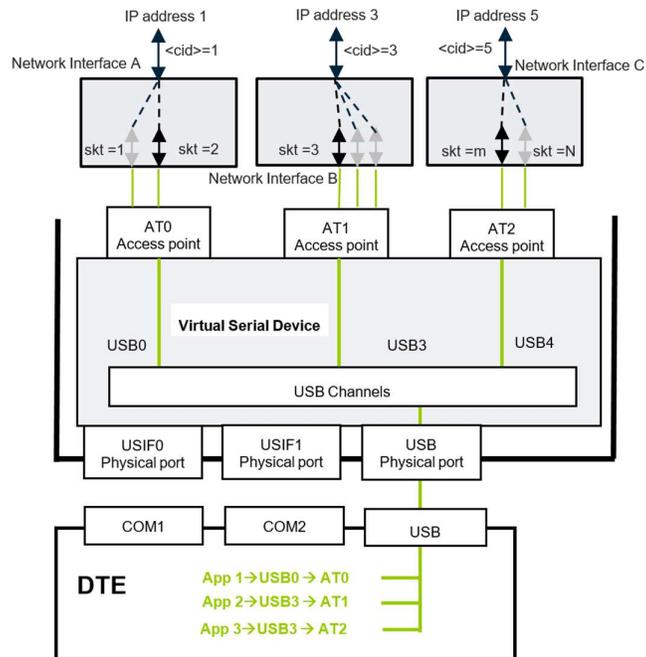


Fig. 2: Example of Ports Config. and Skt Distribution

In ONLINE mode, the data entered by the user are forwarded on the current connection toward the remote host. The data sent by the remote host are forwarded, on the same physical line, to the user.

You can use #PORTCFG command to manage the physical ports configuration of the module. In Fig. 2, and Fig. 3 the gray rows indicate sockets that are open, working in background, and not exchanging data with the user. The black rows indicate the sockets that currently are exchanging data with the user.

The Fig. 3 shows an example using the #PORTCFG command and the CMUX tool (VC1, VC2, VC3):

- App 1 connected on VC1/AT0 manages two sockets. The picture shows the time interval in which socket=1 is suspended, and socket=2 is exchanging data with the user through AT0 Interface.
- App 2 connected on VC2/AT1 manages two sockets. The picture shows the time interval in which socket=4 is suspended, and socket=3 is exchanging data with the user through AT1 Interface. Note that the sockets belong to two different Network Interfaces, and the Network Interface can manage multiple sockets.
- App 3 connected on VC3/AT2 manages two sockets. The picture shows the time interval in which socket=5 is suspended, and socket=N is exchanging data with the user through AT2 Interface.

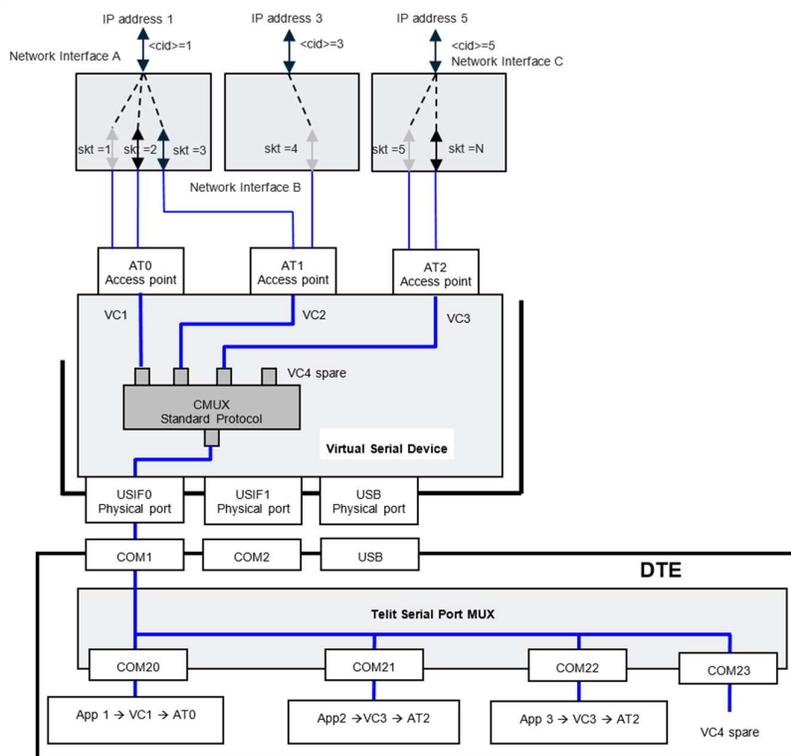


Fig. 3: Example of CMUX and Socket Distribution

To have information on CMUX tool refer to document [6], and on #PORTCFG command refer to documents [7], [8], [10], [12], or [13] according to the used module.

- II. The max number of the PDP context that can be defined (+CGDCONT command), and the max number of PDP contexts that can be activated at the same time (#SGACT command) depend on the module you are using. For each active PDP context, the network assigns to the module an IP address, see Fig. 2 and Fig. 3.
  
- III. In Multi-socket, the ONLINE mode can be suspended with the escape sequence (+++), and the AT Interface enters COMMAND mode. During COMMAND mode, the data received from the remote host are buffered in the socket buffer.

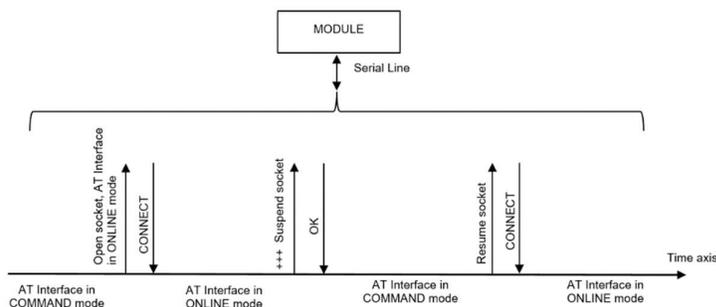


Fig. 4: Switching from COMMAND to ONLINE mode

Data will be displayed on AT Interface after socket resumption (#SO command), see examples in chapters 8.7.1.

- IV. Multiple Multi-socket connections and simultaneous client Services, example FTP, can be opened. See example in chapter 12.3.1.

## 4. 2G/3G MODULES

### 4.1. Context Types

The Tab. 1 shows the context types provided by the 2G/3G modules. Two different AT commands are used to define PDP and GSM contexts, but only one command (#SGACT) activates them.

PDP Context Multi-Socket environment <cid>≠0	GSM Context Used in CSD connection Only <cid>=0
AT+CGDCONT=<cid>,... AT#SGACT=<cid>,1	AT#GSMCONT=0,... AT#SGACT=0,1
Refer to next chapters	Refer to chapter 14

Tab. 1: 2G/3G Context Types

#### 4.1.1. PDP Context Setting, +CGDCONT Command

Use +CGDCONT command to define a PDP context. The command syntax is:

**AT+CGDCONT** =[<cid>[,<PDP\_type>[,<APN>[,<PDP\_addr>[,<d\_comp>  
[,<h\_comp>[,<pd1>[,...[,pdN]]]]]]]]]]]

To get parameters descriptions, refer to documents [1] or [2], according to the module in use.

Here is an example of PDP context definition.

**AT+CGDCONT= 1,"IP","Access\_Point\_Name","0.0.0.0",0,0**  
OK

Use the AT+CGDCONT=? test command to know the parameters ranges of the command. Here are two examples.

#### Example (2G)

Assume to use a module belonging to Platform Version ID 10, 13, 16. Type in the AT+CGDCONT=? test command to get the supported parameters values. The returned message shows the <cid> range: 1-5 of the PDP contexts.

**AT+CGDCONT=?**  
+CGDCONT: (1-5),"IP",,,(0,1),(0,1)  
+CGDCONT: (1-5),"IPV6",,,(0,1),(0,1)  
OK

### Example (3G)

Assume to use a module belonging to Platform Version ID 12. Type in the AT+CGDCONT=? test command to get the supported parameters values. The returned message shows the <cid> range: 1-5 of the PDP contexts.

#### AT+CGDCONT=?

```
+CGDCONT: (1-5),"IP",,,(0,1),(0,1)
+CGDCONT: (1-5),"IPV6",,,(0,1),(0,1)
+CGDCONT: (1-5),"IPV4V6",,,(0,1),(0,1)
OK
```

#### 4.1.2. Platform Version ID 25

Use +CGDCONT command to define a PDP context. The command syntax is:

**AT+CGDCONT=[<cid>[,<PDP\_type>[,<APN>[,<PDP\_addr>[,<d\_comp>[,<h\_comp>[,<IPv4AddrAlloc>[,<Emergency\_ind>]]]]]]]]]**

##### 4.1.2.1. Modules: LE910Cx SERIES

To get parameters descriptions, refer to documents [13], according to the module in use.

Here is an example of PDP context definition.

```
AT+CGDCONT= 1,"IP","Access_Point_Name","",0,0,0,0
OK
```

Use the AT+CGDCONT=? test command to know the parameters ranges of the command. Here are two examples.

Assume to use a module belonging to Platform Version ID 25. Type in the AT+CGDCONT=? test command to get the supported parameters values. The returned message shows the <cid> range: 1-24 of the PDP contexts.

#### AT+CGDCONT=?

```
+CGDCONT: (1-24),"IP",,,(0-2),(0-4),(0-1),(0-1)
+CGDCONT: (1-24),"PPP",,,(0-2),(0-4),(0-1),(0-1)
+CGDCONT: (1-24),"IPV6",,,(0-2),(0-4),(0-1),(0-1)
+CGDCONT: (1-24),"IPV4V6",,,(0-2),(0-4),(0-1),(0-1)
OK
```

## 4.2. PDP Context Activation, #SGACT Command

The #SGACT command activates/deactivates one of the PDP contexts defined with +CGDCONT command. The command syntax is:

**AT#SGACT= <cid>,<stat>[,<userId>,<pwd>]**

To get parameters descriptions, refer to documents [1] or [2], according to the used module.

Type in the AT#SGACT=? test command to get the supported <cid> values range.

**AT#SGACT=?**

#SGACT: (0-5),(0,1) ← <cid> range: 0-5

OK

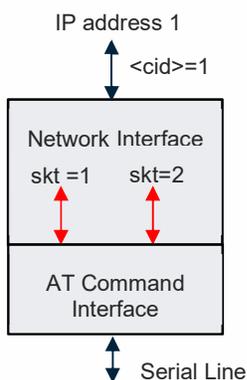
The <cid>=0 is reserved for GSM context, refer to chapter 14.

Tab. 2 shows the five PDP contexts in not active state, and the default sockets/contexts binding configuration provided by the modules. Use the #SCFG command to manage the socket configuration, refer to chapter 6.1.

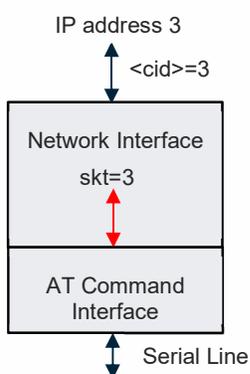
IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Not active				
1	bound	/	/	/	/
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	bound	/	/	/
5	/	bound	/	/	/
6	/	bound	/	/	/

Tab. 2: 2G/3G Default Sockets/Contexts Binding Configuration & PDP State

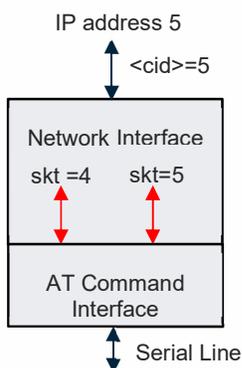
AT#SGACT=<cid>,1 activates the PDP context identified by the <cid>, and creates a Network Interface, related to the <cid>, to use the TCP/IP protocol running on the module. The Network Interface is the connection between the IP address, assigned to the module by the network, and the internal structure of the module.



bind <cid>=1 to socket=1: **AT#SCFG=1,1,...** (see chapter 6.1)  
 bind <cid>=1 to socket=2: **AT#SCFG=2,1,...**  
 Activate <cid>=1: **AT#SGACT=1,1**  
**Note:** the sockets are not open.



bind <cid>=3 to socket=3: **AT#SCFG=3,3,...**  
 Activate <cid>=3: **AT#SGACT=3,1**  
**Note:** the sockets are not open.



bind <cid>=5 to socket=4: **AT#SCFG=4,5,...**  
 bind <cid>=1 to socket=2: **AT#SCFG=5,5,...**  
 Activate <cid>=5: **AT#SGACT=5,1**  
**Note:** the sockets are not open.

The command AT#SGACT=<cid>,0 deletes the PDP context and its Network Interface.

#### 4.2.1. Max Number of Active PDP Contexts

As stated in chapter 3, the max number of the PDP contexts that can be defined (+CGDCONT command), and the max number of PDP contexts that can be activated at the same time (#SGACT command) depend on the module you are using.

This example shows how to check the max number of PDP contexts that can be activated at the same time by the module under test.

Assume to use the HE910 module. It supports up to three active PDP contexts, for each active PDP context the network assigns to the module an IP address. This is valid for all modules having Platform Version ID = 12.

Check the module type under test.

```
AT+CGMM  
HE910  
OK
```

Enable ERROR report in verbose format.

```
AT+CMEE=2  
OK
```

Check the current PDP contexts configuration.

```
AT+CGDCONT?  
+CGDCONT: 1,"IP"," Access_Point_Name ","",0,0  
+CGDCONT: 2,"IP"," Access_Point_Name ","",0,0  
+CGDCONT: 3,"IP"," Access_Point_Name ","",0,0  
+CGDCONT: 4,"IP"," Access_Point_Name ","",0,0  
+CGDCONT: 5,"IP"," Access_Point_Name ","",0,0  
OK
```

Check the current Multi-sockets/PDP contexts configuration.

```
AT#SCFG?  
#SCFG: 1,5,300,90,600,50  
#SCFG: 2,5,300,90,600,50  
#SCFG: 3,4,300,90,600,50  
#SCFG: 4,3,300,90,600,50  
#SCFG: 5,2,300,90,600,50  
#SCFG: 6,1,300,90,600,50  
OK
```

Check if some PDP context is active. The following response shows that no PDP contexts are active.

```
AT#SGACT?  
#SGACT: 1,0  
#SGACT: 2,0  
#SGACT: 3,0  
#SGACT: 4,0  
#SGACT: 5,0  
OK
```

Activate PDP context identified by <cid>=1.

**AT#SGACT=1,1**

#SGACT: 2.192.12.25

OK

Activate PDP context identified by <cid>=2.

**AT#SGACT=2,1**

#SGACT: 2.192.17.170

OK

Activate PDP context identified by <cid>=3.

**AT#SGACT=3,1**

#SGACT: 2.192.20.171

OK

Check which are the active PDP contexts.

**AT#SGACT?**

#SGACT: 1,1

#SGACT: 2,1

#SGACT: 3,1

#SGACT: 4,0

#SGACT: 5,0

OK

Try to activate the fourth and fifth PDP context

**AT#SGACT=4,1**

+CME ERROR: activation failed

**AT#SGACT=5,1**

+CME ERROR: activation failed

#### 4.2.2. Automatic Activation/Reactivation, #SGACTCFG Command

Use #SGACTCFG command to enable/disable the automatic activation/reactivation of the PDP context specified by the <cid> identifier. The PDP context is automatically activated after every:

- attach procedure,
- network PDP context deactivation,
- SIM removal,

if at least one IP Easy socket has been configured to use the PDP context (see #SCFG). The command syntax is:

**AT#SGACTCFG=<cid>,<retry>[,<delay>[,<urcmode>]]**

To get parameters descriptions, refer to documents [1] or [2], according to the used module.

#### 4.2.3. Abort Context Activation, #SGACTCFGEXT Command

Use #SGACTCFGEXT command to abort a context activation attempt. The command syntax is:

**AT#SGACTCFGEXT=<cid>,<abortAttemptEnable>  
[,<unused>[,<unused>[,<unused>]]]**

To get parameters descriptions, refer to documents [1] or [2], according to the used module.

#### 4.2.4. Authentication Setting, #SGACTAUTH Command

Use #SGACTAUTH command to set authentication. The command syntax is:

**AT#SGACTAUTH=<type>**

To get parameters descriptions, refer to documents [1] or [2], according to the used module.

## 4.2.5. Examples

### 4.2.5.1. PDP Contexts and Sockets Bindings

Check the sockets/PDP contexts binding configuration. Refer to chapter 6.1 to get more information on #SCFG command. The following response shows the default configuration.

**AT#SCFG?**

```
#SCFG: 1,1,300,90,600,50
#SCFG: 2,1,300,90,600,50
#SCFG: 3,1,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,2,300,90,600,50
OK
```

Before activating PDP contexts <cid>=3 and <cid>=4, bind the sockets <connId>=1 and <connId>=5 respectively to PDP contexts <cid>=3 and <cid>=4.

**AT#SCFG = 1, 3, 300, 90, 600, 50**

OK

**AT#SCFG = 5, 4, 300, 90, 600, 50**

OK

Check the new sockets/PDP contexts binding configuration.

**AT#SCFG?**

```
#SCFG: 1,3,300,90,600,50
#SCFG: 2,1,300,90,600,50
#SCFG: 3,1,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,4,300,90,600,50
#SCFG: 6,2,300,90,600,50
OK
```

Assume that the PDP contexts has been set as shown below.

**AT+CGDCONT?**

```
+CGDCONT: 3,"IP","Access_Point_Name","0.0.0.0",0,0
+CGDCONT: 4,"IP","Access_Point_Name","0.0.0.0",0,0
OK
```

Check if the module is attached.

**AT+CGATT?**

```
+CGATT: 1          ← the module is attached.
OK
```

**AT#SGACT = 3,1**

← activate the PDP context 3

```
#SGACT: 31.157.55.95
OK
```

**AT#SGACT = 4,1**

← activate the PDP context 4

```
#SGACT: 31.159.34.186
OK
```

On success, the #SGACT command returns the IP address assigned to the module by the network. The user application can use the just received IP address for its targets. The PDP context deactivation frees the network resources.

The Tab. 3 shows the new PDP contexts configuration.

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	No active	No active	Active	Active	No active
1	/	/	bound	/	/
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	bound	/	/	/
5	/	/	/	bound	/
6	/	bound	/	/	/

Tab. 3: Example of PDP Contexts Configuration

#### 4.2.5.2. PDP Context Activation/Reactivation

Set activation/reactivation of PDP context <cid>=1, with 3 attempts.

**AT#SGACTCFG=1,3**

OK

Before using #SGACTCFG command, the <cid> must be connected to a socket, see AT#SCFG? read command.

### 4.3. PDP Context Activation, +CGACT Command

The +CGACT command activates a PDP context defined by the +CGDCONT command, but the PDP context cannot be used with Multi-socket commands. Therefore, if you try to open a socket with #SD, you get an error message indicating “context not opened”.



It is recommended to use the command #SGACT instead of the command +CGACT.

See the following example.

Enable ERROR report in verbose format.

**AT+CMEE=2**

OK

Check the sockets/PDP contexts binding configuration.

**AT#SCFG?**

#SCFG: 1,1,300,90,600,50

#SCFG: 2,2,300,90,600,50

#SCFG: 3,2,300,90,600,50

#SCFG: 4,2,300,90,600,50

#SCFG: 5,2,300,90,600,50

#SCFG: 6,2,300,90,600,50

OK

Check the PDP contexts state.

**AT#SGACT?**

```
#SGACT: 0,0
#SGACT: 1,0
#SGACT: 2,0
OK
```

Activate the PDP context identified by <cid>=1.

```
AT+CGACT=1,1
OK
```

```
AT#SGACT?
#SGACT: 0,0
#SGACT: 1,1
#SGACT: 2,0
OK
```

Open the connection.

```
AT#SD=1,0,20510,"server_address",0,0,1
+CME ERROR: context not opened
```

#### 4.4. IP Address Information, #CGPADDR Command

Once activated the PDP or GSM<sup>1</sup> context, to get the IP address assigned by the network to the module, use #CGPADDR command. If no <cid> are specified, it reports the addresses for all activated contexts.

As already stated:

- <cid>=0 identifies the GSM context
- <cid>=1-5 identify PDP contexts.

```
AT#CGPADDR= [<cid>[,<cid>[,...]]]
```

To get parameters descriptions refer to documents [1], [2] or [13], according to the used module.

Check the defined PDP contexts.

```
AT+CGDCONT?
+CGDCONT: 1,"IP","Access_Point_Name","0.0.0.0",0,0
+CGDCONT: 2,"IP","Access_Point_Name","0.0.0.0",0,0
+CGDCONT: 3,"IP","Access_Point_Name","0.0.0.0",0,0
+CGDCONT: 4,"IP","Access_Point_Name","0.0.0.0",0,0
+CGDCONT: 5,"IP","Access_Point_Name","0.0.0.0",0,0
OK
```

Check the defined GSM context.

```
AT#GSMCONT?
#GSMCONT: 0,"IP","ISP_phone_number"
OK
```

Use AT#CGPADDR=? test command to check which <cid> are defined.

<sup>1</sup> Refer to chapter 14.  
80000ST10028A Rev. 23

**AT#CGPADDR=?**

#CGPADDR: (0,1,2,3,4,5)

OK

Remove GSM context identified by <cid>=0

**AT#GSMCONT=0**

OK

Remove PDP context identified by <cid>=3

**AT+CGDCONT=3**

OK

Use AT#CGPADDR=? test command to check the new group of defined <cid>.

**AT#CGPADDR=?**

#CGPADDR: (1,2,4,5)

OK

Define GSM context. Refer to chapter 14.

**AT#GSMCONT=0,"IP","ISP\_phone\_number"**

OK

Activate the GSM context. The command returns the IP address assigned by the network to the module.

**AT#SGACT = 0,1**

#SGACT: "10.137.93.60"

OK

Get the IP address associated to the GSM context.

**AT#CGPADDR = 0**

#CGPADDR: 0,"10.137.93.60"

OK

**NOTE:** Platform Version ID 25:

- I. AT#GSMCONT is not supported.
- II. <cid>=0 is not supported. For more information about AT+CGDCONT, refer to chapter 4.1.1.1

## 4.5. QoS Setting (2G/3G)

### 4.5.1. Minimum QoS Setting, +CGQMIN Command

Use the +CGQMIN command to specify the minimum QoS profile under which the 2G connection quality is not anymore acceptable, and the connection will be terminated. The command syntax is:

**AT+CGQMIN=[<cid>[,<precedence>[,<delay>[,<reliability>[,<peak>[,<mean>]]]]]]**

Refer to documents [1] or [2] to get parameters descriptions. In addition, refer to standards 3GPP TS 27.007, 3GPP TS 23.107, and 3GPP TS 03.60 (Release 1998).

**NOTE:**

- I. The +CGQMIN command can modify the 3G QoS according to 3GPP 23.107.
- II. If the minimum requirements are too high, PDP activation could be impossible due to the lack of the network resources needed to guarantee the required QoS. In this case, the minimum quality requirements must be reduced. It is suggested to use the default setting values.

Here are two examples.

#### Example 1

Assume to use a module belonging to Platform Version ID 16. Suppose that the PDP context has been already defined and identified by <cid>=1. Now, bind the PDP context <cid>=1 to the desired minimum QoS profile.

```
AT+CGQMIN= 1,2,0,0,5,4
OK
```

Use AT+CGQMIN=? test command to check the supported value of the PDP type and the values ranges of all other parameters. Currently is supported only the "IP" Packet Data Protocol type.

```
AT+CGQMIN=?
+CGQMIN: "IP",(0-3),(0-4),(0-5),(0-9),(0-18,31)
OK
```

#### Example 2

Assume to use a module belonging to Platform Version ID 16.

Check the already defined PDP contexts.

```
AT+CGDCONT?
+CGDCONT: 1,"IP","Access_Point_Name","",0,0
+CGDCONT: 2,"IP","Access_Point_Name","",0,0
+CGDCONT: 3,"IP","Access_Point_Name","",0,0
OK
```

Check the QoS of the defined PDP contexts.

```
AT+CGQMIN?
+CGQMIN: 1,0,0,0,0,0
+CGQMIN: 2,0,0,0,0,0
+CGQMIN: 3,0,0,0,0,0
OK
```

Remove the PDP context <cid>=2.

```
AT+CGDCONT=2
OK
```

Check the remained PDP contexts.

```
AT+CGDCONT?
+CGDCONT: 1,"IP","Access_Point_Name","",0,0
+CGDCONT: 3,"IP","Access_Point_Name","",0,0
```

OK

Check the QoS of the remained PDP contexts

**AT+CGQMIN?**

+CGQMIN: 1,0,0,0,0,0

+CGQMIN: 3,0,0,0,0,0

OK

#### 4.5.2. Required QoS Setting, +CGQREQ Command

Use the +CGQREQ command to specify the required QoS profile to the 2G network. The QoS profile will be used when the PDP context is activated. The command syntax is:

**AT+CGQREQ = [<cid>[,<precedence>[,<delay>[,<reliability>[,<peak>[,<mean>]]]]]]]**

Refer to documents [1] or [2] to get parameters descriptions. In addition, refer to standards 3GPP TS 27.007, 3GPP TS 23.107, 3GPP TS 03.60 (Release 1998).

#### NOTE:

- I. The +CGQREQ command can modify the 3G QoS according to 3GPP 23.107.
- II. It is suggested to use the default setting values.

Check the requested QoS of the defined PDP contexts

**AT+CGQREQ?**

+CGQREQ: 1,0,0,0,0,0

+CGQREQ: 3,0,0,0,0,0

OK

Modified the requested QoS bound to PDP context <cid>=1

**AT+CGQREQ= 1,1,0,0,0,31**

OK

Check the requested QoS of the defined PDP contexts

**AT+CGQREQ?**

+CGQREQ: 1,1,0,0,0,31

+CGQREQ: 3,0,0,0,0,0

OK

Test command returns the supported value of the PDP type and the values ranges of all other parameters. Currently is supported only the "IP" PDP type.

**AT+CGQREQ=?**

+CGQREQ: "IP", (0-3), (0-4), (0-5), (0-9), (0-18,31)

OK



## 5. 4G MODULES

In GPRS and 3G networks, the data session is established by means of the Packet Data Protocol (PDP) Context procedure. Before the PDP context is established, the module performs the attach procedure, which communicates to the network that the module is powered on. After the attach procedure is completed, the module can do the first PDP Context procedure that will establish the data session and allocate an IP address to the module. This PDP Context will have a QoS associated with it, based on the current needs. If the module needs to have multiple data sessions, it will do a second PDP Context activation.

In LTE (4G) modules there are two types of data session setups.

- **Default Evolved Packet System (EPS) Bearer.**

When module attaches to the network at power on, it will be assigned default bearer that remains if module is attached. The default bearer will only support a nominal QoS (non-GBR bearer), but that should be enough for services not requiring GBR. module can have additional default bearer as well. Each default bearer comes with a separate IP address.

When the module needs to establish a service with a required QoS, a dedicated bearer will be established. This will have the QoS requirements needed for the service.

- **Dedicated Evolved Packet System (EPS) Bearer.**

Dedicated bearer is created when the requested service cannot be fulfilled through default bearer. Some services require a high level of QoS like VoIP, video etc. In this case, a dedicated bearer will be established with required QoS (can be GBR or non-GBR).

Dedicated bearer does not require separate IP address, only additional default bearer needs an IP address and therefore dedicated bearer is always linked to one of the default bearers established previously.

It is possible activate in dynamic mode or contextually at the default bearer one or more bearers dedicated to specific services.

## 5.1. Default EPS Bearer at Power ON

Assume to use the module LE910-EU1 (Platform Version ID 20). The following example shows that the PDP context <cid>=1 has not defined the Access Point Name (default configuration). During the attach procedure, the network assigns to the module the Default EPS Bearer that supports the services provided by the used Network Operator.

On the right side, for comparison, is also used a module having the Platform Version ID 13.

### Power on the LE910-EU1 module (4G).

By default, the PDP context <cid>=1 has not defined APN.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IPV4V6","", "",0,0
OK
```

When the network recognizes the attach request with a not defined APN, the network assigns to <cid>=1 its Default EPS Bearer. The module is 4G attached.

**AT+CGATT?**

```
+CGATT: 1
OK
```

After the Attach procedure, the <cid>=1 identifies the Default EPS Bearer assigned by the network, and it is automatically activated.

**AT+CGACT?**

```
+CGACT: 1,1
OK
```

### LE910-EU1 module (con't)

List the dynamic parameters regarding the Default EPS Bearer assigned by the network and associated to <cid>=1 context identifier.

**AT+CGCONTRDP=1**

```
+CGCONTRDP:
1,          cid
5,          bearer_id
"APN_Assigned_by_Network", APN
"10.178.43.36.255.0.0.0", ip_addr and subnet_mask
"10.178.43.37", gw_addr
"213.230.129.10", DNS_prim_addr
"0.0.0.0", DNS_sec_addr
"0.0.0.0", P_CSCF_prim_addr
"0.0.0.0", P_CSCF_sec_addr
OK
```

### Power on a GE910-QUAD module (2G).

By default, the PDP context <cid>=1 is not defined.

**AT+CGDCONT?**

```
OK
```

The module is GPRS attached (2G).

**AT+CGATT?**

```
+CGATT: 1
OK
```

After the attached procedure, no PDP Contexts are active.

**AT+CGACT?**

```
OK
```



Use the AT+CGDCONT? read command to get the current PDP contexts configurations. The following is the default configuration.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IPV4V6",,,,,,0,0
OK
```

This module provides a set of 15 <cid> (1-15). By default, the <cid>=1 is dedicated to the Default EPS Bearers. For more information about the socket/PDP context binding, refer to chapter 6.1.2.1.1.

5.2.1.2. Modules: LE910-SV1, -SVL, SV V2

Use the AT+CGDCONT=? test command to get the ranges of the supported values. <cid> range: 1-6.

**AT+CGDCONT=?**

```
+CGDCONT: (1-6),"IP",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
+CGDCONT: (1-6),"IPV6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
+CGDCONT: (1-6),"IPV4V6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
OK
```

Use the AT+CGDCONT? read command to get the current PDN connections configurations. The following is the default configuration.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IPV4V6","vzwims",,,,,0,0      Service: IMS
+CGDCONT: 2,"IPV4V6","vzwadmin",,,,,0,0     Service: ADMIN
+CGDCONT: 3,"IPV4V6","vzwinternet",,,,,0,0  Service: INTERNET
+CGDCONT: 4,"IPV4V6","vzwapp",,,,,0,0      Unused
+CGDCONT: 5,"IPV4V6","vzw800",,,,,0,0      Unused
+CGDCONT: 6,"IPV4V6","vzwclass6",,,,,0,0   Service: INTERNET
OK
```

For PDN connections identified by <cid>=1 and <cid>=2, PDP Type and APN name cannot be changed. Refer to the following example.

Enable error report in verbose format.

**AT+CMEE=2**

OK

Try to change PDP Type and APN name

**AT+CGDCONT=1,"IP","Access\_Point\_Name"**

+CME ERROR: operation not allowed

Try to change PDP Type and APN name

**AT+CGDCONT=2,"IP","Access\_Point\_Name"**

+CME ERROR: operation not allowed

For more information about the socket/PDN context binding, refer to chapter 6.1.2.1.2.

### 5.2.2. Platform Version ID 23

Use +CGDCONT command to define a PDN connection according to the note II of chapter 5.1. To get information on the command syntax and its parameters refer to document [11].

**AT+CGDCONT=[<cid>[,<PDP\_type>[,<APN>]]]**

#### 5.2.2.1. Modules: LE866-SV1, ME866A1-NV

Use the AT+CGDCONT=? test command to get the range of the supported values. <cid> range is 1-4.

**AT+CGDCONT=?**

```
+CGDCONT: (1-4),"IP",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
+CGDCONT: (1-4),"IPV6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
+CGDCONT: (1-4),"IPV4V6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
OK
```

Use the AT+CGDCONT? read command to get the current PDN connections configuration. The following is the default configuration.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IPV4V6","vzwims",,,,,0,0,0,0,1,0   Service: IMS
+CGDCONT: 2,"IPV4V6","vzwadmin",,,,,0,0         Service: ADMIN
+CGDCONT: 3,"IPV4V6","vzwinternet",,,,,0,0     Service: INTERNET
OK
```

For PDN connections identified by <cid>=1 and <cid>=2, PDP Type and APN name cannot be changed. Refer to the following example.

Enable error report in verbose format.

**AT+CMEE=2**

OK

Try to change PDP Type and APN name

**AT+CGDCONT=1,"IP","Access\_Point\_Name"**

+CME ERROR: operation not allowed

Try to change PDP Type and APN name

**AT+CGDCONT=2,"IP","Access\_Point\_Name"**

+CME ERROR: operation not allowed

For more information about the socket/PDN connection binding, refer to chapter 6.1.2.2.1.

## 5.2.2.2. Modules: LE866A1-NA, -KK, -JS, -KS, ME866A1-NA

Use the AT+CGDCONT=? test command to get the range of the supported values.

**AT+CGDCONT=?**

+CGDCONT: .....

+CGDCONT: .....

.....

OK

Use the AT+CGDCONT? read command to get the current PDN connections configuration.

**AT+CGDCONT?**

+CGDCONT: .....

OK

Assume to use the LE866A1-NA module

**AT+CGDCONT=?**

+CGDCONT: (1-5),"IP",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)

+CGDCONT: (1-5),"IPV6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)

+CGDCONT: (1-5),"IPV4V6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)

OK

Use the AT+CGDCONT? read command to get the current PDN connections configuration.

**AT+CGDCONT?**

+CGDCONT: 1,"IPV4V6","broadband",,,,,0,0

← APN di default

OK



By default, modules LE866A1-NA, -KK, -JS, -KS, and ME866A1-NA support the Internet Service on <cid>=1.

## 5.2.3. Platform Version ID 25

Use +CGDCONT command to define a PDP context. The command syntax is:

**AT+CGDCONT=[<cid>[,<PDP\_type>[,<APN>[,<PDP\_addr>[,<d\_comp>[,<h\_comp>[,<IPv4AddrAlloc>[,<Emergency\_ind>]]]]]]]]]**

## 5.2.3.1. Modules: LE910Cx SERIES

To get parameters descriptions, refer to documents [13], according to the module in use.

Here is an example of PDP context definition.

**AT+CGDCONT= 1,"IP","Access\_Point\_Name",,,,,0,0,0,0**

OK

Use the AT+CGDCONT=? test command to know the parameters ranges of the command. Here are two examples.

Assume to use a module belonging to Platform Version ID 25. Type in the AT+CGDCONT=? test command to get the supported parameters values. The returned message shows the <cid> range: 1-24 of the PDP contexts.

#### AT+CGDCONT=?

```
+CGDCONT: (1-24),"IP",,,(0-2),(0-4),(0-1),(0-1)
+CGDCONT: (1-24),"PPP",,,(0-2),(0-4),(0-1),(0-1)
+CGDCONT: (1-24),"IPV6",,,(0-2),(0-4),(0-1),(0-1)
+CGDCONT: (1-24),"IPV4V6",,,(0-2),(0-4),(0-1),(0-1)
OK
```

### 5.3. Establish a Default EPS Bearer

This chapter lists shortly the commands used to establish a Default EPS Bearer. They will be described in the next chapters.

#### AT+CGACT

Establishes a Default EPS Bearer. If the module is not attached, it does the attach procedure and then activates the Default EPS Bearer, does not create the Network Interface. See chapters 4.3, and 16.5.1.

#### AT#SGACT

Establishes a Default EPS Bearer and opens a Network Interface to use the TCP/IP protocol running on the module, refer to chapter 5.4.

#### ATD

Establishes a Default EPS Bearer. It provides a dialup connection, PPP protocol is used to exchange data between the PC (user device) and the module, the TCP/IP stack is running on the PC, see examples in chapter 15.1.

#### AT#NCM (in mode = 2)

Establishes a Default EPS Bearer, refer to chapter 16.5.

#### AT#MBIMCFG

Use this command to select the <cid> that will be used by MBIM tool running, for example, on the Windows-PC. The MBIM tool provides a "button" to establish the connection.




---

Platform Version ID 20:  
AT#SGACT, AT+CGACT, ATD, NCM, and NBIM cannot use the same <cid> at the same time. The commands are mutually exclusive.

Platform Version ID 23 and 25:  
AT#NCM and AT#MBIMCFG are not supported.

---

### 5.4. PDP Context Activation, #SGACT Command

The #SGACT command activates/deactivates PDP contexts defined with +CGDCONT command. The #SGACT establishes a Default EPS Bearer, and opens a Network Interface

to use the TCP/IP stack running on the module. If the <cid> used by the #SGACT command is already active (as the result of the default bearer context activation procedure that can be part of the attach procedure, see next paragraph), the command opens only a Network Interface, see chapter 12.11.3.

During the attach procedure, the network automatically assigns to the module an APN, supporting some services, and an IP address. The module identifies these parameters with <cid>=1 identifier. This operation, at user level, is equivalent to enter manually the command AT+CGACT=1,1, which establishes a Default EPS Bearer, and does not create a Network Interface associated to <cid>=1. It is responsibility of the user to use the assigned address for a service (IMS, data traffic, etc.) in accordance with the assigned APN, see the example in chapter 12.11.3.

The Fig. 5 shows the concepts described above.

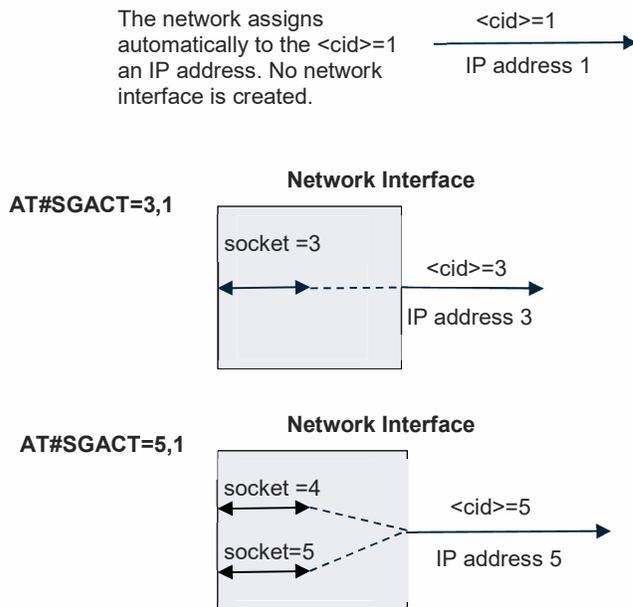


Fig. 5: Assigned APN and Network Interfaces

#### 5.4.1. Max Number of Active PDP Contexts

As stated in chapter 3 point II, the max number of the PDP context that can be defined (+CGDCONT command), and the max number of PDP contexts that can be activated at the same time (#SGACT command) depend on the module you are using.

The example in chapter 4.2.1 shows a guideline, also for 4G modules, to check the max number of PDP contexts that can be activated at the same time.

#### 5.4.2. Platform Version ID 20

The #SGACT command activates/deactivates PDP contexts defined with +CGDCONT command. To get information on the command syntax and its parameters refer to document [4].

**AT#SGACT= <cid>,<stat>[,<userId>,<pwd>]**

Use the AT#SGACT=? test command to know the parameters ranges of the command supported by the module that you are using. See next chapters.

#### 5.4.2.1. Modules: LE910 Series

Modules: LE910-EU1, B1-EU, -JN1, -NA1, B1-NA, B1-SA, -NA V2, B4-NA, -EU V2, AU V2

Type in the AT+SGACT=? test command to get the <cid> parameter values range.

```
AT#SGACT=?
#SGACT: (0-15),(0,1)
OK
```

The return message shows the <cid> range: 0-15.

- <cid>= 0 is reserved for GSM context, refer to chapter 14.  
     <cid>= 1 ÷ 15 are used for PDP contexts, see
- Tab. 13.

Type in the AT#GSMCONT=? test command to verify the GSM context configuration.

```
AT#GSMCONT=?
#GSMCONT: 0,"IP",
OK
```

#### 5.4.2.2. Modules: LE910-SV1, -SVL, SV V2

Type in the AT+SGACT=? test command to get the supported <cid> parameter values range.

```
AT#SGACT=?
#SGACT: (0-15),(0,1)
OK
```

The returned message shows the <cid> range: 0-15.

- <cid>= 0 is not used.  
     <cid>= 1-15 are used for PDP contexts, see
- Tab. 14.

#### 5.4.3. Platform Version ID 23, 25

The #SGACT command activates/deactivates PDN connection defined with +CGDCONT command. To get information on the command syntax and its parameters refer to document [11].

**AT#SGACT= <cid>,<stat>[,<userId>,<pwd>]**

Use the AT#SGACT=? test command to know the parameters ranges of the command supported by the module that you are using. See next chapter.

5.4.3.1. Modules: LE866-SV1, ME866A1-NV

Type in the AT+SGACT=? test command to get the <cid> parameter values range.

```
AT#SGACT=?  
#SGACT: (1-4),(0,1)  
OK
```

The return message shows the <cid> parameter values range: 1-4, see Tab. 15.

### 5.4.3.2. Modules: LE866A1-NA, -KK, -JS, -KS, ME866A1-NA, LE910Cx

Type in the AT+SGACT=? test command to get the <cid> parameter values range.

```
AT#SGACT=?
#SGACT: .....
OK
```

Assume to use the LE866A1-NA module.

```
AT#SGACT=?
#SGACT: (1-5),(0,1)
OK
```

## 5.5. Change a Default EPS Bearer

Following these steps, you can substitute the Default EPS Bearer, assigned automatically to the <cid=1> by the Network Operator, with a user PDP Context defined through the +CGDCONT command.

Set up the user PDP Context identified by <cid>=1.

```
AT+CGDCONT=1,"IP","Access_Point_Name"
OK
```

Check the setting of the user PDP Context.

```
AT+CGDCONT?
+CGDCONT: 1,"IP","Access_Point_Name","",0,0
OK
```

Detach the module from the network.

```
AT+CGATT=0
OK
```

Attach the module. The new PDP Context associated to <cid>=1 is activated.

```
AT+CGATT=1
OK
```

Check the dynamic parameters of the PDP context assigned by the Network Operator in response of the user defined PDP context.

```
AT+CGCONTRDP=1
+CGCONTRDP:
1,          cid
5,          bearer_id
"Access_Point_Name.....", APN
"2.192.4.65.255.0.0.0", ip_addr and subnet_mask
"2.192.4.66",      gw_addr
"10.207.43.46",   DNS_prim_addr
"0.0.0.0",       DNS_sec_addr
"0.0.0.0",       P_CSCF_prim_addr
"0.0.0.0"       P_CSCF_sec_addr
OK
```

The module saves the Default EPS Bearer parameters in NVM. At each reboot, the Protocol Stack uses them.

Now, reboot the module.

Check if the module is attached.

**AT+CGATT?**

+CGATT: 1

OK

Check the user defined PDP context.

**AT+CGDCONT?**

+CGDCONT: 1,"IP","Access\_Point\_Name","",0,0

OK

Check the dynamic parameters of the PDP context assigned by the Network Operator in response of the user defined PDP context.

**AT+CGCONTRDP=1**

+CGCONTRDP:

1,

5,

"Access\_Point\_Name.....",

"2.192.16.112.255.0.0.0",

"2.192.16.113",

"10.207.43.46",

"0.0.0.0","0.0.0.0",

"0.0.0.0"

OK

Remove the PDP context defined by the user.

**AT+CGDCONT=1**

OK

The <cid>=1 context identifier is associated again to a not defined APN (default).

**AT+CGDCONT?**

+CGDCONT: 1,"IPV4V6","", "",0,0

OK

The module saves the new configuration in NVM. At each reboot, the Protocol Stack uses it.

Now, reboot the module and verify the dynamic parameters of the PDP Context.

The <cid>=1 context identifier is automatically associated to the Default EPS Bearer assigned by the Network Operator in response of the not defined APN (see +CGDCONT? command).

**AT+CGCONTRDP=1**

+CGCONTRDP:

1,

5,

"APN\_Assigned\_by\_Network",

"10.178.48.185.255.0.0.0",

"10.178.48.186",

"213.230.129.10",

"0.0.0.0","0.0.0.0",

"0.0.0.0"

OK

**NOTE:** Platform Version ID 25:

- I. AT+CGCONTRDP format is different. Only show the allocated information from the network, refer to document [13].

## 5.6. Dedicated EPS Bearer

As stated in chapter 5, some services like VoIP, video etc. require a high level of QoS. In this case, a dedicated bearer will be established with the required QoS (can be GBR or non-GBR). Generally, the dedicated bearer can be established either by the network or by the module.

Currently, Telit modules supports only the dedicated bearer establishment started by the network side and accepted by the user through, for example, the +CGANS command.

## 5.7. Multiple PDN, #DUALAPN Command (Platform Version ID 20)

### 5.7.1. Modules: LE910 Series

Modules: LE910-EU1, B1-EU, -JN1, -NA1, B1-NA, B1-SA, -NA V2, B4-NA, -EU V2, AU V2

Suppose that the PDP contexts <cid>= 2 and <cid>=3 have the same APN and PDP type. If the module tries to activate both contexts, the second activation could fail because the second PDN Connectivity Request returns an ERROR message. This happens when the 4G Network Operator does not support multiple Default EPS bearers on a single APN. This could be a limitation for those user applications working with two different <cid> having the same APN and PDP type.

Telit has implemented the #DUALAPN command to provide a backward compatibility for the user applications mentioned previously. For example, you can assign to <cid>=3 the same APN and PDP type already in use with <cid>=2. The module routes the <cid>=3 to the <cid>=2, the result is that both <cid> have the same IP address. See the examples in the following sub-chapters. The command syntax is:

**AT#DUALAPN=<mode>[,<UNUSED\_1>[,<UNUSED\_2>[,<UNUSED\_3>]]]**

Where

< mode > - mode:

- 0 - if the module tries to activate a PDP context having the same APN and PDP type used by another PDP context already in use, there is a “routing” to the PDP context already active. The IP address is the same, use +CGPADDR command to check it. It is the default mode.
- 1 - if the module tries to activate a PDP context having the same APN and PDP type used by another PDP context already in use, the module could receive an ERROR message after the sending of the PDN Connectivity Request.



The #DUALAPN=0 mode does not allow to the commands #SGACT, +CGACT, ATD, NCM, and NBIM to use simultaneously two different <cid> having the same APN and PDP type. Only one command at a time can work with a <cid> that uses an APN and a PDP type used also by other cid. See warning note in chapter 5.3.

---

To have information on command syntax and parameters values refer to document [4].

## 5.7.2. Examples on #DUALAPN=0 Mode

### 5.7.2.1. Defined APN

At power on, assume that the PDP context identified by <cid>=1 uses "IP" and "Access\_Point\_Name", as shown below.

**AT+CGDCONT?**

+CGDCONT: 1,"IP","Access\_Point\_Name","",0,0

OK

Check if the module is registered on a 4G network.

**AT+COPS?**

+COPS: 0,0,"Network\_Operator",7 ← 7 = E-UTRAN

OK

Check if the Default EPS Bearer is active

**AT+CGCONTRDP=?**

+CGCONTRDP: (1)

OK

Check the #DUALAPN mode.

**AT#DUALAPN?**

#DUALAPN: 0,0,0 ← 0 mode, default configuration

OK

If you assign to <cid>=2 the same APN and PDP type already in use with <cid>=1, the <cid>=2 is routed to the <cid>=1, according to #DUALAPN=0 mode (default). Therefore, the IP address of both <cid> will be the same, see the following AT commands sequence.

**AT+CGDCONT=2,"IP","Access\_Point\_Name"**

OK

**AT+CGDCONT?**

+CGDCONT: 1,"IP","Access\_Point\_Name","",0,0

+CGDCONT: 2,"IP","Access\_Point\_Name","",0,0

OK

Now, only the first context is active.

**AT+CGPADDR=**

+CGPADDR: 1,"2.192.16.194"

+CGPADDR: 2,""

OK

Activate the second context.

**AT#SGACT=2,1**

#SGACT: 2.192.16.194

OK

The command returns the error code identifying the network reject cause. If no error, the code is 0.

**AT#CEERNET**

#CEERNET: 0

OK

As stated before, both <cid> have the same IP address.

**AT+CGPADDR=**

```
+CGPADDR: 1,"2.192.16.194"
+CGPADDR: 2,"2.192.16.194"
OK
```

## 5.7.2.2. Not Defined APN

At power on, assume that the PDP context identified by <cid>=1 uses a not defined APN, and a group of PDP contexts use the same APN and same PDP type as shown by the following command.

Check the current PDP contexts setting.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IPV4V6","", "",0,0
+CGDCONT: 2,"IP","Access_Point_Name","0.0.0.0",0,0
+CGDCONT: 9,"IP","Access_Point_Name","0.0.0.0",0,0
+CGDCONT: 13,"IP","Access_Point_Name","0.0.0.0",0,0
+CGDCONT: 15,"IP","Access_Point_Name","0.0.0.0",0,0
OK
```

At power on, the context identifier <cid>=1 is automatically associated to the Default EPS Bearer assigned by the Network Operator in response of the not defined APN (see +CGDCONT? command).

**AT+CGCONTRDP=1**

```
+CGCONTRDP:
1,
5,
"APN_Assigned_by_Network",
"10.178.59.124.255.0.0.0",
"10.178.59.125",
"213.230.129.10",
"0.0.0.0","0.0.0.0",
"0.0.0.0"
OK
```

Check the current binding between sockets and PDP contexts.

**AT#SCFG?**

```
#SCFG: 1,1,300,90,600,50
#SCFG: 2,9,300,90,600,50
#SCFG: 3,1,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,13,300,90,600,50
#SCFG: 6,15,300,90,600,50
OK
```

Check the #DUALAPN mode.

**AT#DUALAPN?**

```
#DUALAPN: 0,0,0,0 ← 0 mode, default configuration
OK
```

Check which are the PDP contexts active.

**AT#SGACT?**

#SGACT: 1,0  
#SGACT: 2,0  
#SGACT: 9,0  
#SGACT: 13,0  
#SGACT: 15,0  
OK

Activate PDP context <cid>=2.

**AT#SGACT=2,1**  
#SGACT: 2.192.8.164  
OK

Activate PDP context <cid>=9.

**AT#SGACT=9,1**  
#SGACT: 2.192.8.164  
OK

Activate PDP context <cid>=13.

**AT#SGACT=13,1**  
#SGACT: 2.192.8.164  
OK

Activate PDP context <cid>=15.

**AT#SGACT=15,1**  
#SGACT: 2.192.8.164  
OK

**AT+CGPADDR=**

+CGPADDR: 1,"10.178.59.124"  
+CGPADDR: 2,"2.192.8.164"  
+CGPADDR: 9,"2.192.8.164"  
+CGPADDR: 13,"2.192.8.164"  
+CGPADDR: 15,"2.192.8.164"  
OK

PDP contexts 2, 9, 13, 15 are using the same IP address, according to the #DUALAPN=0 mode.

### 5.7.3. Examples on #DUALAPN=1 Mode

#### 5.7.3.1. Defined APN

At power on, assume that the PDP context identified by <cid>=1 uses "IP" and "Access\_Point\_Name", as shown below.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IP","Access_Point_Name","",0,0
OK
```

Check if the module is registered on a 4G network.

**AT+COPS?**

```
+COPS: 0,0,"Network_Operator",7      ← 7 = E-UTRAN
OK
```

Check if the Default EPS Bearer is active

**AT+CGCONTRDP=?**

```
+CGCONTRDP: (1)
OK
```

Set #DUALAPN=1 mode.

**AT#DUALAPN=1**

```
OK
```

Assigns to <cid>=2 the APN and PDP type associated to <cid>=1 already in use.

**AT+CGDCONT=2,"IP","Access\_Point\_Name"**

```
OK
```

**AT+CGDCONT?**

```
+CGDCONT: 1,"IP","Access_Point_Name","",0,0
+CGDCONT: 2,"IP","Access_Point_Name","",0,0
OK
```

Only the first context is active.

**AT+CGPADDR=**

```
+CGPADDR: 1,"2.192.16.194"
+CGPADDR: 2,""
OK
```

Enable error report in verbose format.

**AT+CMEE=2**

```
OK
```

Activate the context <cid>=2. When a PDN Connectivity Request is sent, the command return an ERROR.

**AT#SGACT=2,1**

```
+CME ERROR: activation failed
```

The command returns the error code identifying the network reject cause.

**AT#CEERNET**

```
#CEERNET: 55
OK
```

**AT+CGPADDR=**

+CGPADDR: 1,"2.192.16.194"

+CGPADDR: 2,""

OK

## 5.7.3.2. Not Defined APN

At power on, assume that the PDP context identified by <cid>=1 uses a not defined APN.

Enable error report in verbose format.

**AT+CMEE=2**

OK

Set #DUALAPN=1 mode.

**AT#DUALAPN=1**

OK

Check the current user defined PDP contexts.

**AT+CGDCONT?**

+CGDCONT: 1,"IPV4V6","",",",0,0

+CGDCONT: 2,"IP","Access\_Point\_Name","0.0.0.0",0,0

+CGDCONT: 9,"IP","Access\_Point\_Name","0.0.0.0",0,0

+CGDCONT: 13,"IP","Access\_Point\_Name","0.0.0.0",0,0

+CGDCONT: 15,"IP","Access\_Point\_Name","0.0.0.0",0,0

OK

The context identifier <cid>=1 is automatically associated to the Default EPS Bearer assigned by the Network Operator in response of the not defined APN (see +CGDCONT? command).

**AT+CGCONTRDP=1**

+CGCONTRDP:

1,

5,

"APN\_Assigned\_by\_Network",

"10.178.47.1.255.0.0.0",

"10.178.59.125",

"213.230.129.10",

"0.0.0.0","0.0.0.0",

"0.0.0.0"

OK

Check the current binding between sockets and PDP contexts.

**AT#SCFG?**

#SCFG: 1,1,300,90,600,50

#SCFG: 2,9,300,90,600,50

#SCFG: 3,1,300,90,600,50

#SCFG: 4,2,300,90,600,50

#SCFG: 5,13,300,90,600,50

#SCFG: 6,15,300,90,600,50

OK

Activate PDP context <cid>=2.

```
AT#SGACT=2,1
#SGACT: 2.192.18.18
OK
```

Activate PDP context <cid>=9.

```
AT#SGACT=9,1
+CME ERROR: activation failed
The command returns the error code identifying the network reject cause.
```

```
AT#CEERNET
#CEERNET: 55
OK
```

Check which are the active PDP contexts. PDP context <cid>=1 is automatically activated during the attach procedure, but its right state is shown by the #SGACT command only if AT#SGACT=1,1 command is used, see the following AT commands sequence.

```
AT#SGACT?
#SGACT: 1,0
#SGACT: 2,1
#SGACT: 9,0
#SGACT: 13,0
#SGACT: 15,0
OK
```

```
AT#SGACT=1,1
#SGACT: 10.178.47.1
OK
```

```
AT#SGACT?
#SGACT: 1,1
#SGACT: 2,1
#SGACT: 9,0
#SGACT: 13,0
#SGACT: 15,0
OK
```

```
AT+CGPADDR=
+CGPADDR: 1,"10.178.47.1"
+CGPADDR: 2,"2.192.18.18"
+CGPADDR: 9,""
+CGPADDR: 13,""
+CGPADDR: 15,""
OK
```

```
AT+CGCONTRDP=2
+CGCONTRDP:
2,
6,
"Access_Point_Name.....",
"2.192.18.18.255.0.0.0",
"2.192.18.19",
"10.207.43.46",
"0.0.0.0",
"0.0.0.0",
"0.0.0.0"
```

OK

## 5.8. Authentication Setting, #PDPAUTH Command

### 5.8.1. Platform Version ID 20 and 25

The #PDPAUTH command specifies PDP authentication parameters values for a PDP context identified by the (local) context identifier <cid>. To get information on the command syntax and its parameters refer to document [4].

**AT#PDPAUTH=<cid>,<auth\_type>,[< username>,[<password >]]**

#### 5.8.1.1. Modules: LE910 and, LE910Cx SERIES

Modules: LE910-EU1, B1-EU, -JN1, -NA1, B1-NA, B1-SA, -NA V2, B4-NA, -EU V2, AU V2 and LE910Cx SERIES

At power on, assume that the context identifier <cid>=1 is assigned to a not defined APN.

**AT+CGDCONT?**

+CGDCONT: 1,"IPV4V6","", "",0,0

OK

At power on, after the attach procedure, the context identifier <cid>=1 is automatically associated to the Default EPS Bearer assigned by the Network Operator in response of the not defined APN.

**AT+CGCONTRDP=1**

+CGCONTRDP:

1,

5,

"APN\_Assigned\_by\_Network",

"100.80.28.204.255.0.0.0","100.80.28.205",

"80.201.237.239",

"0.0.0.0","0.0.0.0",

"0.0.0.0"

OK

Set up the new PDP Context associated to <cid>=1.

**AT+CGDCONT=1,"IPV4V6","Access\_Point\_Name"**

OK

Verify the entered PDP Context.

**AT+CGDCONT?**

+CGDCONT: 1,"IPV4V6","Access\_Point\_Name","",0,0

OK

Set the PDP authentication parameters on context <cid>=1.

**AT#PDPAUTH=1,1,"UserName","PassWord"**

OK

Detach the module from the network.

**AT+CGATT=0**

OK

Attach the module to the network, the new PDP Context is activated. It is the new Default EPS Bearer.

**AT+CGATT=1**

OK

Check the dynamic parameters of the PDP context assigned by the Network Operator in response of the user defined PDP context.

**AT+CGCONTRDP=1**

+CGCONTRDP:

1,

5,

"Access\_Point\_Name.....",

"10.113.0.66.255.0.0.0",

"10.113.0.67","8.8.4.4",

"0.0.0.0","0.0.0.0",

"0.0.0.0"

OK

## 6. OUTGOING SOCKETS IN ONLINE MODE

### 6.1. Sockets Configuration, #SCFG Command

Use the #SCFG command to configure a socket belonging to the Multi-socket environment, the <connId> parameter identifies the socket. The Multi-socket environment provides N sockets, the N value depends on the module you are using. The configuration is saved in NVM, the command syntax is:

**AT#SCFG = <connId>, <cid>, <pktSz>, <maxTo>, <connTo>, <txTo>**

To get parameters descriptions refer to documents [1], [2], [4], [11], or [13] according to the used module.

#### 6.1.1. 2G/3G Modules

Tab. 4 shows the default sockets/contexts binding returned by the AT#SCFG? read command. The socket configuration cannot be modified when the socket is open.

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
1	bound	/	/	/	/
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	bound	/	/	/
5	/	bound	/	/	/
6	/	bound	/	/	/

Tab. 4: 2G/3G Default Sockets/Contexts Binding Configuration

#### Example

Starting from the default configuration, bind the socket <connId>=2 to the PDP context <cid>=3.

**AT#SCFG = 2, 3, 300, 90, 600, 50**

OK

Check the new sockets/contexts binding.

**AT#SCFG?**

#SCFG: 1,1,300,90,600,50

#SCFG: 2,3,300,90,600,50

#SCFG: 3,1,300,90,600,50

#SCFG: 4,2,300,90,600,50

#SCFG: 5,2,300,90,600,50

#SCFG: 6,2,300,90,600,50

OK

Tab. 5 shows the new sockets/contexts binding configuration.

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
1	bound	/	/	/	/
2	/	/	bound	/	/
3	bound	/	/	/	/
4	/	bound	/	/	/
5	/	bound	/	/	/
6	/	bound	/	/	/

Tab. 5: 2G/3G Sockets/Contexts Binding Configuration

## 6.1.2. 4G Modules

### 6.1.2.1. Platform Version ID 20

#### 6.1.2.1.1. Modules: LE910 Series

Modules: LE910-EU1, B1-EU, -JN1, -NA1, B1-NA, B1-SA, -NA V2, B4-NA, -EU V2, AU V2

Use the AT#SCFG? read command to get the current socket/cid configurations. The following is the default configuration.

#### AT#SCFG?

AT#SCFG?

#SCFG: 1,1,300,90,600,50

#SCFG: 2,1,300,90,600,50

#SCFG: 3,1,300,90,600,50

#SCFG: 4,2,300,90,600,50

#SCFG: 5,2,300,90,600,50

#SCFG: 6,2,300,90,600,50

#SCFG: 7,1,300,90,600,50

#SCFG: 8,1,300,90,600,50

#SCFG: 9,2,300,90,600,50

#SCFG: 10,2,300,90,600,50

OK

Use the AT#SCFG=? test command to get the range of the supported values. <cid> range is 0-15.

#### AT#SCFG=?

#SCFG: (1-10),(0-15),(0-1500),(0-65535),(10-1200),(0-264)

OK

Use the AT+CGDCONT=? test command to get the range of the supported values. <cid> range, identifying PDP contexts, is 1-15.

#### AT+CGDCONT=?

+CGDCONT: (1-15),"IP",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)

+CGDCONT: (1-15),"IPV6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)

+CGDCONT: (1-15),"IPV4V6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)

OK

Use the AT+CGDCONT? read command to get the current PDP contexts configurations. The following is the default configuration.

```
AT+CGDCONT?
+CGDCONT: 1,"IPV4V6","", "",0,0
OK
```

Tab. 6 summarizes how <cid> and sockets can be used.

Sockets	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>= 5-15
<b>Internal Sockets (not directly accessible to the user)</b>	During the attach procedure, the network assigns to the module the Default EPS Bearer that supports the service provided by the used Network Operator. By default, is used <cid>=1 bound to an internal socket. See chapter 5.1.				
<b>IP Easy Multi-Socket &lt;connid&gt;</b>	The user can bind all <cid> to sockets belonging to the Multi-socket environment, at the same time.				
1-10					

Tab. 6: LE910-EU1, Network Operator Service & Multi-Socket

#### 6.1.2.1.2. Modules: LE910-SV1, -SVL, SV V2

Use the AT#SCFG? read command to get the current socket/cid configurations. The following is the default configuration.

```
AT#SCFG?
#SCFG: 1,3,300,90,600,50
#SCFG: 2,3,300,90,600,50
#SCFG: 3,3,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,2,300,90,600,50
#SCFG: 7,3,300,90,600,50
#SCFG: 8,3,300,90,600,50
#SCFG: 9,3,300,90,600,50
#SCFG: 10,3,300,90,600,50
OK
```

Use the AT#SCFG=? test command to get the range of the supported values. <cid> range is 0-15

```
AT#SCFG=?
#SCFG: (1-10),(0-15),(0-1500),(0-65535),(10-1200),(0-264)
OK
```

Use the AT+CGDCONT=? test command to get the range of the supported values. <cid> range is 1-6.

**AT+CGDCONT=?**

```
+CGDCONT: (1-6),"IP",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
+CGDCONT: (1-6),"IPV6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
+CGDCONT: (1-6),"IPV4V6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
OK
```

Use the AT+CGDCONT? read command to get the current PDN contexts configurations. The following is the default configuration.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IPV4V6","vzwims",,,,,0,0      Service: IMS
+CGDCONT: 2,"IPV4V6","vzwadmin",,,,,0,0    Service: ADMIN
+CGDCONT: 3,"IPV4V6","vzwinternet",,,,,0,0 Service: INTERNET
+CGDCONT: 4,"IPV4V6","vzwapp",,,,,0,0      Unused
+CGDCONT: 5,"IPV4V6","vzw800",,,,,0,0     Unused
+CGDCONT: 6,"IPV4V6","vzwclass6",,,,,0,0   Service: INTERNET
OK
```

Tab. 7 summarizes how <cid> and sockets can be used.

Sockets	<cid>=1	<cid>=2	<cid>=3	<cid>=4/5	<cid>= 6	<cid>= 7-15
<b>Internal Sockets (not directly accessible to the user)</b>	By default, <cid>=1 is used for IMS Service assigned by the Network Operator and is bound to an internal socket.	By default, <cid>=2 is used for ADMIN Service assigned by the Network Operator and is bound to an internal socket.	By default, <cid>=3 is used for INTERNET Service assigned by the Network Operator and is bound to an internal socket.	Unused	By default, <cid>=6 is used for INTERNET Service assigned by the Network Operator and is bound to an internal socket.	
<b>IP Easy Multi-Socket &lt;connId&gt;</b>	The user can bind all <cid> to sockets belonging to the Multi-socket environment, at the same time.					
1-10						

Tab. 7: LE910-SV1, IMS, ADMIN, INTERNET Services & Multi-Socket

### 6.1.2.2. Platform Version ID 23

#### 6.1.2.2.1. Module: LE866-SV1, ME866A1-NV

Use the AT#SCFG? read command to get the current socket/cid configurations. The following is the default configuration.

```

AT#SCFG?
#SCFG: 1,3,300,90,600,50
#SCFG: 2,3,300,90,600,50
#SCFG: 3,3,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,2,300,90,600,50
OK

```

Use the AT#SCFG=? test command to get the range of the supported values. <cid> range is 1-4.

```

AT#SCFG=?
#SCFG: (1-6),(1-4),(0-1500),(0-65535),(10-1200),(0-264)
OK

```

Use the AT+CGDCONT=? test command to get the range of the supported values. <cid> range is 1-4.

```

AT+CGDCONT=?
+CGDCONT: (1-4),"IP",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
+CGDCONT: (1-4),"IPV6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
+CGDCONT: (1-4),"IPV4V6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
OK

```

Use the AT+CGDCONT? read command to get the current PDN connections configuration. The following is the default configuration.

```

AT+CGDCONT?
+CGDCONT: 1,"IPV4V6","vzwims",,,,0,0,0,0,1,0   Service: IMS
+CGDCONT: 2,"IPV4V6","vzwadmin",,,,0,0,0,0   Service: ADMIN
+CGDCONT: 3,"IPV4V6","vzwinternet",,,,0,0,0,0 Service: INTERNET
OK

```

Tab. 8 summarizes how <cid> and sockets can be used.

Sockets	<cid>=1	<cid>=2	<cid>=3	<cid>=4
<b>Internal Sockets (not directly accessible to the user)</b>	By default, <cid>=1 is used for IMS Service assigned by the Network Operator, and is bound to a dedicated internal socket.	By default, <cid>=2 is used for ADMIN Service assigned by the Network Operator, and is bound to a dedicated internal socket.	By default, <cid>=3 is used for INTERNET Service assigned by the Network Operator, and is bound to a dedicated internal socket.	
<b>IP Easy Multi-Socket &lt;connId&gt;</b>	The user can bind all <cid> to sockets belonging to the Multi-socket environment, at the same time.			
1-6				

Tab. 8: LE866-SV1, IMS, ADMIN, INTERNET Services & Multi-Socket

## 6.2. Open a Socket, #SD Command

The #SD command (Socket Dial) opens the TCP/UDP connection towards the host. If required, DNS query is done to resolve the IP address. To open the remote connection, the PDP context to which the <connId> is associated must be active, otherwise the command returns an ERROR message. The command syntax is:

**AT#SD=<connId>,<txProt>, <rPort>, <IPAddr> [, <closureType> [, <IPort>[,<connMode>]]]**

To get information on the command syntax and its parameters refer to document [1], [2], [4], [11], and [13] according to the used module.

### 6.2.1. Example on Opening & Suspension, (2G/3G)

This example shows the opening and suspension of the connections <connId>=3, and 2.

Tab. 9 shows the starting sockets/contexts configuration:

- PDP context <cid>=1, and 3 active
- Sockets <connId>=1, and 3 bound to <cid>=1 (default)
- Socket <connId>=2 bound to <cid>=3
- Sockets <connId>=4, 5, and 6 bound to <cid>=2 (default)

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	No active	Active	No active	No active
1	bound	/	/	/	/
2	/	/	bound	/	/
3	bound	/	/	/	/
4	/	bound	/	/	/
5	/	bound	/	/	/
6	/	bound	/	/	/

Tab. 9: Sockets/Contexts Binding and PDPs Active

Open the connection in ONLINE mode.

**AT#SD = 3,0,80,"Host\_Name\_1"**

CONNECT

... data exchange ...

**+++** ← suspend the connection.

OK ← connection 3 has been successfully suspended.

After the CONNECT message, the module is in ONLINE mode. Issue the escape sequence (+++) to suspend the connection and enter the COMMAND mode, the escape sequence does not close the connection.

If an escape sequence is included in the data to be sent, at the beginning of the sending, you must use the #SKIPESC command to guarantee that the connection is not suspended.

See the ATS12 command to set timing relevant to escape sequence. Set the guard time before and after escape sequence.

One more socket connection can be established.

**AT#SD = 2, 0, 80,"Host\_Name\_2"**

CONNECT

... data exchange ...

**+++** ← suspend the connection.

OK ← connection 2 has been successfully suspended.

The module is again in COMMAND mode.

If a suspended connection receives data, the DTE displays an unsolicited SRING: <connId> indication. The unsolicited SRING indication appears only in COMMAND mode. For example, if Host\_Name\_1 sends data, the DTE displays the following URC:

SRING: 3 ← 3 is the number of the <connId> with pending data.

Tab. 10 shows the sockets/contexts configuration after the execution of the #SD commands.

- PDP contexts <cid>= 1, and 3 active
- Sockets <connId>= 4, 5, and 6 bound to <cid>=2
- Socket <connId>=1 bound to <cid>=1
- Socket <connId>=2, bound to <cid>=3, and open
- Socket <connId>=3, bound to <cid>=1, and open

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	No active	Active	No active	No active
1	bound	/	/	/	/
2	/	/	open	/	/
3	open	/	/	/	/
4	/	bound	/	/	/
5	/	bound	/	/	/
6	/	bound	/	/	/

Tab. 10: Sockets/Contexts Binding and Sockets Opened

### 6.3. Pending Data, #PADCMD, #PADFWD Commands

The following two AT commands enable the flushing of the pending data on the local socket toward the remote side.

**AT#PADCMD=1**

OK

Set the ASCII code of the character that will trigger the flushing of the pending data. In this example is "A" = 65 (decimal value).

**AT#PADFWD=65**

OK

Open the connection.

**AT#SD = 3, 0, 80, "Host Name 1"**  
CONNECT

.....

The data, pending on the socket, are sent to the remote side when one of the three conditions is verified:

- data size reaches <pktSz> (see #SCFG command), or
- <txTo> timeout is over (see #SCFG command), or
- "A" character (in this example) is sent. Also the character "A" is sent to the remote side.

.....

.....

#### 6.4. Resume a Connection, #SO Command

Use #SO command to resume a suspended connection. The command syntax is:

**AT#SO= <connId>**

To get parameters descriptions refer to documents [1], [2], [4], [11], or [13] according to the used module.

##### Example

**AT#SD = 2,0,80,"Host\_Name"**

CONNECT ← the connection is open.

... data exchange ...

**+++** ← suspend the connection.

OK ← the module is in COMMAND mode.

Data are pending on socket.

SRING: 2

Resume the suspended connection.

**AT#SO = 2**

CONNECT

... pending data are displayed ...

.....

.....

**+++** ← suspend the connection.

OK

You can resume a suspended connection without waiting for SRING unsolicited indication. In this case, the module returns the CONNECT message. If the connection is not open the returned message is NO CARRIER.

## 6.5. Close a Connection, #SH, #SLASTCLOSURE Commands

Use #SH command to close a socket connection. The command returns the OK message if the connection is already closed. The command syntax is:

**AT#SH= <connId>**

To get parameters descriptions, refer to documents [1], [2], [4], [11], or [13] according to the used module.

### Example

Open a socket connection.

**AT#SD = 2,0,80,"Host\_Name"**

CONNECT ← the connection is open

... data exchange ...

**+++** ← suspend the connection.

OK ← the module is in COMMAND mode

Type in #SH command to close the socket connection.

**AT#SH = 2**

OK ← the connection is closed.

A connection can be closed by the following events:

- Remote host closes the connection.
- Socket inactivity timeout is expired.
- DTE issues the escape sequence and #SH command.
- Network deactivates the connection.

Use #SLASTCLOSURE command to get the socket disconnection cause.

**AT#SLASTCLOSURE=[<connId>]**

#SLASTCLOSURE: <connId>,<cause> ← response format.

To get parameters descriptions, refer to documents [1], [2], [4], [11], or [13] according to the used module.

## 6.6. TCP/IP Options Configuration

### 6.6.1. Reassemble TCP Packets, #TCPREASS Command (2G)

Use #TCPREASS command to enable the reassembling of the fragmented TCP packets. The command syntax is:

**AT#TCPREASS=<n>**

Refer to document [1] to get parameters descriptions.

### 6.6.2. Max TCP Payload Size, #TCPMAXDAT Command (2G)

Use #TCPMAXDAT command to configure the maximum TCP/IP payload size accepted in one single TCP/IP datagram from the module. The <size> is sent by the module to the peer when the socket connection is opened.

**AT#TCPMAXDAT=<size>**

Refer to document [1] to get parameters descriptions. Here is an example:

The module advises the peer that it does not accept TCP/IP datagrams with a payload longer than 1000 bytes.

**AT#TCPMAXDAT=1000**  
OK

### 6.6.3. #BASE64 Command

The #BASE64 command enables base64 encoding/decoding of data sent/received on a socket. The command syntax is:

**AT#BASE64=<connId>,<enc>,<dec>[,<unused\_B >[,<unused\_C >]]**

To get parameters descriptions, refer to documents [1], [2], [4], or [13] according to the used module.



#BASE64 command is not supported by Platform Version ID 23.

---

#### Example

Skip the escape sequence, its transmission is not enabled.

**AT#SKIPESC=1**  
OK

Open the connection.

**AT#SD=1,0,<Remote Host Port>,"IP Address"**  
CONNECT

... data exchange ...

**+++** ← suspend the connection.

OK ← the module is in COMMAND mode

Encode all data coming from serial port and sent to the socket.

**AT#BASE64=1,1,0**  
OK

Resume the connection.

**AT#SO=1**  
CONNECT

.....  
... data received from serial port are base64 encoded and sent to the socket. ...  
.....

**+++** ← suspend the connection.

OK ← the module is in COMMAND mode

Decode all data coming from the socket and sent to the serial line.

**AT#BASE64=1,0,1**  
OK

Resume the connection.

**AT#SO=1**  
CONNECT

.....  
... data received from socket are base64 decoded and sent to the serial port. ...  
.....

**+++** ← suspend the connection.

OK ← the module is in COMMAND mode

## 7. INCOMING SOCKETS IN ONLINE MODE

Use the #SL (Socket Listening) command to open a socket in listening mode for an incoming TCP connection.

- You can configure up to N sockets in listening mode. The N value depends on the module you are using.

Two sockets in listening mode cannot bound to the same PDP context (same IP address) and using the same port. To use the same port, the two sockets must be bound to two different PDP contexts (different IP addresses), refer to

- Tab. 11.

When a remote host tries to connect, the module sends to the DTE the +SRING: <connId> unsolicited indication. The user can accept (#SA) or refuse (#SH) the incoming connection. The command syntax is:

**AT#SL= <connId>, <listenState>, <listenPort>[, <closure type>]**

Refer to documents [1], [2], [4], [11], or [13] according to the used module, to get parameters descriptions.

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	No active	No Active	Active	No active
1	listen, port 1	/	/	/	/
2	/	/	bound	/	/
3	listen, port 2	/	/	/	/
4	bound	/	/	/	/
5	/	/	/	listen, port 2	/
6	/	bound	/	/	/

Tab. 11: An Example of Sockets in Listening Mode

### Example

Open the <connId> = 2 socket in listening mode on <port> = 6543.

**AT#SL = 2,1,6543**

OK

Now, if a remote host is trying to connect, the module receives a SRING unsolicited indication with the listening <connId>:

SRING: 2



If the incoming connection request is rejected, the listening socket will be closed. Use #SL command to reopen the socket.



## 7.1. Accept an Incoming Connection, #SA Command

Use the #SA command without <connMode> parameter to accept the incoming connection, notified by the SRING unsolicited indication, in ONLINE mode. The command syntax is:

**AT#SA=<connId>[,<connMode>]**

Refer to documents [1], [2], [4], [11], or [13] according to the used module, to get parameters descriptions.

### Example

Open the socket connection <connId> = 3 in listening mode (only for TCP) on <ListenPort> = xxxx.

```
AT#SL = 3, 1, xxxx
OK
```

Now, if a remote host tries to connect, the module receives a SRING unsolicited indication with the listening <connId>:

```
SRING: 3          ← a remote host is trying to connect.
```

Accept the incoming connection <connId>=3 in ONLINE mode.

```
AT#SA = 3
CONNECT          ← the connection is accepted.
... exchange data ...
```

The module is in ONLINE mode, the connection is established, and the two hosts can exchange data. With the escape sequence (+++) the connection can be suspended, and the module is back to COMMAND mode. Use #SO command to resume the suspended connection, refer to chapter 6.4.

To accept automatically incoming connection use #SCFGEXT command, refer to chapter 8.3.1.

## 7.2. Check the Socket State, #SS Command

Use the #SS command to get the current socket state. The command syntax is:

**AT#SS[=<connId>]**

Here is the response message format of the command.

```
#SS: <connId>,<state>,<locIP>,<locPort>,<remIP>,<remPort>
```

Refer to documents [1], [2], [4], [11], or [13] according to the used module, to get parameters descriptions.

## Example

Check the state of all available sockets. In this case, the <connId> parameter is not used.

### AT#SS

```
#SS: 1,4,217.201.131.110,21
#SS: 2,2,217.201.131.110,1033,194.185.15.73,10510
#SS: 3,3,217.201.131.110,1034,194.185.15.73,10510
#SS: 4,1,217.201.131.110,1035,194.185.15.73,10510
#SS: 5,0
#SS: 6,0
OK
```

By issuing AT#SS=<connId> the command return only the state of the <connId> socket.

## 7.3. Open UDP Connection, #SLUDP Command

Use #SLUDP command to open a socket listening for an incoming UDP connection on a specified port. The command syntax is:

**AT#SLUDP=<connId>, <listenState>, <listenPort>**

Refer to documents [1], [2], [4], [11], or [13] according to the used module, to get parameters descriptions.

Also in this case, it is possible to receive SRING unsolicited and decide to accept (#SA) or refuse (#SH) the connection. To accept automatically incoming connection use #SCFGEXT command, refer to chapter 8.3.1.

## 7.4. Firewall Setting, #FRWL Command

2G/3G/4G modules provide an internal firewall that controls the access of the incoming connections. DROP is the firewall policy, all packets not matching the rules are silently discarded. The firewall applies for incoming (listening) connections. Outgoing connections are established regardless of the firewall settings. Use the #FRWL command to add or remove an ACCEPT chain to/from internal firewall. The command syntax is:

**AT#FRWL[=<action>,<ip\_address>,<net mask>]**

When a packet comes in from the <incoming IP> address, the firewall rules chain is scanned to match the following criteria:

<incoming IP> & <net mask> = <ip\_address> & <net mask>

If matching is found, the packet is accepted and the rule scan is finished, otherwise the next chain is considered until the end of the rules. If no matching is found the packet is silently dropped.



#FRWL command is not enabled by default.

Refer to documents [1], [2], [4], [11], or [13] according to the used module, to get parameters descriptions.

### Example

Assume to accept connections only from devices having the IP addresses ranging from 197.158.1.1 to 197.158.255.255. The following ACCEPT chain must be added to the firewall.

```
AT#FRWL=1,"197.158.1.1","255.255.0.0"
OK
```

## 7.5. Examples

### 7.5.1. Modules in Server and Client Configurations

This example describes a remote connection between two modules. The first one is configured as a server. It opens a socket connection in listen mode (#SL). The second one is configured as a client, opens a socket dial connection (#SD) toward the server. Each module is connected to a terminal emulator. See also the example described in chapter 14.3.6, just for comparison.

#### Server side

Check the sockets/PDP contexts binding configuration (default configuration).

```
AT#SCFG?
#SCFG: 1,1,300,90,600,50
#SCFG: 2,1,300,90,600,50
#SCFG: 3,1,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,2,300,90,600,50
OK
```

Assume that only PDP context <cid>=1 has been set, all other contexts have been removed.

```
AT+CGDCONT?
+CGDCONT: 1,"IP","Access_Point_Nam ","0.0.0.0",0,0
OK
```

Activate the PDP context <cid>=1. The command returns the IP address assigned by the network to the module. Assume that no UserID and Password are needed.

```
AT#SGACT=1,1
#SGACT: 2.192.9.179
OK
```

The client to open the connection will use this IP address, see client side.

Define an ACCEPT firewall chain to accept the IP address of the client; see client side.

```
AT#FRWL=1,"37.176.239.214","0.0.0.0"
```

```
OK
```

Open <connId>=1 socket in listening mode on <port>=1024. The client will use it.

```
AT#SL=1,1,1024
```

```
OK
```

When the client opens the connection, the server displays on the terminal emulator a SRING unsolicited indication with <connId>=1:

```
SRING: 1
```

Accept the connection 1.

```
AT#SA=1
```

```
CONNECT
```

```
... exchange data ...
```

```
NO CARRIER
```

← on the client side has been entered #SH command, the connection is closed.

### Client side

Check the sockets/PDP contexts binding configuration (default configuration).

```
AT#SCFG?
```

```
#SCFG: 1,1,300,90,600,50
```

```
#SCFG: 2,1,300,90,600,50
```

```
#SCFG: 3,1,300,90,600,50
```

```
#SCFG: 4,2,300,90,600,50
```

```
#SCFG: 5,2,300,90,600,50
```

```
#SCFG: 6,2,300,90,600,50
```

```
OK
```

Assume that only PDP context <cid>=1 has been set, all other contexts have been removed.

```
AT+CGDCONT?
```

```
+CGDCONT: 1,"IP","Access_Point_Nam ","0.0.0.0",0,0
```

```
OK
```

Activate the PDP Context <cid>=1, assume that no UserID and Password are needed. The command returns the IP address assigned by the network to the module.

```
AT#SGACT=1,1
```

```
#SGACT: 37.176.239.214
```

```
OK
```

The server will use this IP address to set the firewall, see server side.

Open the connection toward the server using its IP address and port 1024, see sever side.

```
AT#SD=1,0,1024," 2.192.9.179"
```

```
CONNECT
```

... exchange data ...

**+++** ← suspend the connection.

OK ← the module is in COMMAND mode

**AT#SH=1** ← close the connection. On server side is displayed the NO CARRIER message.

OK

Use #CGPADDR= command to verify the IP address assigned by the network to the modules after the PDP contexts activations; refer to chapter 4.4.

## 8. SOCKET CONNECTION IN COMMAND MODE

The #SD (Socket Dial) and #SA (Accept connection) commands are introduced respectively in chapter 6.2 and 7.1, and their descriptions are focused on the ONLINE mode connection. In this case, the optional parameter <connMode> is set to 0 (default).

The present section describes the use of the two commands with the parameter <connMode> set to 1. The #SD command with <connMode> set to 1 opens a connection, leaves the module in COMMAND mode, and allows the user to enter AT commands even when the socket connection is open. If data are received on the connection, a SRING unsolicited message is displayed on DTE.

To get parameters descriptions of the commands described in the following chapters, refer to documents [1], [2], [4], [11], or [13] according to the used module,

### 8.1. Open a Socket, #SD Command

Use #SD command with <connMode>=1 to open a connection in COMMAND mode. <connMode> is the last parameter in the syntax command.

**AT#SD=<connId>,<txProt>,<rPort>,<IPAddr>[,<closure type>[,<IPort>[,<1>]]]**

### 8.2. Accept an Incoming Connection, #SA Command

Use #SA command with <connMode>=1 to accept the connection in COMMAND mode. <connMode> is the last parameter in the syntax command.

**AT#SA = <connId>[,<1>]**

Here is the unsolicited indication notifying the user that pending data are available on <connId> connection. For examples, refer to chapter 8.7.

SRING: <connId>.

### 8.3. Extended Socket Parameters Configuration

#### 8.3.1. #SCFGEXT Command

#SCFGEXT command configures the set of the extended parameters for each socket provided by the Multi-socket environment; the values are saved in NVM. The command syntax is:

**AT#SCFGEXT=<connId>,<srMode>,<recvDataMode>,  
<keepalive>[,<ListenAutoRsp>[,<sendDataMode>] ]**

Here is the <srMode> range. For examples, refer to chapter 8.7.

- 0 – Normal (default)
- 1 – Data amount
- 2 – Data view
- 3 – Data view with UDP datagram information

### Example

Use the test command to get the range of the supported values. Pay attention to:  
<connId> range: 1-6, <srMode> range: 0-3.

**AT#SCFGEXT=?**

#SCFGEXT: (1-6),(0-3),(0,1),(0-240),(0,1),(0,1)

OK

Use the read command to get the current socket extended configuration.

**AT#SCFGEXT?**

#SCFGEXT: 1,0,0,0,0,0

#SCFGEXT: 2,0,0,0,0,0

#SCFGEXT: 3,0,0,0,0,0

#SCFGEXT: 4,0,0,0,0,0

#SCFGEXT: 5,0,0,0,0,0

#SCFGEXT: 6,0,0,0,0,0

OK

Set SRING Data amount mode on socket 1

**AT#SCFGEXT = 1,1,0,0**

OK

Set SRING Data view mode on socket 2

**AT#SCFGEXT = 2,2,1,0,0,1**

OK

Set SRING Data view UDP info mode on socket 6

**AT#SCFGEXT = 6,3,1,0**

OK

Check the current socket extended configuration.

**AT#SCFGEXT?**

#SCFGEXT: 1,1,0,0,0,0

#SCFGEXT: 2,2,1,0,0,1

#SCFGEXT: 3,0,0,0,0,0

#SCFGEXT: 4,0,0,0,0,0

#SCFGEXT: 5,0,0,0,0,0

#SCFGEXT: 6,3,1,0,0,0

OK

### 8.3.2. #SCFGEXT2 Command

Use the # SCFGEXT2 command to configure additional extended parameters. Here is the command syntax for each module technology.

#### 8.3.2.1. 2G modules

**AT#SCFGEXT2=<connId>,<bufferStart>[,<abortConnAttempt>  
[,<sringLen >[,<sringTo >[,<noCarrierMode>]]]]**

Refer to documents [1].

#### 8.3.2.2. 3G/4G modules

**AT#SCFGEXT2=<connId>,<bufferStart>[,<abortConnAttempt>  
[,<unused\_B >[,<unused\_C >[,< noCarrierMode >]]]]**

To get parameters descriptions, refer to documents [2], [4], [11], or [13] according to the used module.

## 8.4. Send Data, #SSEND and #SENDEXT Commands

Use #SSEND command to send data on a connection when the module is in COMMAND mode. When the <CR> is entered to close the entering of the command, the ">" prompt appears to indicate that the command is ready to accept the data to be sent. Enter Ctrl-Z to close the data entering and send the data. Before using the command, the socket must be opened. The command syntax is:

**AT#SSEND=<connId>**

Use #SENDEXT command to include all bytes (0x00 to 0xFF) in the block of data to send. This command allows to include special characters as ESC (0x1B), Ctrl-Z (0x1A), BS (0x08) not accepted by #SSEND. The command syntax is:

**AT#SENDEXT=<connId>,<bytestosend>**

To get parameters descriptions, refer to documents [1], [2], [4], [11], or [13] according to the used module. For examples refer to chapter 8.7.

## 8.5. Receive Data, #SRECV Command

The module is in COMMAND mode and assume to have received a SRING unsolicited indication notifying that received data are pending in the socket. Use #SRECV command to get the pending data in the socket buffer. The command syntax is:

**AT#SRECV=<connId>,<maxByte>[,<UDPinfo>**

To get parameters descriptions, refer to documents [1], [2], [4], [11], or [13] according to the used module. For examples refer to chapter 8.7.

Example

Assume to have received a SRING Data amount unsolicited indication, see #SCFGEXT command.

SRING: 1,5

Extract all the five bytes.

**AT#SRECV=1,5**

#SRECV: 1,5

hello

← here are the five bytes, as example: hello.

OK

## 8.6. Socket Information, #SI Command

Use #SI command to get additional information on all sockets or on the selected socket. The command syntax is:

**AT#SI[= <connId>]**

The format of the command response is:

#SI: <connId>,<sent>,<received>,<buff\_in>,<ack\_waiting>

**NOTE:** LE866 Series does not support <ack\_waiting> parameter?

Example

Get information on all the available sockets.

**AT#SI**

#SI: 1,123,400,10,50

#SI: 2,0,100,0,0

#SI: 3,589,100,10,100

#SI: 4,0,0,0,0

#SI: 5,0,0,0,0

#SI: 6,0,98,60,0

OK

## 8.7. Examples

### 8.7.1. Switching from COMMAND to ONLINE Mode and Back

Assume that the SRING unsolicited mode has been configured in Data amount mode through the #SCFGEXT command. The socket <connId>=1 has been configured, and the used PDP context has been configured and activated. Enable the error report in verbose format (+CMEE=2).

Use #SO command to switch from COMMAND to ONLINE mode, and +++ escape sequence to go back in COMMAND mode.

Open socket connection <connId>=1 in COMMAND mode on echo port.

**AT#SD=1,0,echo\_port,"address",0,0,1**

OK ← the socket connection is open; the module is in COMMAND mode.

Send data on the socket. Text view mode for received data (#SCFGEXT).

**AT#SSEND=1**

> **hello**Ctrl-Z ← type in the first message.

OK

SRING: 1,5 ← SRING in Data amount mode, 1 is the connection Id, 5 is the amount of the echoed data (**hello**).

**AT#SO = 1** ← resume the suspended connection.

CONNECT

**hello** ← echoed data pending on the socket are sent to the terminal emulator.

Now, the AT interface is in ONLINE mode. All the entered characters are interpreted as data to send on the connection. The remote side returns the sent message (echo).

**+++** ← suspend the connection. The escape sequence is not displayed on the terminal emulator.

OK ← the module is in COMMAND mode.

SRING: 1,3 ← SRING in Data amount mode, 1 is the connection Id, 3 is the amount of echoed data (+++).

Now, close the socket connection.

**AT#SH=1**

OK

If the socket connection is not closed, its time out expires, and the NO CARRIER message is displayed on the terminal emulator.

### 8.7.2. Accept a Socket Connection in COMMAND Mode

Open the socket connection <connId> = 1 in listening mode (only for TCP) on <port> = xxxx.

**AT#SL = 1,1,xxxx**

OK

Now, if a remote host tries to connect, the module receives a SRING unsolicited indication with the listening <connId>=1.

SRING: 1

Accept the incoming connection in COMMAND mode.

**AT#SA = 1,1**

OK ← the connection is accepted. The command returns OK message if the connection is successful. The module stays in COMMAND mode. When <connMode>=0 (default) the #SA command returns the CONNET message, in this case the module is in ONLINE mode.

Check the sockets state.

**AT#SS**

#SS: 1,2,loc\_ip,loc\_port,rem\_ip,rem\_port ← <connId>=1 is open in suspended state.

#SS: 2,0

#SS: 3,0

#SS: 4,0

#SS: 5,0

#SS: 6,0

OK

### 8.7.3. SRING Unsolicited Mode: Normal

Assume that the SRING unsolicited mode has been configured in Normal mode through the #SCFGEXT command, the socket <connId>=2 has been configured, and the PDP context has been configured and activated. Enable the error report in verbose format (+CMEE=2).

Open socket connection <connId>=2 in COMMAND mode on echo port.

**AT#SD=2,0,echo\_port,"address",0,0,1**

OK ← the socket connection is open; the module is in COMMAND mode.

Send data on the socket. Text view mode for received data (#SCFGEXT).

**AT#SEND=2**

> helloCtrl-Z ← type in the first message.

OK

SRING: 2 ← SRING in Normal mode, 2 is the socket connection Id. Only one SRING unsolicited indication is received, regardless if you must send other data.

**AT#SEND=2**

> helloCtrl-Z ← type the second message.

OK

...

Close the socket connection.

**AT#SH=2**

OK

#### 8.7.4. SRING Unsolicited Mode: Data amount

Assume that the SRING unsolicited mode has been configured in Data amount mode through the #SCFGEXT command, the socket <connId>=2 has been configured, and the PDP context has been configured and activated. Enable the error report in verbose format (+CMEE=2).

Open socket connection <connId>=2 in COMMAND mode on echo port.

**AT#SD=2,0,echo\_port,"address",0,0,1**

OK ← socket connection is open in COMMAND mode.

**AT#SEND=2** ← send data.

> helloCtrl-Z ← type in the first message

OK

SRING: 2,5 ← SRING in Data amount mode, 2 is the connection Id, 5 is the amount of the echoed data.

**AT#SEND=2**

> helloCtrl-Z ← type the second message

OK

SRING: 2,10 ← SRING in Data amount mode, unsolicited indication is updated every time new data arrives on the socket.

**AT#SI=2** ← get socket info.

#SI: 2,10,0,10,0 ← ten bytes sent, and ten pending on the socket.

OK

**AT#SO = 2** ← resume the suspended socket connection.

CONNECT

hellohello ← data pending on the socket are sent to the terminal emulator.

Now, the AT interface is in ONLINE mode. All entered characters are interpreted as data to send on the connection. The remote side returns the sent message (echo).

**+++** ← suspend the connection. The escape sequence is not displayed on the terminal emulator.

OK ← the module is in COMMAND mode.

SRING: 2,3 ← SRING in Data amount mode, 2 is the connection Id, 3 is the amount of echoed data (+++).

Now, close the socket connection.

**AT#SH=2**

OK

If the socket connection is not closed, its time out expires, and the NO CARRIER message is displayed on the terminal emulator.



Open sockets <connId>=1 in COMMAND mode on an echo port.

**AT#SD=1,0,echo\_port,"address",0,0,1**

OK

Send data in hexadecimal format:

**AT#SEND=1**

> 68656C6C6FCtrl-Z

OK

SRING: 1,5,68656C6C6F ← receive data in hexadecimal format.

#### 8.7.6. SRING Unsolicited Mode: Data view UDP

Assume that the SRING mode has been configured in Data view UDP mode through the #SCFGEXT command, the sockets <connId>=1/2 have been configured, and the used PDP context has been configured and activated. Enable the error report in verbose format (+CMEE=2).

Configure <connId>=1 in Data view mode with UDP info, and set text view.

**AT#SCFGEXT = 1,3,0,0**

OK

Configure <connId>=2 in Data view mode, and set hex view.

**AT#SCFGEXT = 2,3,1,0**

OK

Open the UDP connections <connId>=1 in COMMAND mode, on echo port.

**AT#SD=1,1,echo\_port,"address",0,0,1**

OK ← the socket connection is open; the module is in COMMAND mode.

Open the UDP connections <connId>=2 in COMMAND mode, on echo port.

**AT#SD=2,1,echo\_port,"address",0,0,1**

OK ← the socket connection is open; the module is in COMMAND mode.

Send data on the first socket, text mode:

**AT#SEND=1**

> helloCtrl-Z

OK

SRING: "address",echo\_port,1,5,0,hello

Send data on the second socket, hex mode for received data:

**AT#SEND=2**

> helloCtrl-Z

OK

SRING: "address",echo\_port,2,5,0,68656C6C6F



## 9. MULTI-SOCKET & PROTOCOL SERVICE SOCKETS

### 9.1. 2G/3G Modules

Tab. 12 shows:

- Multi-socket (IP Easy environment):
  - Sockets identified by <connId>= 1, 2, 3, are bound to <cid>=1 (as example)
  - Sockets identified by <connId>= 4, 5, 6, are bound to <cid>=3 (as example)
- Protocol Service Sockets:
  - HTTP Service socket bound to <cid>=1 (by default), refer to chapter 10
  - FTP Service socket bound to <cid>=1 (mandatory), refer to chapter 12
  - SMTP Service socket bound to <cid>=1 (mandatory), refer to chapter 11
  - PING Service socket bound to <cid>=1 (mandatory), refer to chapter 13
  - SSL Service socket bound to <cid>=1 (mandatory), refer to document [5]
- Maximum active PDP contexts in the IP Easy Environment at the same time: refer to chapter 3 feature II, and chapter 4.2.1. The table shows, for example, active the PDP contexts identified by <cid>=1, and <cid>=3.

The first six sockets belong to the Multi-socket (IP Easy environment). You can use them through the #SD, #SA, etc. commands described before. Each protocol has its own service socket not belonging to the Multi-socket set.

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	No active	Active	No active	No active
1	bound	/	/	/	/
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	/	bound	/	/
5	/	/	bound	/	/
6	/	/	bound	/	/
<b>Protocol Service Sockets</b>					
HTTP	bound	/	/	/	/
FTP	bound				
SMTP	bound				
PING	bound				
SSL	bound				

Tab. 12: 2G/3G Multi-Socket & Protocol Sockets

## 9.2. 4G Modules

### 9.2.1. #PROTOCOLCFG Command

4G modules provide the #PROTOCOLCFG command to change the default binding configuration between protocols (FTP, SMTP, PING, SSL, etc.) and <cid> value identifier. Here is the command syntax.

**AT#PROTOCOLCFG =<protocol>,<cid>**

See documents [4], [11], or [13] according to the module, to get parameters descriptions.

### 9.2.2. Platform Version ID 20

#### 9.2.2.1. Modules: LE910 Series

Modules: LE910-EU1, B1-EU, -JN1, -NA1, B1-NA, B1-SA, -NA V2, B4-NA, -EU V2, AU V2

The AT#PROTOCOLCFG=? test command returns the supported protocols, and for each protocol shows the <cid> range. Refer to document [4] to get parameters descriptions.

```
AT#PROTOCOLCFG=?
#PROTOCOLCFG: "FTP", (0-15), (0), (0), (0)
#PROTOCOLCFG: "SMTP", (0-15), (0), (0), (0)
#PROTOCOLCFG: "PING", (0-15), (0), (0), (0)
#PROTOCOLCFG: "SSL", (0-15), (0), (0), (0)
#PROTOCOLCFG: "NTP", (0-15), (0), (0), (0)
OK
```

The AT#PROTOCOLCFG? read command returns the current setting. The following is the default setting.

```
AT#PROTOCOLCFG?
#PROTOCOLCFG: "FTP", 1, 0, 0, 0
#PROTOCOLCFG: "SMTP", 1, 0, 0, 0
#PROTOCOLCFG: "PING", 1, 0, 0, 0
#PROTOCOLCFG: "SSL", 1, 0, 0, 0
#PROTOCOLCFG: "NTP", 1, 0, 0, 0
OK
```

The AT+CGDCONT? read command returns the current setting for each defined PDP context. The following is the default setting.

```
AT+CGDCONT?
+CGDCONT: 1, "IPV4V6", "", "", 0, 0
OK
```

By default, the protocols service sockets are configured to use the PDP context identified by <cid>=1 but, <cid>=1 - by default - identifies a PDP context without the Access Point Name. It means that, during the attach procedure, the network assigns to <cid>=1 the Default EPS Bearer that supports the services provided by the used Network Operator. Use

the #PROTOCOLCFG command to change the default configuration, in accordance with your application.

Tab. 13 summarizes how <cid> and sockets can be used.

Sockets	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5-15
<b>Internal Sockets (not directly accessible to the user)</b>	During the attach procedure, the network assigns to the module the Default EPS Bearer that supports the services provided by the used Network Operator. By default, is used <cid>=1 bound to an internal socket. See chapter 5.1.				
<b>IP Easy Multi- Socket &lt;connId&gt;</b>	The user can bind all <cid> to sockets belonging to the Multi-socket environment, and Protocol Service Socket, at the same time.				
1-10					
<b>Protocol Service Sockets</b>					
FTP					
SMTP					
PING					
SSL					
NTP					

Tab. 13: LE910-EU1, Network Operator Service, Multi-Socket, and Protocols

The Fig. 6 summarizes an example of configuration of Multi-socket, Protocol Services, Internal Sockets, Network Interface, and <cid>.

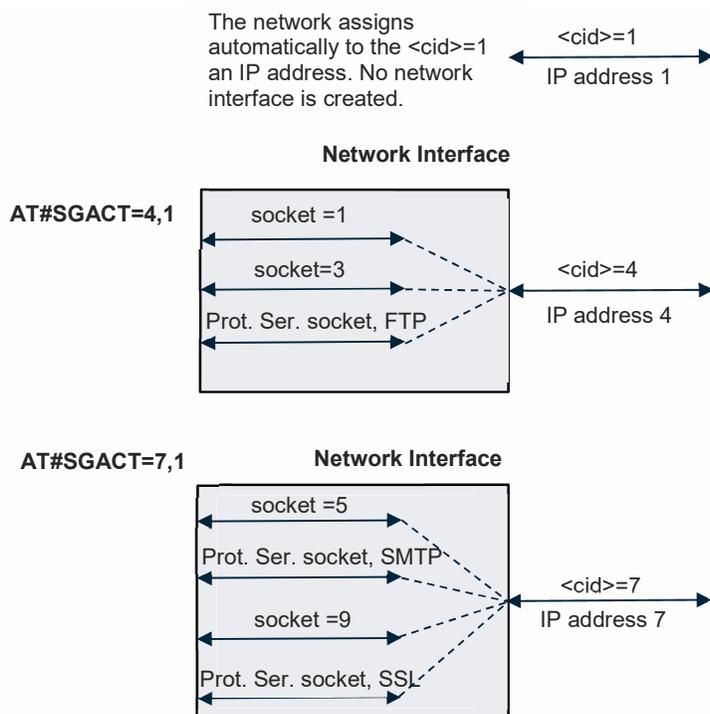


Fig. 6: LE910-EU1, Multi/Protocol/Internal Sockets, and Network Interface

#### 9.2.2.2. Modules: LE910-SV1, -SVL, SV V2

The AT#PROTOCOLCFG=? test command returns the supported protocols, and for each protocol shows the <cid> range. Refer to document [4] to get parameters descriptions.

```

AT#PROTOCOLCFG=?
#PROTOCOLCFG: "FTP", (0-15), (0), (0), (0)
#PROTOCOLCFG: "SMTP", (0-15), (0), (0), (0)
#PROTOCOLCFG: "PING", (0-15), (0), (0), (0)
#PROTOCOLCFG: "SSL", (0-15), (0), (0), (0)
#PROTOCOLCFG: "NTP", (0-15), (0), (0), (0)
OK

```

The AT#PROTOCOLCFG? read command returns the current setting. The default setting is the following.

```

AT#PROTOCOLCFG?
#PROTOCOLCFG: "FTP", 3, 0, 0, 0
#PROTOCOLCFG: "SMTP", 3, 0, 0, 0
#PROTOCOLCFG: "PING", 3, 0, 0, 0
#PROTOCOLCFG: "SSL", 3, 0, 0, 0
#PROTOCOLCFG: "NTP", 3, 0, 0, 0
OK

```

The AT+CGDCONT? read command returns the current setting for each defined PDP context. The default setting is the following.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IPV4V6","vzwims","",0,0
+CGDCONT: 2,"IPV4V6","vzwadmin","",0,0
+CGDCONT: 3,"IPV4V6","vzwinternet","",0,0
OK
```

```
Service: IMS
Service: ADMIN
Service: INTERNET
```

By default, the protocols service sockets are using the PDP context identified by <cid>=3 which is used – by default - by the INTERNET service as shown by the command above. Use the #PROTOCOLCFG command to change the default configuration, in accordance with your application.

Tab. 14 summarizes how <cid> and sockets can be used.

Sockets	<cid>=1	<cid>=2	<cid>=3	<cid>=4/5	<cid>=6	<cid>= 7-15
<b>Internal Sockets (not directly accessible to the user)</b>	By default, <cid>=1 is used for IMS Service assigned by the Network Operator, and is bound to an internal socket.	By default, <cid>=2 is used for ADMIN Service assigned by the Network Operator, and is bound to an internal socket.	By default, <cid>=3 is used for INTERNET Service assigned by the Network Operator, and is bound to an internal socket.	Unused	By default, <cid>=6 is used for INTERNET Service assigned by the Network Operator, and is bound to an internal socket.	
<b>IP Easy Multi- Socket &lt;connId&gt;</b>	The user can bind all <cid> to sockets belonging to the Multi-socket environment, and Protocol Service Sockets, at the same time.					
1-10						
<b>Protocol Service Sockets</b>						
FTP						
SMTP						
PING						
SSL NTP						

Tab. 14: LE910-SV1, IMS, ADMIN, INTERNET Services, Multi-Socket, and Protocols

### 9.2.3. Platform version ID 23

#### 9.2.3.1. Modules: LE866-SV1, ME866A1-NV

The AT#PROTOCOLCFG=? test command returns the supported protocols, and for each protocol shows the <cid> range. Refer to document [11] to get parameters descriptions.

#### AT#PROTOCOLCFG=?

```
#PROTOCOLCFG: "FTP",(1-5),(0),(0),(0)
#PROTOCOLCFG: "SMTP",(1-5),(0),(0),(0)
#PROTOCOLCFG: "PING",(1-5),(0),(0),(0)
#PROTOCOLCFG: "SSL",(1-5),(0),(0),(0)
#PROTOCOLCFG: "HTTP0",(1-5),(0),(0),(0)
#PROTOCOLCFG: "HTTP1",(1-5),(0),(0),(0)
#PROTOCOLCFG: "HTTP2",(1-5),(0),(0),(0)
OK
```

The AT#PROTOCOLCFG? read command returns the current setting. The default setting is the following.

#### AT#PROTOCOLCFG?

```
#PROTOCOLCFG: "FTP",3,0,0,0
#PROTOCOLCFG: "SMTP",3,0,0,0
#PROTOCOLCFG: "PING",3,0,0,0
#PROTOCOLCFG: "SSL",3,0,0,0
#PROTOCOLCFG: "HTTP0",3,0,0,0
#PROTOCOLCFG: "HTTP1",3,0,0,0
```

```
#PROTOCOLCFG: "HTTP2",3,0,0,0
OK
```

The AT+CGDCONT? read command returns the current setting for each defined PDP context. The default setting is the following.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IPV4V6","vzwims","",0,0,0,1,0 Service: IMS
+CGDCONT: 2,"IPV4V6","vzwapadmin","",0,0 Service: ADMIN
+CGDCONT: 3,"IPV4V6","vzwinternet","",0,0 Service: INTERNET
OK
```

By default, the protocols service sockets are using the PDP context identified by <cid>=3 which is used – by default - by the INTERNET service as shown by the command above. Use the #PROTOCOLCFG command to change the default configuration, in accordance with your application.

Tab. 15 summarizes how <cid> and sockets can be used.

Sockets	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
<b>Internal Sockets (not directly accessible to the user)</b>	By default, <cid>=1 is used for IMS Service assigned by the Network Operator, and is bound to a dedicated internal socket.	By default, <cid>=2 is used for ADMIN Service assigned by the Network Operator, and is bound to a dedicated internal socket.	By default, <cid>=3 is used for INTERNET Service assigned by the Network Operator, and is bound to a dedicated internal socket.		
<b>IP Easy Multi-Socket &lt;connid&gt;</b>	The user can bind all <cid> to sockets belonging to the Multi-socket environment, and Protocol Service Sockets, at the same time.				
1-6					
<b>Protocol Service Sockets</b>					
FTP					
SMTP					
PING					
SSL					
HTTP0/1/2					

Tab. 15: LE866-SV1, IMS, ADMIN, INTERNET Services, Multi-Socket, and Protocols

9.2.4. Platform version ID 25

9.2.4.1. Modules: LE910Cx

The AT#PROTOCOLCFG=? test command returns the supported protocols, and for each protocol shows the <cid> range. Refer to document [13] to get parameters descriptions.

**AT#PROTOCOLCFG=?**

```
#PROTOCOLCFG: "FTP",(1-5),(0),(0),(0)
#PROTOCOLCFG: "SMTP",(1-5),(0),(0),(0)
#PROTOCOLCFG: "PING",(1-5),(0),(0),(0)
#PROTOCOLCFG: "SSL",(1-5),(0),(0),(0)
```

OK

The AT#PROTOCOLCFG? read command returns the current setting.

The default setting is the following.

**AT#PROTOCOLCFG?**

```
#PROTOCOLCFG: "FTP",1,0,0,0
#PROTOCOLCFG: "SMTP",1,0,0,0
#PROTOCOLCFG: "PING",1,0,0,0
#PROTOCOLCFG: "SSL",1,0,0,0
```

OK

The AT+CGDCONT? read command returns the current setting for each defined PDP context. The following is the default setting.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IPV4V6","", "",0,0,0,0
```

OK

By default, the protocols service sockets are configured to use the PDP context identified by <cid>=1 but, <cid>=1 - by default - identifies a PDP context without the Access Point Name. It means that, during the attach procedure, the network assigns to <cid>=1 the Default EPS Bearer that supports the services provided by the used Network Operator. Use the #PROTOCOLCFG command to change the default configuration, in accordance with your application.

Tab. 16: LE910Cx, Network Operator Service, Multi-Socket, and Protocols summarizes how <cid> and sockets can be used.

Sockets	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5-15
<b>Internal Sockets (not directly accessible to the user)</b>	During the attach procedure, the network assigns to the module the Default EPS Bearer that supports the services provided by the used Network Operator. By default, is used <cid>=1 bound to an internal socket. See chapter 5.1.				
<b>IP Easy Multi- Socket &lt;connId&gt;</b>					
1-10					
<b>Protocol Service Sockets</b>					
FTP					
SMTP					
PING					
SSL					

The user can bind all <cid> to sockets belonging to the Multi-socket environment, and Protocol Service Socket, at the same time.

Tab. 16: LE910Cx, Network Operator Service, Multi-Socket, and Protocols

## 10. HTTP PROTOCOL

HTTP protocol is used in World Wide Web. To establish an HTTP connection, you can use:

- I. the #HTTPQRY/#HTTPSND commands, that create an HTTP Protocol Service socket, and establishes the HTTP connection.

Assume to use #PORTCFG command or CMUX tool, the module can provide three serial lines connected to three AT parsers, see Fig. 2, and Fig. 3. Suppose that one HTTP connection is already started and not still closed on a serial port. If you try to open a second HTTP connection on another serial port, the #HTTPQRY/#HTTPSND commands return an error message. When the old HTTP connection is closed, you can open the new one.

- II. the #SD command that uses a Multi-socket, see chapter 10.4.4. In this case, the user application must manage the HTTP protocol.

### 10.1. #HTTPCFG Command

Use #HTTPCFG command to set one of the three profiles, provided by modules, to configure an HTTP connection. The profiles are identified by the <prof\_id> parameter and are automatically saved in NVM.

#### 10.1.1. 2G Modules

```
AT#HTTPCFG=<prof_id>[,<server_address>[,<server_port>[,<auth_type>
[,<username>[,<password>[,<ssl_enable>[,<timeout>[,<cid> ]]]]]]]
```

The AT#HTTPCFG=? test command returns the supported range of parameters value.  
Example: <prof\_id> range: 0-2, <cid> range: 1-5.

**AT#HTTPCFG=?**

```
#HTTPCFG: (0-2),50,(1-65535),(0,1),50,50,(0,1),(1-65535),(1-5)
OK
```

Referring to the example shown in Tab. 17:

- HTTP Service socket bound to <cid>=1 (by default, it can be changed)
- Sockets <connId>= 1, 2, 3 not opened, and bound to active <cid>=1
- Sockets <connId>= 4, 5, 6 not opened, and bound to active <cid>=3

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	No active	Active	No active	No active
1	bound	/	/	/	/
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	/	bound	/	/
5	/	/	bound	/	/
6	/	/	bound	/	/
<b>Protocol Service Sockets</b>					
HTTP Service Socket	bound (default)	/	/	/	/

Tab. 17: 2G Multi-Socket & HTTP Service Socket

The AT#HTTPCFG? read command returns the current setting for each profile with the following format:

```
#HTTPCFG:<prof_id>,<server_address>,<server_port>,<auth_type>,<username>,<password>,<ssl_enabled>,<timeout>,<cid><CR><LF>
```

These are the default profiles

**AT#HTTPCFG?**

```
#HTTPCFG: 0,"",80,0,"", "*****",0,120,1
#HTTPCFG: 1,"",80,0,"", "*****",0,120,1
#HTTPCFG: 2,"m2mlocate.telit.com",9978,0,"", "*****",0,120,1
OK
```

Use AT#HTTPCFG=<prof\_id> command form to reset the values of the <prof\_id> profile and set the default values.

See documents [1] to get parameters descriptions.

10.1.2. 3G Modules

```
AT#HTTPCFG=<prof_id>[,<server_address>[,<server_port>[,<auth_type>[,<username>[,<password>[,<ssl_enable>[,<timeout>[,<cid>[,<pkt_size>[,<UNUSED_1>[,<UNUSED_2>]]]]]]]]]]
```

The AT#HTTPCFG=? test command returns the supported range of parameters values.  
Example: <prof\_id> range: 0-2, <cid> range: 1-5.

**AT#HTTPCFG=?**

```
#HTTPCFG: (0-2),50,(1-65535),(0,1),50,50,(0,1),(1-65535),(1-5),(0-1500),(0),(0)
OK
```

Referring to the example shown in Tab. 18:

- HTTP Service socket bound to <cid>=1 (by default, it can be changed)
- Sockets <connId>= 1, 2, 3 not opened, and bound to active <cid>=1
- Sockets <connId>= 4, 5, 6 not opened, and bound to active <cid>=3

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	No active	Active	No active	No active
1	bound	/	/	/	/
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	/	bound	/	/
5	/	/	bound	/	/
6	/	/	bound	/	/
Protocol Service Sockets					
HTTP Service Socket	bound (default)	/	/	/	/

Tab. 18: 3G Multi-Socket & HTTP Service Socket

The AT#HTTPCFG? read command returns the current setting for each profile with the following format:

```
#HTTPCFG:<prof_id>,<server_address>,<server_port>,<auth_type>,<
```



## 10.1.3.1. Platform Version ID 20

## 10.1.3.1.1. Modules: LE910 Series

Modules: LE910-EU1, B1-EU, -JN1, -NA1, B1-NA, B1-SA, -NA V2, B4-NA, -EU V2, AU V2

The AT#HTTPCFG=? test command returns the supported range of parameters values. Pay attention to: <prof\_id> range: 0-2, <cid> range: 1-15.

**AT#HTTPCFG=?**

#HTTPCFG: (0-2),50,(1-65535),(0,1),50,50,(0,1),(1-65535),(1-15),(0-1500),(0),(0)  
OK

## 10.1.3.1.2. Modules: LE910-SV1, -SVL, SV V2

The AT#HTTPCFG=? test command returns the supported range of parameters values. Pay attention to: <prof\_id> range: 0-2, <cid> range: 1-15.

**AT#HTTPCFG=?**

#HTTPCFG: (0-2),50,(1-65535),(0,1),50,50,(0,1),(1-65535),(1-15),(0-1500),(0),(0)  
OK

## 10.1.3.2. Platform Version ID 23

## 10.1.3.2.1. Modules: LE866-SV1, ME866A1-NV

The AT#HTTPCFG=? test command returns the supported range of parameters values. Pay attention to: <prof\_id> range: 0-2, <cid> range: 1-5.

**AT#HTTPCFG=?**

#HTTPCFG: (0-2),50,(1-65535),(0,1),50,50,(0,1),(1-65535),(1-5),(0-1500),(0),(0)  
OK

## 10.1.3.3. Platform Version ID 25

## 10.1.3.3.1. Modules: LE910Cx

The AT#HTTPCFG=? test command returns the supported range of parameters values. Pay attention to: <prof\_id> range: 0-2, <cid> range: 1-5.

**AT#HTTPCFG=?**

#HTTPCFG: (0-2),63,(1-65535),(0,1),50,50,(0),(1-65535),(1-5)

OK

## 10.2. #HTTPQRY: GET, HEAD, DELETE

Use the #HTTPQRY command to send an HTTP command request (GET, HEAD, DELETE, see RFC 2616 document) to the HTTP server. If the request succeeds, the #HTTPQRY command returns the OK message, otherwise an error code is reported.

#HTTPQRY sends the HTTP request header containing the "Connection: close" option, see RFC 2616 document, and it cannot be removed. The "close" connection option signals that the connection will be closed after the current request/response is complete.

**AT#HTTPQRY=<prof\_id>,<command>,<resources>[,<extra\_header\_lines>]**

When the HTTP server answer is received, the following URC is displayed on the DTE:

#HTTPRING: <prof\_id>,<http\_status\_code>,<content\_type>,<data\_size>

To get parameters descriptions, see documents [1], [2], [4], or [11], according to the used module.

### NOTES:

- I. If no data are received from the HTTP server, or the HTTP server does not answer within the time interval specified in <timeout> parameter of #HTTPCFG command, the <http\_status\_code> parameter of the #HTTPRING URC is 0.
- II. The time required to receive the #HTTPRING URC could be greater than the one specified in <timeout> parameter of #HTTPCFG command, because it also includes the time needed to send the HTTP request to the server.

## 10.3. #HTTPSND: POST, PUT

Use the #HTTPSND command to send data to HTTP server using POST or PUT HTTP command requests, see RFC 2616 document. After command line is terminated with <CR>, the command responds with the following three characters: >>>. Now, you can digit the characters to send to the server. If the operation succeeds, the #HTTPSND command returns the OK message, otherwise an error code is reported.

#HTTPSND sends the HTTP request header containing the "Connection: close" option, see RFC 2616 document, and it cannot be removed. The "close" connection option signals that the connection will be closed after the current request/response is complete.

**AT#HTTPSND=<prof\_id>,<command>,<resource>,<data\_len>[,<post\_param> [,<extra\_header\_line>]]**

When the HTTP server answer is received, the following URC is displayed on the DTE:

#HTTPRING: <prof\_id>,<http\_status\_code>,<content\_type>,<data\_size>

To get parameters descriptions, see documents [1], [2], [4], or [11], according to the used module.

### NOTES

- I. If no data are received from the HTTP server, or the HTTP server does not answer within the time interval specified in <timeout> parameter of #HTTPCFG command, then the <http\_status\_code> parameter of the #HTTPRING URC is 0.
- II. The time required to receive the #HTTPRING URC could be greater than the one specified in <timeout> parameter of #HTTPCFG command, because it also includes the time needed to send the HTTP request to the server.

## 10.4. Examples

### 10.4.1. GET Command (2G)

This example shows the use of the GET command.

Define PDP context <cid=1>.

```
AT+CGDCONT=1,"IP", "Access_Point_Name"
```

```
OK
```

Activate PDP Context <cid>=1. The #SGACT command returns the IP address assigned by the network to the module.

```
AT#SGACT=1,1
```

```
#SGACT: 10.7.125.7
```

```
OK
```

Configure the HTTP server parameters.

```
AT#HTTPCFG=0,"server_address",80,0,,0,120,1
```

```
OK
```

Use #HTTPQRY to open the HTTP connection and send the GET command to the HTTP server. The connection will be close after completion of the response ("Connect: close" option).

```
AT#HTTPQRY=0,0,"/"
```

```
OK
```

← GET command succeeds.

When the HTTP server answer is received, the HTTP connection is closed, and a URC is displayed on the terminal emulator.

```
#HTTPRING: 0,200,"text/html", ...
```

Type in the #HTTPRCV command to read data sent by HTTP server and stored in the socket buffer.

```
AT#HTTPRCV=0
```

```
<!doctype html>
```

```
<html>
```

```
.....
```

```
</html>
```

```
OK
```

#### 10.4.2. POST Command (2G)

This example shows the use of the POST command.

Define PDP context.

```
AT+CGDCONT=1,"IP","Access_Point_Name"
```

OK

Context activation.

```
AT#SGACT=1,1
```

```
#SGACT: 10.7.125.7
```

OK

Configure HTTP server parameters.

```
AT#HTTPCFG=0,"server_address",80,0,,0,120,1
```

OK

Use #HTTPSND to open the HTTP connection and send the POST command to the HTTP server. The connection will be close after completion of the response ("Connect: close" option). In this example, you digit 27 characters.

```
AT#HTTPSND=0,0,"/",27
```

```
>>> digit 27 characters
```

OK

When the HTTP server answer is received, the following URC is put on the serial port.

```
#HTTPRING: 0,200,"text/html",1270
```

```
AT#HTTTRCV=0
```

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
<title>Example POST response</title> ...
```

OK

#### 10.4.3. PUT Command (2G)

This example shows the use of the PUT command.

Define PDP context.

```
AT+CGDCONT=1,"IP","Access_Point_Name"
```

OK

Context activation.

```
AT#SGACT=1,1
```

```
#SGACT: 10.7.125.7
```

OK

Configure HTTP server parameters.

```
AT#HTTPCFG=0,"server_address",80,0,,0,120,1
```

OK

Use #HTTPSND to open the HTTP connection and send the PUT command to the HTTP server. The connection will be close after completion of the response ("Connect: close" option. In this example, you digit 27 characters.

**AT#HTTPSND=0,1,"/",27**

>>> digit 27 characters

OK

When the HTTP server answer is received, the following URC is put on the serial port.

```
#HTTPRING: 0,200,"text/html",1270
```

**AT#HTTPRCV=0**

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
<title>Example PUT response</title> ...
```

OK

#### 10.4.4. HTTP Client with #SD

This example establishes an HTTP connection toward an HTTP server and retrieves an HTML page using a socket dial connection (#SD). Assume that a terminal emulator is connected to the module. See, for comparison, the example described in chapter 14.3.3.

Check the sockets/PDP contexts binding configuration (default).

**AT#SCFG?**

```
#SCFG: 1,1,300,90,600,50
```

```
#SCFG: 2,1,300,90,600,50
```

```
#SCFG: 3,1,300,90,600,50
```

```
#SCFG: 4,2,300,90,600,50
```

```
#SCFG: 5,2,300,90,600,50
```

```
#SCFG: 6,2,300,90,600,50
```

OK

Define PDP Context <cid>=1.

**AT+CGDCONT = 1,"IP","Access\_Point\_Name","0.0.0.0",0,0**

OK

Check which are the currently defined PDP contexts.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IP","Access_Point_Name","0.0.0.0",0,0
```

OK

Activate the PDP context <cid>=1, assume that no UserID and Password are needed. The command returns the IP address assigned by the network to the module.

**AT#SGACT=1,1**

```
#SGACT: 2.43.168.172
```

OK

Open the connection toward the HTTP server on port=80. Transport protocol is TCP.

**AT#SD=1,0,80,"server\_address",0,0**

```
CONNECT ← the connection is open. The module is in ONLINE mode.
```

Now, following the HTTP protocol, ask the homepage by sending the next lines to the module. The entered data are sent directly to the HTTP server. The strings sent to the HTTP server must be ended by line feed character. Enable the local echo on terminal emulator to see the issued commands.

```
GET / HTTP/1.1<cr><lf>
Host: www.telit.com<cr><lf>
Connection: keep-alive<cr><lf>
<cr><lf>
```

The HTTP server replies with the HTML code of the homepage and some debugging responses.

```
HTTP/1.1 200 OK
.....
<!DOCTYPE html>
<html>
... here is all the HTML code of the page ...
</html>
```

NO CARRIER ← HTTP connection is closed.

Check the disconnection cause.

**AT#SLASTCLOSURE=1**

#SLASTCLOSURE: 1,1

OK

← HTTP connection closed by remote host.

## 11. SMTP PROTOCOL

SMTP is a standard protocol for electronic mail (e-mail). To establish a SMTP connection, you can use:

- I. the #EMAILD/#SMTPCL commands, that create a SMTP Protocol Service socket, and establishes the SMTP connection.

Assume to use #PORTCFG command or CMUX tool, the module can provide three serial lines connected to three AT parsers, see Fig. 2, and Fig. 3. Suppose that one SMTP connection is already started and not still closed on a serial port. If you try to open a second SMTP connection on another serial port, the #EMAILD/ #SMTPCL commands return an error message. When the old SMTP connection is closed, you can open the new one.

- II. the #SD command that uses a Multi-socket, as shown by the examples in chapters 11.6.3, and 11.6.4. In this case, the user application must manage the SMTP protocol.

### 11.1. 2G Modules

The SMTP connection can use only the PDP Context identified by <cid>=1, or the GSM context identified by <cid>=0. The SMTP command uses one of the two contexts according to the command previously entered: AT#SGACT=1,1 or AT#SGACT =0,1 respectively. For GSM context (<cid>=0) see chapter 14.3.5.

Use #SMTPCFG command to configure the SMTP connection. The command syntax is:

**AT#SMTPCFG=<ssl\_enable>[,<port>[,<mode>[,<charset>[,<UNUSED\_1> [,<UNUSED\_2>]]]]]**

See document [1] to get parameters descriptions.

Referring to the example shown in Tab. 19:

- SMTP Service socket bound to <cid>=1 (mandatory)
- Sockets <connId>= 1, 2, 3 not opened, and bound to the active <cid>=1
- Sockets <connId>= 4, 5, 6 not opened, and bound to the active <cid>=3

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
1	Active	No active	Active	No active	No active
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	/	bound	/	/
5	/	/	bound	/	/
6	/	/	bound	/	/
<b>Protocol Service Sockets</b>					
SMTP Service Socket	bound				

Tab. 19: 2G Multi-Socket & SMTP Service Socket

## 11.2. 3G Modules

The SMTP connection can use only the PDP Context identified by <cid>=1, or the GSM context identified by <cid>=0, both settings are mandatory. The SMTP command selects one of the two contexts according to the command previously entered: AT#SGACT=1,1 or AT#SGACT =0,1 respectively. For GSM context (<cid>=0) see chapter 14.3.5.

**AT#SMTPCFG=<ssl\_enable>[,<port>,<mode>,<UNUSED\_1>  
[,<pkt\_size>,<UNUSED\_2]]]]]**

See document [2] to get parameters descriptions.

Referring to the example shown in Tab. 20:

- SMTP Service socket bound to <cid>=1 (mandatory)
- Sockets <connId>= 1, 2, 3 not opened, and bound to the active <cid>=1
- Sockets <connId>= 4, 5, 6 not opened, and bound to the active <cid>=3

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	No active	Active	No active	No active
1	bound	/	/	/	/
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	/	bound	/	/
5	/	/	bound	/	/
6	/	/	bound	/	/
<b>Protocol Service Sockets</b>					
SMTP Service Socket	bound				

Tab. 20: 3G Multi-Socket & SMTP Service Socket

## 11.3. 4G Modules

**AT#SMTPCFG=<ssl\_enable>[,<port>,<mode>,<UNUSED\_1>  
[,<pkt\_size>,<UNUSED\_2]]]]]**

To get parameters descriptions, see documents [4], or [11], according to the module, and refer to chapter 9.2 (#PROTOCOLCFG) to change the <cid> identifier bound to SMTP protocol.

NOTE: Platform Version ID 25:

1. It has different command format than others such as

**AT#SMTPCFG=<ssl\_enabled>[,<port>,<mode>,<UNUSED\_1>,<UNUSED\_2>,<UNUSED\_3>]]]]]**

## 11.4. E-mail Sending, with no Attachment

Use the #EMAILD command to send an e-mail to the SMTP server. When you enter the command, and the following <CR>, the command returns the ">" prompt and waits for the message body text. To close the message, enter Ctrl-Z (char 0x1A), to exit without writing the message enter ESC (char 0x1B).

**AT#EMAILD=<da>,<subj>**

If e-mail message is successfully sent, the command returns the OK message, otherwise returns an error code.

To get parameters descriptions, see documents [1], [2], [4], [11], or [13], according to the module.

## 11.5. E-mail Sending with Attachment

Use the #SMTPCL command to send an e-mail with attachment.

**AT#SMTPCL=<da>,<subj>,<att>[,<filename>,<encod>]**

To get parameters descriptions, see documents [1], [2], [4], or [11], according to the module.

NOTE: Platform Version ID 25:

- I. AT#SMTPCL is not supported.

## 11.6. Examples

### 11.6.1. With no Attachment

This example shows how to send an e-mail to a recipient. A terminal emulator is connected to the module.

Enable error in verbose format.

**AT+CMEE=2**

OK

Define PDP context <cid>=1.

**AT+CGDCONT=1,"IP", "Access\_Point\_Name"**

OK

To activate successfully a PDP context, it must be bound to a socket by means of the #SCFG command.

Assume that no user name and password are required to activate it. The command returns the IP address assigned by the network to the module.

**AT#SGACT=1,1**

#SGACT: 10.7.125.7

OK

Set the SMTP connection parameters.

**AT#SMTPCFG=0,587**

OK

**AT#ESMTP="SMTP\_server\_address"** ← config. SMTP server address of the sender.

OK

**AT#EADDR="sender\_address"** ← configure e-mail address of the sender.

OK

**AT#EUSER="sender\_user\_name"** ← configure user name of the sender.

OK

**AT#EPASSW="sender\_user\_password"** ← configure user password of the sender.

OK

Send the e-mail with no attachment to the recipient having recipient\_address.

**AT#EMAILD="recipient\_address","mail with no attachment"**

> Hello<Ctrl-Z>

OK ← text of the e-mail has been sent successfully.

### 11.6.2. With Attachment

This example shows how to send an e-mail to a recipient with an attachment. A terminal emulator is connected to the module.

Enable error in verbose format.

**AT+CMEE=2**

OK

Define PDP context.

**AT+CGDCONT=1,"IP", "Access\_Point\_Name"**

OK

To activate successfully a PDP context, it must be bound to a socket by means of the #SCFG command.

The command returns the IP address assigned by the network to the module.

**AT#SGACT=1,1**

#SGACT: 10.7.125.7

OK

Set the SMTP connection parameters.

**AT#SMTPCFG=0,587**

OK

**AT#ESMTP="SMTP\_server\_address"** ← config. SMTP server address of the sender.

OK

**AT#EADDR="sender\_address"** ← configure e-mail address of the sender.

OK

**AT#EUSER= "sender\_user\_name"** ← configure user name of the sender.

OK

**AT#EPASSW="sender\_user\_password"** ← configure user password of the sender.

OK

Send the e-mail, with a picture in the attachment, to the recipient having recipient\_address.

**AT#SMTPCL="recipient\_address","mail with picture",2,"picture.jpg"**

> Hello.<CR><LF>

This is the body.<CR><LF>

The picture is in the attachment.<CR><LF><Ctrl-Z>

CONNECT ← the module enters ONLINE mode to send the attachment.

... binary data of the attachment ...

**+++** ← type in the escape sequence to close the connection.

NO CARRIER ← the module returns this message and enters COMMAND mode.

### 11.6.3. Sending with #SD

This example describes how create a SMTP connection, and sends an e-mail using the socket dial command (#SD). The socket is one of the sockets provided by the Multi-socket feature. Assume that a terminal emulator is connected to the module.

Enable reports in verbose format.

**AT+CMEE=2**

OK

Define PDP context <cid>=1.

**AT+CGDCONT=1,"IP", "Access\_Point\_Name"**

OK

To activate successfully a PDP context, it must be bound to a socket by means of the #SCFG command.

Assume that no user name and password are required to activate it. The command returns the IP address assigned by the network to the module.

**AT#SGACT=1,1**

#SGACT: 193.199.234.255

OK

Open the connection toward the SMTP server using "SMTP\_server\_address" and port 25.

**AT#SD=1, 0,25," SMTP\_server\_address ",0,0**

CONNECT ← the module is in ONLINE mode

Now, the user application sends the SMTP protocol commands: HELO, MAIL FROM, RCPT TO, DATA, QUIT.

.....  
 .....  
 .....

**+++** ← type in the escape sequence to suspend the connection.

OK ← the module is in COMMAND mode.

**AT#SH=1**

OK ← the connection is closed.

#### 11.6.4. Receiving with #SD

This example describes how create a connection toward a POP server, and receive an e-mail using a socket dial command (#SD). The socket is one out of the N provided by the Multi-socket feature. A terminal emulator is connected to the module.

Enable reports in verbose format.

**AT+CMEE=2**

OK

Define PDP context <cid>=1.

**AT+CGDCONT=1,"IP", "Access\_Point\_Name"**

OK

Configure the socket <connId>=1 and bind it to PDP context <cid>=1.

**AT#SCFG=1,1,300,90,600,50**

OK

To activate successfully a PDP context, it must be bound to a socket by means of the #SCFG command.

Assume that no user name and password are required to activate it. The command returns the IP address assigned by the network to the module.

**AT#SGACT=1,1**

#SGACT: 193.199.234.255

OK

Open the connection toward the POP server on port=110. Transport protocol is TCP.

**AT#SD=1,0,110,"POP\_server\_address",0,0**

CONNECT ← the module is in ONLINE mode.

Now, the user application use the POP3 protocol.

.....  
 .....  
 .....

**+++** ← type in the escape sequence to suspend the connection.

OK ← the module is in COMMAND mode.

**AT#SH=1**

OK ← the connection is closed.

## 12. FTP PROTOCOL

FTP is a standard network protocol used to transfer files between a client and server on a network. To establish an FTP connection, you can use:

- I. the #FTPOPEN command, that creates a FTP Protocol Service socket, and establishes the FTP control connection. Then, you can use #FTPGET or #FTPGET command to open the next data connection.

Assume to use #PORTCFG command or CMUX tool, the module can provide three serial lines connected to three AT parsers, see Fig. 2, and Fig. 3. Suppose that one FTP control connection is already started on a serial port. If you try to open a second FTP control connection on another serial port, the #FTPOPEN command returns an error message. Close the old FTP control connection, to open the new one. See example in chapter 12.3.1.

- II. the #SD command that uses a Multi-socket. In this case, the user application must manage the FTP protocol.

### 12.1. 2G/3G Modules

The FTP connection can use only the PDP Context identified by <cid>=1, or the GSM context identified by <cid>=0. The FTP command uses one of the two contexts according to the command previously entered: AT#SGACT=1,1 or AT#SGACT =0,1 respectively. For GSM context (<cid>=0) see chapter 14.3.4.

Use #FTPOPEN command to open a FTP connection (control connection). The command syntax is:

**AT#FTPOPEN=<server:port>,<username>,<password>[,<mode>]**

Referring to the example shown in Tab. 21:

- FTP Service socket bound to <cid>=1 (mandatory)
- Sockets <connId>= 1, 2, 3, not opened, and bound to the active <cid>=1
- Sockets <connId>= 4, 5, 6, not opened, and bound to the active <cid>=3

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	No active	Active	No active	No active
1	bound	/	/	/	/
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	/	bound	/	/
5	/	/	bound	/	/
6	/	/	bound	/	/
<b>Protocol Service Sockets</b>					
FTP Service Socket	bound				

Tab. 21: 2G/3G Multi-Socket & FTP Service Socket

Use #FTPCLOSE command to close a FTP control connection. The command syntax is:

**AT#FTPCLOSE**

Refer to example 12.8.1.

To get parameters descriptions see documents [1], [2], or [13], according to the used module.

## 12.2. 4G Modules

Use #FTPOPEN command to open a FTP connection (control connection). The command syntax is:

**AT#FTPOPEN=<server:port>,<username>,<password>[,<mode>]**

Refer to chapter 9.2 (#PROTOCOLCFG command) to change the <cid> identifier bound to FTP protocol.

Use #FTPCLOSE command to close a FTP connection (control connection). The command syntax is:

**AT#FTPCLOSE**

Refer to example 12.8.1.

To get parameters descriptions see documents [4], [11], or [13], according to the used module.

## 12.3. Examples

### 12.3.1. Open Six Sockets and the FTP Connection

This example shows the opening of six sockets (Multi-socket) in COMMAND mode, plus the FTP connection. It is assumed to use only one serial port, a terminal emulator is connected to the module.

Enable reports in verbose format.

**AT+CMEE=2**  
OK

Check the socket configuration (default).

**AT#SCFG?**  
#SCFG: 1,1,300,90,600,50  
#SCFG: 2,1,300,90,600,50  
#SCFG: 3,1,300,90,600,50  
#SCFG: 4,2,300,90,600,50  
#SCFG: 5,2,300,90,600,50  
#SCFG: 6,2,300,90,600,50  
OK

Check PDP contexts.

**AT+CGDCONT?**  
OK ← no PDP context are defined.

Define PDP context <cid>=1.

**AT+CGDCONT=1,"IP", "Access\_Point\_Name"**  
OK

Define PDP context <cid>=2.

**AT+CGDCONT=2,"IP", "Access\_Point\_Name"**  
OK

Check PDP contexts.

**AT+CGDCONT?**

+CGDCONT: 1,"IP","Access\_Point\_Name","0.0.0.0",0,0

+CGDCONT: 2,"IP","Access\_Point\_Name","0.0.0.0",0,0

OK

To activate successfully a PDP context, it must be previously bound to a socket by means of the #SCFG command.

Activate PDP context <cid>=1. The command returns the IP address assigned by the network to the module.

**AT#SGACT=1,1**

#SGACT: 37.176.124.199

OK

Activate PDP context <cid>=2

**AT#SGACT=2,1**

#SGACT: 31.157.11.11

OK

Open the six sockets connection in COMMAND mode on echo port.

**AT#SD=1,0,remote\_host\_port," remote\_IP\_address ",0,0,1**

OK

**AT#SD=2,0,remote\_host\_port," remote\_IP\_address ",0,0,1**

OK

**AT#SD=3,0,remote\_host\_port," remote\_IP\_address ",0,0,1**

OK

**AT#SD=4,0,remote\_host\_port," remote\_IP\_address ",0,0,1**

OK

**AT#SD=5,0,remote\_host\_port," remote\_IP\_address ",0,0,1**

OK

**AT#SD=6,0,remote\_host\_port," remote\_IP\_address ",0,0,1**

OK

Before using #FTPOPEN command, the PDP context identified by <cid>=1, mandatorily assigned to the FTP Application, must be defined. Then <cid>=1 must be bound to a Multi-socket and activated through #SGACT command to create the related Network Interface. The #FTPOPEN command uses a Protocol Service Socket, but #SGACT command requires the PDP context bound to a Multi-socket.

Open a FTP connection (control port) toward the FTP server. The FTP server port is not specified; therefore, the command uses the default value 21. The FTP connection is in active mode (default).

**AT#FTPOPEN="server","username","password"**

OK

← FTP connection open, and in active mode.

Check the state of the six sockets (Multi-socket environment)

**AT#SS**

#SS: 1,2,37.176.124.199,64916,remote\_IP\_address, remote\_host\_port

```
#SS: 2,2,37.176.124.199,64917,remote_IP_address, remote_host_port
#SS: 3,2,37.176.124.199,64918,remote_IP_address, remote_host_port
#SS: 4,2,31.157.11.11,64919,remote_IP_address, remote_host_port
#SS: 5,2,31.157.11.11,64920,remote_IP_address, remote_host_port
#SS: 6,2,31.157.11.11,64921,remote_IP_address, remote_host_port
OK
```

Try to open again the FTP connection just to verify that it is already open.

```
AT#FTPOPEN="server","username","password"
```

```
+CME ERROR: Already connected
```

The table below shows the current connections configuration.

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	Active	No active	No active	No active
1	open	/	/	/	/
2	open	/	/	/	/
3	open	/	/	/	/
4	/	open	/	/	/
5	/	open	/	/	/
6	/	open	/	/	/
<b>Protocol Service Sockets</b>					
FTP	open				

Tab. 22: 2G/3G Multi-Socket & FTP Service Socket Opened

#### 12.4. FTP Time Out Setting, #FTPTO Command

Use the #FTPTO command to set the timeout used when opening either the FTP control connection or the FTP data connection, it is not saved in NVM. If <tout> parameter is omitted, the command returns the current FTP time-out. The command syntax is:

```
AT#FTPTO [=<tout>]
```

To get parameters descriptions see documents [1], [2], [4], [11], or [13], according to the used module.

Refer to example 12.8.1.

#### 12.5. FTP Transfer Type Setting, #FTPTYPE Command

Use the #FTPTYPE command to configure the file transfer type (binary or ASCII). The command must be used after the FTP control connection has been opened. If the <type> parameter is omitted, the command returns the current file transfer type. The command syntax is:

```
AT#FTPTYPE [=<type>]
```

To get parameters descriptions see documents [1], [2], [4], [11], or [13], according to the used module.

Refer to example 12.8.1.

## 12.6. FTP Uploading, #FTPPUT, #FTPAPPEXT Commands

Assume that the FTP connection is open (control connection). The #FTPPUT command opens the data connection. The command syntax is:

**AT#FTPPUT =[[<filename>],[<connMode>]]**

Refer to examples 12.8.1, and 12.8.2.

Use #FTPAPPEXT command to upload data into the file stored on the FTP server. The command syntax is:

**AT#FTPAPPEXT=<bytestosend>[,<eof>]**

Refer to example 12.8.2.

To get parameters descriptions see documents [1], [2], [4], [11], or [13], according to the used module.

## 12.7. #FTPFSIZE, #FTPAPP Commands

Use the #FTPFSIZE command to know how many bytes have been received by the FTP server if the file uploading has been interrupted during the #FTPPUT command. Enter the command during the FTP control connection. Before typing in #FTPFSIZE command, it is mandatory to issue #FTPTYPE=0 command to set binary transfer type. The command syntax is:

**AT#FTPFSIZE=<filename>**

Use #FTPAPP command to append the missing part of the file. Use the #FTPFSIZE response to know the restart position of the local file.

**AT#FTPAPP =[[<filename>],[connMode>]**

To get parameters descriptions see documents [1], [2], [4], [11], or [13], according to the used module.

## 12.8. Examples

### 12.8.1. Data Uploading in ONLINE Mode

This example shows how to upload a file to a FTP server when the module is in ONLINE mode. A terminal emulator is connected to the module.

Enable reports in verbose format.

**AT+CMEE=2**

OK

Check the socket configuration (default).

**AT#SCFG?**

#SCFG: 1,1,300,90,600,50

#SCFG: 2,1,300,90,600,50

```
#SCFG: 3,1,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,2,300,90,600,50
OK
```

Check the PDP contexts.

```
AT+CGDCONT?
```

```
OK ← no PDP context
```

Define PDP context <cid>=1.

```
AT+CGDCONT=1,"IP", "Access_Point_Name"
```

```
OK
```

Check the PDP contexts.

```
AT+CGDCONT?
```

```
+CGDCONT: 1,"IP","Access_Point_Name","0.0.0.0",0,0
```

```
OK
```

To activate successfully a PDP context, it must be previously bound to a socket by means of the #SCFG command.

Activate PDP context <cid>=1. The command returns the IP address assigned by the network to the module.

```
AT#SGACT=1,1
```

```
#SGACT: 193.199.234.255
```

```
OK
```

Set the FTP time-out.

```
AT#FTPTO=1000
```

```
OK
```

Before using #FTPOPEN command, the PDP context identified by <cid>=1, mandatorily assigned to the FTP Application, must be defined. Then <cid>=1 must be bound to a Multi-socket and activated through #SGACT command to create the related Network Interface. The #FTPOPEN command uses a Protocol Service Socket, but #SGACT command requires the PDP context bound to a Multi-socket.

Open a FTP connection toward the FTP server. The FTP server port is not specified, therefore the command uses the default value 21. The FTP connection is in active mode (default).

```
AT#FTPOPEN="server","username","password"
```

```
OK ← FTP connection open and in active mode.
```

Set the file type (ASCII).

```
AT#FTPTYPE=1
```

```
OK
```

The following #FTPPUT command configuration opens the data connection, and the module enters ONLINE mode. filename.txt is the file name where the data will be stored on the FTP server. If the file you are sending is a text file, the extension must be .txt.

```
AT#FTPPUT="filename.txt",0
```

```
CONNECT ← data connection is open.
```

... type in the data to write in the filename.txt file stored on the FTP server ...

**+++** ← close the FTP data connection.

NOCARRIER

Close FTP control connection.

**AT#FTPCLOSE**

OK ← FTP control connection is closed.

Deactivate the PDP context.

**AT#SGACT=1,0**

OK

## 12.8.2. Data Uploading in COMMAND Mode

This example shows how to upload data toward a FTP server when the module is in COMMAND mode. A terminal emulator is connected to the module.

Before using #FTPOPEN command, the PDP context identified by <cid>=1, mandatorily assigned to the FTP Application, must be defined. Then <cid>=1 must be bound to a Multi-socket and activated through #SGACT command to create the related Network Interface. The #FTPOPEN command uses a Protocol Service Socket, but #SGACT command requires the PDP context bound to a Multi-socket.

Open the FTP connection toward the FTP server. The FTP server port is not specified; therefore, the command uses the default value 21. The FTP connection is in passive mode.

**AT#FTPOPEN="server","username","password",1**

OK ← FTP control connection open in passive mode.

The #FTPPUT command opens the data connection, and the module remains in COMMAND mode. filename.txt is the file name where the data will be stored on the FTP server.

**AT#FTPPUT="filename.txt",1**

OK ← data connection is open, the module is in COMMAND mode

Enter the #FTPAPPEXT command to upload data. After entering <CR>, the command returns the ">" prompt. Now, enter the data to be sent to the FTP server. As soon <bytestosend> bytes are written, data are sent to the FTP server, and the #FTPAPPEXT message is returned.

**AT#FTPAPPEXT=bytestosend**

>... type in data...

#FTPAPPEXT: <SentBytes>

OK

Use again the #FTPAPPEXT=bytestosend command to send a new data chunk.

To send the last data chunk and close the FTP connection, use the following:

**AT#FTPAPPEXT=bytestosend,1**

If a new file must be sent, repeat the sequence. If the file must be appended, use the AT#FTPAPP command (with <connMode> = 1).

**NOTE:** if while sending the chunks, the data connection is closed from remote side, user will be notified of it because #FTPAPPEXT command will return an error message, and the

related cause (if +CMEE=2 is set). Data connection must be reopened with #FTPPUT command.

## 12.9. FTP Downloading

### 12.9.1. Downloading in ONLINE mode

Assume that the FTP connection is open (control connection). Use the #FTPGET command to open a data connection and download a file from the FTP server. If the data connection succeeds, the command returns the CONNECT indication followed by the content of the file, at the end NO CARRIER indication is displayed. If no FTP control connection has been previously opened, the command returns an error message. Refer to example 12.11.1. The command syntax is:

**AT#FTPGET =<filename>**

To get parameters descriptions see documents [1], [2], [4], [11], or [13], according to the used module.

### 12.9.2. Downloading in COMMAND mode

Assume that the FTP connection is open (control connection). Use the #FTPGETPKT command to open a data connection, and download a file from the FTP server. If the data connection succeeds, the command stores the received data on the socket and returns the OK message. The module remains in COMMAND mode. If no FTP control connection has been previously opened, the command returns an error message. Refer to example 12.11.2. The command syntax is:

**AT#FTPGETPKT=<filename>[,<viewMode>]**

To get parameters descriptions see documents [1], [2], [4], [11], or [13], according to the used module.

After issuing #FTPGETPKT command, you can type in AT commands as usual in COMMAND mode, except for FTP commands that open data connection like #FTPLIST, because the data connection has been already opened by #FTPGETPKT itself.

When the data connection is open, use the #FTP\_RECV? read command to check how many buffered bytes are currently available.

With the #FTP\_RECV=<blocksize> command, you can transfer at most <blocksize> bytes onto the serial port. This number is limited to the current number of bytes of the remote file which have been transferred from the FTP server.

## 12.10. FTP Downloading Restart

To start a FTP downloading from a specific position (byte) of the file, use the #FTPREST command before typing in #FTPGET or #FTPGETPKT command. The command syntax is:

**AT#FTPREST=<restartposition>**

To get parameters descriptions see documents [1], [2], [4], [11], or [13], according to the used module.

**NOTE:** it is mandatory to issue `#FTPTYPE=0` command to set binary transfer type before entering `#FTPGET` or `#FTPGETPKT` command.

## 12.11. Examples

### 12.11.1. File Downloading in ONLINE Mode

Enable reports in verbose format.

```
AT+CMEE=2
OK
```

Assume that the PDP context `<cid>=1` has been bound to a socket by means of the `#SCFG` command. Now, define PDP context `<cid>=1`.

```
AT+CGDCONT=1,"IP", "Access_Point_Name"
OK
```

Activate the PDP Context. The command returns the IP address assigned by the network to the module.

```
AT#SGACT=1,1
#SGACT: 193.199.234.255
OK
```

Set the FTP time-out.

```
AT#FTPTO=1000
OK
```

Before using `#FTPOPEN` command, the PDP context identified by `<cid>=1`, mandatorily assigned to the FTP Application, must be defined. Then `<cid>=1` must be bound to a Multi-socket and activated through `#SGACT` command to create the related Network Interface. The `#FTPOPEN` command uses a Protocol Service Socket, but `#SGACT` command requires the PDP context bound to a Multi-socket.

Open a FTP connection toward the FTP server. The FTP server port is not specified, therefore the command uses the default value 21. The FTP connection is in active mode (default).

```
AT#FTPOPEN="server","username","password"
OK          ← FTP control connection open, and in active mode.
```

Set the file type.

```
AT#FTPTYPE=0
OK
```

Check the working directory.

```
AT#FTPPWD
#FTPPWD: 257 "/"
OK
```

Use `#FTPLIST` command to get the list of files on the working directory of the FTP server.

Use the #FTPGET command to open a data connection and download a file from the FTP server.

**AT#FTPGET="filename.txt"**

CONNECT ← data connection is open.

... the content of the file appears on terminal emulator ...

NO CARRIER ← data connection is closed automatically when the downloading is terminated.

Close FTP control connection.

**AT#FTPCLOSE**

OK ← FTP control connection is closed.

Deactivate the PDP context.

**AT#SGACT=1,0**

OK

### 12.11.2. File Downloading in COMMAND Mode

Assume that the FTP connection is open. Use the #FTPGETPKT command to open a data connection and download a file from the FTP server. Data are buffered on the socket; the module remains in COMMAND mode.

**AT#FTPGETPKT="filename.txt"**

OK ← data connection is open.

The following command reports the number of bytes buffered on the socket.

**AT#FTP\_RECV?**

#FTP\_RECV: 600

OK

Read the first 400 bytes of the available buffered data.

**AT#FTP\_RECV=400**

#FTP\_RECV: 400

Text row number 1 \* 11111111111111111111111111111111 \*

Text row number 2 \* 22222222222222222222222222222222 \*

Text row number 3 \* 33333333333333333333333333333333 \*

Text row number 4 \* 44444444444444444444444444444444 \*

Text row number 5 \* 55555555555555555555555555555555 \*

Text row number 6 \* 66666666666666666666666666666666 \*

Text row number 7 \* 77777777777777777777777777777777 \*

Text row number 8 \* 88888888888888888888888888888888

OK

Read 200 bytes – if available – starting from the position + 1 of the last byte read with the previous #FTP\_RECV command.

**AT#FTP\_RECV=200**

#FTP\_RECV: 200

88888 \*

Text row number 9 \* 99999999999999999999999999999999 \*

Text row number 10 \* AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA \*

Text row number 11 \* BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB \*

Text row number 12 \* CCCCCCCCCCCCCCCCCC

OK

The #FTPGETPKT? read command reports the download state.

**AT#FTPGETPKT?**

#FTPGETPKT: filename.txt,0,1

OK

The first parameter is the file name, the second indicates text or hex mode. The third parameter indicates <EOF> (End of File): 0 file transfer is in progress; 1 file transfer is ended.

Data connection (data port) is automatically closed by the last AT#FTPGETPKT? read command when the whole file has been read.

A new FTP download in ONLINE or COMMAND mode can be started by issuing respectively #FTPGET or #FTPGETPKT command.

### 12.11.3. FTP and 4G Modules

As stated before, in 2G/3G modules the FTP Service uses the <cid>=1 (or <cid>=0). In 4G modules the <cid>=1 is used by the Default EPS Bearer created during the attach procedure. The 4G modules provide the #PROTOCOLCFG command which allows to configure the <cid> used by the FTP Service. This example shows a case in which the APN assigned by the network, during the attach procedure, does not support data traffic, therefore it is required to move the FTP service, for example, on <cid>=2.

Check if the module is registered on a 4G network.

**AT+COPS?**

+COPS: 1,0,"Network\_Operator",7 ← 7 = E-UTRAN

OK

Enable ERROR report in verbose format.

**AT+CMEE=2**

OK

By default, the <cid>=1 context identifier is assigned to a not defined APN

**AT+CGDCONT?**

+CGDCONT: 1,"IPV4V6","", "",0,0

OK

When the network recognizes the attach request with a not defined APN, the network assigns to <cid>=1 an APN supporting some services. The module is 4G attached.

**AT+CGATT?**

+CGATT: 1 ← the module is 4G attached

OK

After the Attach procedure, the <cid>=1 identifies the Default EPS Bearer assigned by the network, and it is already activated.

**AT+CGACT?**

+CGACT: 1,1

OK

**AT+CGCONTRDP=?**

+CGCONTRDP: (1) ← <cid>=1 is associated to the active context.  
OK

List the parameters regarding the Default EPS Bearer assigned by the network and associated to <cid>=1 context identifier.

**AT+CGCONTRDP=**

+CGCONTRDP:  
1,  
5,  
"APN\_Assigned\_by\_Network",  
"10.178.0.31.255.0.0.0",  
"10.178.0.32","213.230.129.10",  
"0.0.0.0",  
"0.0.0.0",  
"0.0.0.0"

Check the <cid> value used for each protocol, default configuration.

**AT#PROTOCOLCFG?**

#PROTOCOLCFG: "FTP",1,0,0,0  
#PROTOCOLCFG: "SMTP",1,0,0,0  
#PROTOCOLCFG: "PING",1,0,0,0  
#PROTOCOLCFG: "SSL",1,0,0,0  
OK

The Default EPS Bearer identified by <cid>=1 is already active, contrary to what is indicated by the #SGACT command below. This means that a Network Interface has not been created.

**AT#SGACT?**

#SGACT: 1,0  
OK

Try to open a FTP connection (control connection). The opening fails because the Network Interface has not been created.

**AT#FTPOPEN="server","username","password"**

+CME ERROR: Cannot resolve name

The <cid>=1 used by the following #SGACT command is already active. Its activation is the result of the default bearer context activation procedure part of the attach procedure. The command below returns the IP address shown by the +CGCONTRDP= command and opens only a Network Interface.

**AT#SGACT=1,1**

#SGACT: 10.178.0.31  
OK

Try to open the FTP connection. Now, the opening fails again because the APN assigned by the network on <cid>=1 does not support data traffic, see Fig. 7.

**AT#FTPOPEN="server","username","password"**

+CME ERROR: Cannot connect control socket

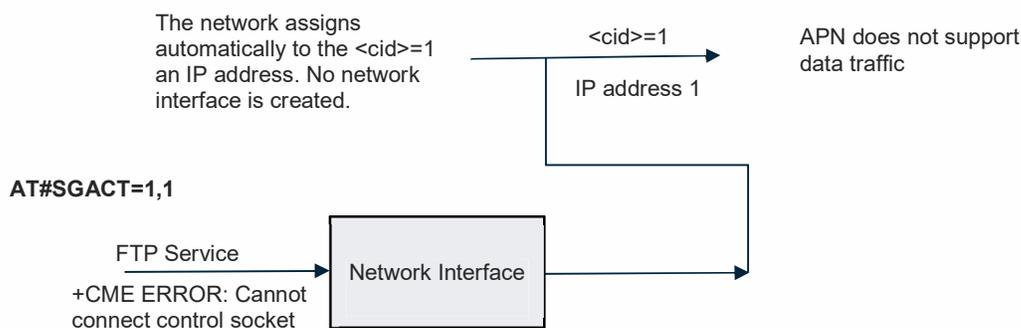


Fig. 7: Assigned APN does not Support Data Traffic

It is required to change the <cid> used by the FTP Service and define a PDP context identified by the <cid> associated to the FTP Service.

```
AT#PROTOCOLCFG="FTP",2
OK
```

Verify the new configuration.

```
AT#PROTOCOLCFG?
#PROTOCOLCFG: "FTP",2,0,0,0
#PROTOCOLCFG: "SMTP",1,0,0,0
#PROTOCOLCFG: "PING",1,0,0,0
#PROTOCOLCFG: "SSL",1,0,0,0
OK
```

Define the PDP context identified by <cid>=2.

```
AT+CGDCONT=2,"IP"," Access_Point_Name ","0.0.0.0"
OK
```

```
AT+CGDCONT?
```

```
+CGDCONT: 1,"IPV4V6","", "",0,0
+CGDCONT: 2,"IP","Access_Point_Name","0.0.0.0",0,0
OK
```

Activate PDP context <cid>=2 and create a Network Interface related to <cid>=2.

```
AT#SGACT=2,1
#SGACT: 2.192.17.237
OK
```

Check which contexts are active.

```
AT#SGACT?
#SGACT: 1,1
#SGACT: 2,1
OK
```

Open a FTP connection (control connection) toward the FTP server. The FTP server port is not specified; therefore, the command uses the default value 21. The FTP connection is in active mode (default), see Fig. 8.

**AT#FTPOPEN="server","username","password"**  
 OK ← FTP control connection open in active mode.

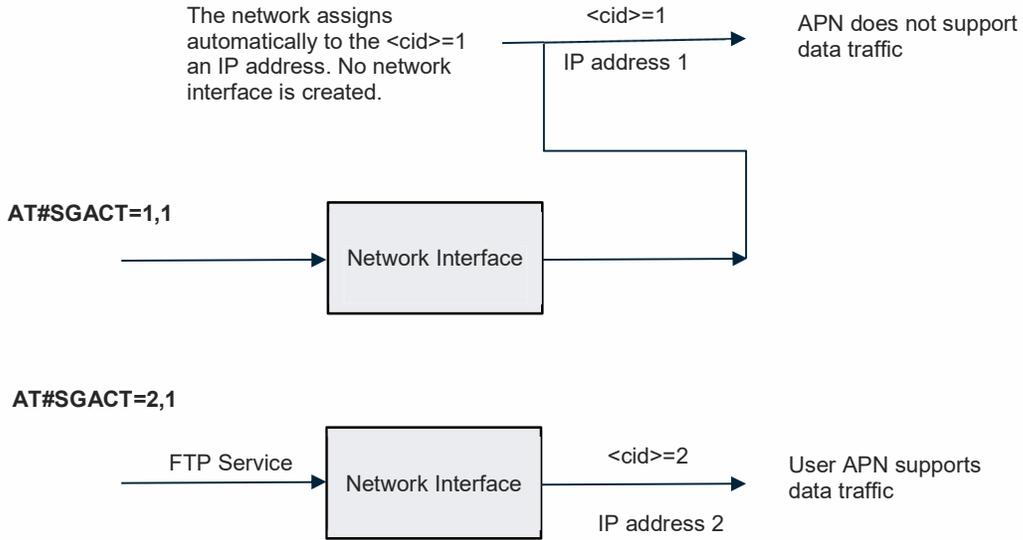


Fig. 8: User APN Support Data Traffic

Use the FTP Service as shown for 2G/3G modules.

## 13. PING PROTOCOL

Ping protocol is used to measure the round-trip time for messages sent by the module to the remote host that are echoed back to the sender.

### 13.1. 2G/3G Modules

The PING connection can use only the PDP context identified by <cid>=1, or the GSM context identified by <cid>=0. The PING command uses one of the two contexts according to the command previously entered: AT#SGACT=1,1 or AT#SGACT =0,1 respectively.

Referring to the example shown in Tab. 23:

- PING Protocol Service Socket bound to <cid>=1 (mandatory)
- Sockets <connId>= 1, 2, 3 not opened, and bound to the active <cid>=1
- Sockets <connId>= 4, 5, 6 not opened, and bound to the active <cid>=3

IP Easy Multi-Socket <connId>	<cid>=1	<cid>=2	<cid>=3	<cid>=4	<cid>=5
	Active	No active	Active	No active	No active
1	bound	/	/	/	/
2	bound	/	/	/	/
3	bound	/	/	/	/
4	/	/	bound	/	/
5	/	/	bound	/	/
6	/	/	bound	/	/
<b>Protocol Service Sockets</b>					
PING	bound				

Tab. 23: 2G/3G Multi-Socket & PING Service Socket

Use #PING command to send ICMP Echo Request packets to the specified host and wait for an ICMP Echo Reply. The command syntax is:

**AT#PING=<IPaddr>[,<retryNum>[,<len>[,<timeout>[,<ttl>]]]]**

Echo Replay message format:

#PING: <replyId>,<IP Address>,<replyTime>,<ttl>

Use #ICMP command to enable ICMP Echo Replay. The command syntax is:

**AT#ICMP=<mode>**

To get parameters descriptions see documents [1], or [2], according to the used module.

#### Example

It is not mandatory to use #ICMP before #PING to receive the ICMP Echo Replay.

Check ICMP Ping support.

**AT#ICMP?**

#ICMP: 0 ← support disabled (default)

OK

Define PDP context <cid>=1.

**AT+CGDCONT=1,"IP","Access\_Point\_Name"**

OK

Activate the PDP context <cid>=1. The command returns the IP address assigned by the network to the module.

**AT#SGACT=1,1**

#SGACT: 10.160.118.113

OK

Before using #PING command, the PDP context identified by <cid>=1, mandatorily assigned to the PING Application, must be defined. Then <cid>=1 must be bound to a Multi-socket and activated through #SGACT command to create the related Network Interface. The #PING command uses a Protocol Service Socket, but #SGACT command requires the PDP context bound to a Multi-socket.

Ping the Primary Google DNS.

**AT#PING="8.8.8.8"**

#PING: 01,"8.8.8.8",600,255

#PING: 02,"8.8.8.8",5,55

#PING: 03,"8.8.8.8",5,55

#PING: 04,"8.8.8.8",3,55

OK

## 13.2. 4G Modules

4G modules provide the #PROTOCOLCFG command to configure which <cid> the PING Application can use, in accordance with the ISP requirements. In addition, the <cid> range can vary according to the 4G product you are using. Refer to chapter 9.2.1 (#PROTOCOLCFG) to change the <cid> identifier bound to PING Application.

Use #PING command to send ICMP Echo Request packets to the specified host and wait for an ICMP Echo Reply. The command syntax is:

**AT#PING=<IPaddr>[,<retryNum>[,<len>[,<timeout>[,<tTl>]]]]**

Echo Replay message format:

#PING: <replyId>,<IP Address>,<replyTime>,<tTl>

Use #ICMP command to enable ICMP Echo Replay. The command syntax is:

**AT#ICMP=<mode>**

To get parameters descriptions see documents [4], [11], or [13] according to the used module.

## 14. EASY GSM: IP OVER CSD

In the Easy GSM configuration, the user device, connected by a serial line to the Telit module, exchanges data, over a GSM CSD call, with a remote device on Internet network, refer to Fig. 9.

Telit module establishes a GSM CSD call, and exchanges TCP/IP packets, encapsulated in the PPP protocol, with the remote device. The data are sent as if it were a voice signal, the CSD call occupies the radio channels during the entire duration of the connection, regardless of whether data are transmitted or not. The user device does not need an own TCP/IP stack.

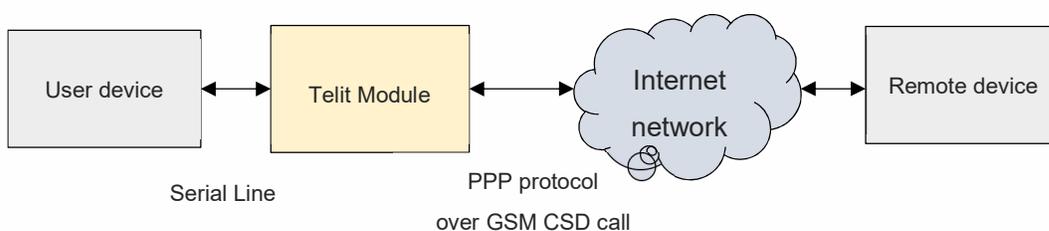


Fig. 9: GSM CSD Call

GSM CSD call supports #SD command (IP Easy Environment), HTTP, SMTP, FTP, and PING connections. To establish a GSM CSD call, must be defined and activated the GSM context identified by <cid>= 0.

This section describes the AT commands to configure and activate the GSM context used to establish a GSM CSD call. For information on outgoing and incoming connections, refer respectively to chapter 6, and chapter 7. There are no differences at socket level.

Here are the AT commands to define, bind, and activate the GSM context type.

- AT#GSMCONT to configure the GSM context identified by <cid> = 0
- AT#SCFG to bind the GSM context to a socket belonging to the Multi-socket environment.
- AT#SGACT to activate the GSM Context.

### 14.1. GSM Context Setting

The GSM context is configured by the #GSMCONT command, its syntax is:

**AT#GSMCONT=<cid>[,<P\_type>, <CSD\_num>]**

To get parameters descriptions see documents [1], or [2], according to the used module.

## 14.2. GSM Context Activation

GSM context type is activated through the #SGACT command using <cid> = 0. The activation may require the authentication parameters (User Name and Password), depending on the Internet Service Provider. The command syntax is the same used for PDP context type.

**AT#SGACT= 0,<stat>[,<userID>,<pwd>]**

### NOTES

- I. When the GSM context is active, PDP contexts are not allowed because Telit modules work in Class B. Therefore, if a GSM CSD call is on, no GPRS operation is allowed, see example in chapter 14.3.1.
- II. GSM context activation is affected by the +CBST command, like all CSD calls. The maximum data rate that can be set through +CBST command is 14400 bps (network dependent). Context activation is just allowed in "NON TRANSPARENT" mode by default.
- III. The +COPS=? and #CSURV commands return an error message if a data call is active. Therefore, they return an error message also if a GSM context is active; see example 14.3.2.

## 14.3. Examples

### 14.3.1. GSM/PDP Contexts & FTP Protocol

This example shows that when the GSM context (<cid>=0) is activated, it is not allowed to use PDP contexts. To show this, the example tries to open two socket connections, and a FTP connection.

Enable error report in verbose format.

**AT+CMEE=2**

OK

Check the sockets/PDP contexts binding configuration (default).

**AT#SCFG?**

#SCFG: 1,1,300,90,600,50

#SCFG: 2,1,300,90,600,50

#SCFG: 3,1,300,90,600,50

#SCFG: 4,2,300,90,600,50

#SCFG: 5,2,300,90,600,50

#SCFG: 6,2,300,90,600,50

OK

Assume that the PDP contexts have been set as shown below.

**AT+CGDCONT?**

+CGDCONT: 3,"IP","Access\_Point\_Name","0.0.0.0",0,0

+CGDCONT: 4,"IP","Access\_Point\_Name","0.0.0.0",0,0

OK

Activate PDP context <cid>=3. The command returns an error message because the PDP context is not bound to any sockets. The binding is mandatory.

**AT#SGACT=3,1**

+CME ERROR: operation not supported

Before activating PDP contexts <cid>=3 and <cid>=4, bind them respectively to the socket <connId>=1 and <connId>=5.

**AT#SCFG = 1, 3, 300, 90, 600, 50**

OK

**AT#SCFG = 5, 4, 300, 90, 600, 50**

OK

Check the new sockets/PDP contexts binding configuration.

**AT#SCFG?**

#SCFG: 1,3,300,90,600,50

#SCFG: 2,1,300,90,600,50

#SCFG: 3,1,300,90,600,50

#SCFG: 4,2,300,90,600,50

#SCFG: 5,4,300,90,600,50

#SCFG: 6,2,300,90,600,50

OK

Define GSM context <cid>=0.

**AT#GSMCONT=0,"IP","ISP\_phone\_number"**

OK

Check GSM context.

**AT#GSMCONT?**

#GSMCONT: 0,"IP","ISP\_phone\_number"

OK

Activate GSM context <cid>=0. The command returns an error message because the GSM context is not bound to any sockets. The binding is mandatory.

**AT#SGACT=0,1**

+CME ERROR: operation not supported

Before activating GSM context <cid>=0 bind it to the socket 3.

**AT#SCFG = 3, 0, 300, 90, 600, 50**

OK

Check the new sockets/contexts binding configuration.

**AT#SCFG?**

#SCFG: 1,3,300,90,600,50

#SCFG: 2,1,300,90,600,50

#SCFG: 3,0,300,90,600,50

#SCFG: 4,2,300,90,600,50

#SCFG: 5,4,300,90,600,50

#SCFG: 6,2,300,90,600,50

OK

If PDP contexts are activated before the activation of the GSM context, no error message is returned by the #SGACT.

**AT#SGACT = 3,1**

← activate the PDP context 3

#SGACT: 109.112.140.34

OK

**AT#SGACT = 4,1** ← activate the PDP context 4  
 #SGACT: 37.159.24.75  
 OK

**AT#SGACT = 0,1** ← activate the GSM context 0  
 #SGACT: 10.141.94.204  
 OK

Check which contexts are active.

**AT#SGACT?**  
 #SGACT: 0,1  
 #SGACT: 3,1  
 #SGACT: 4,1  
 OK

Make sure that the server is in listening mode and try to open the socket connection <connId>=1 toward it, the command return an error message.

**AT#SD=1,0,1024,"server\_address"**  
 +CME ERROR: timeout in opening socket

Try to open the socket connection <connId>=5, the command returns an error message.

**AT#SD=5,0,1024,"server\_address"**  
 +CME ERROR: timeout in opening socket

Try to open a FTP control connection toward a FTP server. The #FTOPEN command uses the GSM context; refer to chapter 12. The connection works.

**AT#FTOPEN="server","username","password"**  
 OK ← FTP control connection is open, and in active mode.

To open successfully a socket connection (in this example <connId>=1, or <connId>=5) you must close the FTP connection and deactivate the GSM context.

Close the FTP connection.

**AT#FTPCLOSE**  
 OK

Deactivate the GSM context.

**AT#SGACT=0,0**  
 OK

**AT#SGACT?**  
 #SGACT: 0,0  
 #SGACT: 3,1  
 #SGACT: 4,1  
 OK

For example, open the socket connection 5.

**AT#SD=5,0,1024,"server\_address"**  
 CONNECT ← the socket connection works.

.....  
 .....

### 14.3.2. GSM Context & +COPS=? Command

Enable error report in verbose format.

```
AT+CMEE=2  
OK
```

Check the current sockets/PDP contexts binding configuration (default).

```
AT#SCFG?  
#SCFG: 1,1,300,90,600,50  
#SCFG: 2,1,300,90,600,50  
#SCFG: 3,1,300,90,600,50  
#SCFG: 4,2,300,90,600,50  
#SCFG: 5,2,300,90,600,50  
#SCFG: 6,2,300,90,600,50  
OK
```

Before activating GSM context (<cid>=0) bind it to the socket 3.

```
AT#SCFG = 3, 0, 300, 90, 600, 50  
OK
```

Check the new sockets/PDP contexts binding configuration.

```
AT#SCFG?  
#SCFG: 1,1,300,90,600,50  
#SCFG: 2,1,300,90,600,50  
#SCFG: 3,0,300,90,600,50  
#SCFG: 4,2,300,90,600,50  
#SCFG: 5,2,300,90,600,50  
#SCFG: 6,2,300,90,600,50  
OK
```

Define GSM context.

```
AT#GSMCONT=0,"IP","ISP_phone_number"  
OK
```

Check GSM context.

```
AT#GSMCONT?  
#GSMCONT: 0,"IP","ISP_phone_number"  
OK
```

AT+COPS=? test command returns information about the operators present on the network.

```
AT+COPS=?  
+COPS: .....  
OK
```

Now, activate the GSM context.

```
AT#SGACT = 0,1  
#SGACT: 10.141.94.204  
OK
```

Type in again AT+COPS=? test command. In this case, it returns an error message.

```
AT+COPS=?  
+CME ERROR: no network service
```

### 14.3.3. HTTP Client with #SD

This example describes how establish a CSD call, create an HTTP connection toward an HTTP server, and retrieves an HTML page using a socket dial connection (#SD). The socket connection is running over the CSD call. A terminal emulator is connected to the module. See also the example 10.4.4, just for comparison.

Define GSM context.

```
AT#GSMCONT = 0,"IP","ISP_phone_number"
```

```
OK
```

Before activating GSM context (<cid>=0) bind it to the socket 6.

```
AT#SCFG = 6, 0, 300, 90, 600, 50
```

```
OK
```

Check the sockets/contexts binding configuration.

```
AT#SCFG?
```

```
#SCFG: 1,1,300,90,600,50
```

```
#SCFG: 2,1,300,90,600,50
```

```
#SCFG: 3,1,300,90,600,50
```

```
#SCFG: 4,2,300,90,600,50
```

```
#SCFG: 5,2,300,90,600,50
```

```
#SCFG: 6,0,300,90,600,50
```

```
OK
```

Activate the GSM context, and assume that no UserID and Password are needed. The command returns the IP address assigned by the network.

```
AT#SGACT=0,1
```

```
#SGACT: 10.137.93.60
```

```
OK
```

Open the connection toward the HTTP server on port=80. Transport protocol is TCP.

```
AT#SD=6,0,80,"HTTP_server_address",0,0
```

```
CONNECT ← the connection is open. The module is in ONLINE mode.
```

Now, following the HTTP protocol, ask the homepage by sending the next lines to the module. The entered data are sent directly to the HTTP server. The strings sent to the HTTP server must be ended by line feed character. To see the issued commands, enable the local echo on terminal emulator.

```
GET / HTTP/1.1<cr><lf>
```

```
Host: www.xxx.com<cr><lf>
```

```
Connection: keep-alive<cr><lf>
```

```
<cr><lf>
```

The HTTP server replies with the HTML code of the homepage and some debugging responses.

```
HTTP/1.1 200 OK
```

```
Date: Thu, 06 2003 10:21:58 GMT
```

```
Server: Apache/1.3.27 (Unix)
```

```
Last-Modified: Thu, 06 2003 10:21:58 GMT
```

```
Content-Type: text/html
```

```
Connection: close
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 FINAL//EN">
<HTML>
... here is all the HTML code of the page ...
</HTML>
```

**+++** ← type in the escape sequence to suspend the connection.

OK ← the module is in COMMAND mode.

**AT#SH=6**

OK ← the connection is closed; the GSM context is still active.

#### 14.3.4. FTP File Uploading

This example describes how to establish a CSD call, create a FTP connection toward a FTP server, and upload a file. The FTP connection is running over the CSD call. A terminal emulator is connected to the module.

Check the socket configuration (default).

**AT#SCFG?**

```
#SCFG: 1,1,300,90,600,50
#SCFG: 2,1,300,90,600,50
#SCFG: 3,1,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,2,300,90,600,50
OK
```

Check that no PDP contexts are defined.

**AT+CGDCONT?**

OK

Bind socket <connId>=4 to GSM context.

**AT#SCFG = 4,0,300, 90, 600, 50**

OK

Check the new configuration

**AT#SCFG?**

```
#SCFG: 1,1,300,90,600,50
#SCFG: 2,1,300,90,600,50
#SCFG: 3,1,300,90,600,50
#SCFG: 4,0,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,2,300,90,600,50
OK
```

Define the GSM context.

**AT#GSMCONT=0,"IP", "ISP\_phone\_number"**

OK

AT#GSMCONT?

**#GSMCONT: 0,"IP", "ISP\_phone\_number"**

OK

Activate the GSM context. The command returns the IP address assigned by the network to the module.

```
AT#SGACT=0,1
#SGACT: 10.137.93.60
OK
```

Set FTP time out

```
AT#FTPTO=1000
OK
```

Open a FTP control connection toward the FTP server in active mode. The FTP server port is not specified, therefore the command uses the default value 21.

```
AT#FTPOPEN="server","username","password"
OK ← FTP control connection is open.
```

```
AT#FTPTYPE=1 ← set ASCII file type.
OK
```

The following command opens the data connection, and the module enters ONLINE mode. filename.txt is the file name where the data will be stored on the FTP server.

```
AT#FTPPUT="filename.txt"
CONNECT ← the data connection is open.
```

... type in the data to write in the filename.txt file stored on the FTP server ...

```
+++ ← close the data connection.
```

```
NOCARRIER ← the module is in COMMAND mode.
```

```
AT#FTPCLOSE ← close FTP control connection.
OK
```

```
AT#SGACT=0,0 ← deactivate GSM context
OK
```

#### 14.3.5. E-mail with no Attachment

This example describes how to establish a CSD call, create a SMTP connection, and send an e-mail.

Before activating GSM context (<cid>=0), for example, bind it to the socket 6.

```
AT#SCFG = 6, 0, 300, 90, 600, 50
OK
```

Check the sockets/contexts binding configuration.

```
AT#SCFG?
#SCFG: 1,1,300,90,600,50
#SCFG: 2,1,300,90,600,50
#SCFG: 3,1,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,0,300,90,600,50
OK
```

Define GSM context.

```
AT#GSMCONT = 0,"IP","ISP_phone_number"
OK
```

Activate the GSM context, and assume that no UserID and Password are needed. The command returns the IP address assigned by the network.

```
AT#SGACT=0,1
#SGACT: 10.137.93.60
OK
```

Set the SMTP connection parameters.

```
AT#SMTPCFG=1,587,1
OK
```

```
AT#ESMTP="SMTP_server_address" ← configure the SMTP server address of the sender.
```

```
OK
```

```
AT#EADDR="sender_address" ← configure e-mail address of the sender.
```

```
OK
```

```
AT#EUSER="sender_user_name" ← configure user name of the sender.
```

```
OK
```

```
AT#EPASSW="sender_user_password" ← configure user password of the sender.
```

```
OK
```

Send the e-mail with no attachment to the recipient having recipient\_address.

```
AT#EMAILD="recipient_address","mail with no attachment"
> Hello<Ctrl-Z>
```

```
OK ← text of the e-mail has been sent successfully.
```

#### 14.3.6. Modules in Server and Client Configuration

This example describes a remote connection between two modules. The first one is configured as a server that opens a socket connection in listen mode. The second one is configured as a client, it establishes a CSD call and opens a connection using the #SD command - over the CSD call - toward the server. Each module is connected to a terminal emulator. See also the example described in chapter 7.5.1, just for comparison.

##### Server Side

Check the sockets/PDP contexts binding configuration (default).

```
AT#SCFG?
#SCFG: 1,1,300,90,600,50
#SCFG: 2,1,300,90,600,50
#SCFG: 3,1,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,2,300,90,600,50
OK
```

Only PDP context <cid>=1 has been set, all other contexts have been removed.

**AT+CGDCONT?**

+CGDCONT: 1,"IP","Access\_Point\_Nam ","0.0.0.0",0,0  
OK

Activate the PDP context <cid>=1, and no UserID and Password are needed. The command returns the IP address assigned by the network. The client, to open the connection, will use this IP address. See client-side chapter.

**AT#SGACT=1,1**

#SGACT: 2.192.8.30  
OK

Define an ACCEPT firewall chain to accept IP address of the client. See client side.

**AT#FRWL=1," 10.141.94.204","0.0.0.0"**

OK

Open <connId>=2 socket in listening mode on <port>=1024. The client will use this port.

**AT#SL=2,1,1024**

OK

When the client opens the connection, the server displays on the terminal emulator a SRING unsolicited indication with <connId>=2.

SRING: 2

Enter the following command to accept the connection 2.

**AT#SA=2**

CONNECT

... exchange data...

NO CARRIER ← on the client side has been entered #SH command, the connection is closed.

### Client Side

Check the sockets/PDP contexts binding configuration (default configuration)

**AT#SCFG?**

#SCFG: 1,1,300,90,600,50  
#SCFG: 2,1,300,90,600,50  
#SCFG: 3,1,300,90,600,50  
#SCFG: 4,2,300,90,600,50  
#SCFG: 5,2,300,90,600,50  
#SCFG: 6,2,300,90,600,50  
OK

Before activating GSM context (<cid>=0) bind it to the socket 6.

**AT#SCFG = 6, 0, 300, 90, 600, 50**

OK

Check the new configuration

**AT#SCFG?**

#SCFG: 1,1,300,90,600,50

```
#SCFG: 2,1,300,90,600,50
#SCFG: 3,1,300,90,600,50
#SCFG: 4,2,300,90,600,50
#SCFG: 5,2,300,90,600,50
#SCFG: 6,0,300,90,600,50
OK
```

```
Define GSM Context (<cid>=0)
AT+CGDCONT=0,"IP","ISP_phone_number"
OK
```

```
Check GSM context.
AT#GSMCONT?
#GSMCONT: 0,"IP","ISP_phone_number"
OK
```

Activate the GSM Context, no UserID and Password are needed. The command returns the IP address assigned by the network to the module. The server will use this IP address to set its firewall. See server side.

```
AT#SGACT=0,1
#SGACT: 10.141.94.204
OK
```

Open the connection toward the server using its IP address and port 1024, see sever side.

```
AT#SD=6,0,1024,"2.192.8.30"
CONNECT ← the module is in ONLINE mode.
... exchange data...
```

```
+++ ← suspend the connection.
OK ← the module is in COMMAND mode.
```

```
AT#SH=6 ← close the connection. On server side is displayed the NO CARRIER
message.
```

```
OK
```

Use #CGPADDR= command to verify the IP address assigned by the network to the modules after the GSM/PDP contexts activations.

## 15. DIALUP CONNECTION

The `ATD*99***<cid>#` command establishes a PSD connection between the module and an external PDN over the radio interface. This type of connection is initiated by using a character string instead of an ISP phone number, see chapter 14 for comparison only. After entering the dialup command, the data exchange between the user device and the module uses the PPP protocol that provides connection, authentication, transmission encryption, and compression, refer to Fig. 10.

Dial up connection and Multi-socket connections can be active at the same time, but using different PDP context identifiers. In this configuration, the performances get worse. Usually, the dial up connection is used to make Web browsing.

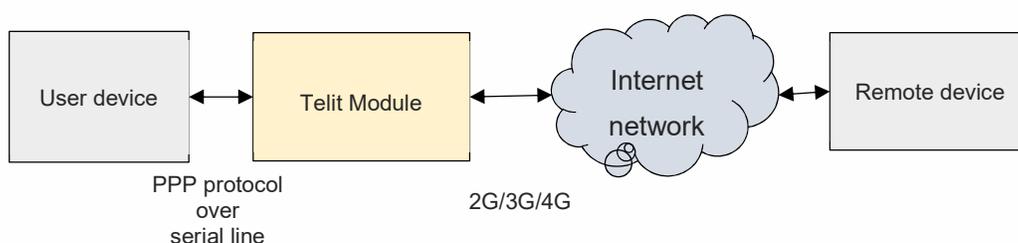


Fig. 10: GPRS Dialup Connection

The command syntax is:

**ATD\*99\*\*\*<cid>#**

Where:

**99** GPRS Service Code  
**<cid>** identifies the PDP context. To know the **<cid>** range see **+CGDCONT** command.

### 15.1. Examples

#### 15.1.1. Start PPP Protocol

Assume that the PDP context `<cid>=3` and the QoS have been configured. Establish a dial up connection using `<cid>=3`.

**ATD\*99\*\*\*3#**

CONNECT ← the module is in ONLINE mode.

At this point, the user application must start the PPP protocol, which triggers the PDP context activation.

→ LCP Configure Request  
 ← LCP Configure Acknowledge

→ PAP Authentication  
 ← PAP-Ack

→ NCP (IP) Configure Request  
 ← NCP (IP) Configure Acknowledge

..... start data packets exchange .....

Assume that the data packet exchange is ended. The user application must send the Terminate Request message.

→ LCP Terminate Request  
 ← LCP Terminate Acknowledge

Wait for NO CARRIER response. In alternative, send escape sequence.

+++            ← Use the +++ escape sequence to suspend the data connection. The module exits the ONLINE mode and enters the COMMAND mode. See the ATS12 command to set timing relevant to escape sequence  
 OK             ← the module is in COMMAND mode.

Close the data call.

**ATH**  
 OK



Refer to:

- RFC1661 to get LCP protocol and PPP protocol information.
- RFC1334 to get PAP protocol information.
- RFC1332 to get IPCP protocol information.

TCP/IP and PPP protocols description is beyond the scope of this document.

---

### 15.1.2. Dial-up Using <cid>=1 by default (2G)

This example shows ATD\* command using <cid>=1 by default. The module is connected to a 2G network.

Check the current network operator.

**AT+COPS?**  
 +COPS: 0,0,"network operator"  
 OK

Check the current cellular network.

**AT+WS46?**  
 +WS46: 12            ← 12 = GSM  
 OK

Check the current GPRS service attachment state.

**AT+CGATT?**  
 +CGATT: 1  
 OK

Define PDP context identified by <cid>=1.

**AT+CGDCONT=1,"IP","Access\_Point\_Name"**  
 OK

Type in the dialup command. The command uses the <cid>=1 by default.

**ATD\*99#**

CONNECT ← the module is in ONLINE mode.

At this point, the user application must start the PPP protocol, which triggers the PDP context activation.

.....  
.....

### 15.1.3. Dial-up Using <cid>=1 by default (3G)

This example shows the dialup command using <cid>=1 by default. The module is connected to a 3G network.

Check the current network operator.

**AT+COPS?**

+COPS: 1,0,"network operator",2 ← 2 = UTRAN

OK

Check the current cellular network.

**AT+WS46?**

+WS46: 25 ← GERAN and UTRAN

OK

Check the current GPRS service attachment state.

**AT+CGATT?**

+CGATT: 1

OK

Define PDP context identified by <cid>=1.

**AT+CGDCONT=1,"IP","Access\_Point\_Name"**

OK

Type in the dialup command. The command uses the <cid>=1 by default.

**ATD\*99#**

CONNECT ← the module is in ONLINE mode.

At this point, the user application must start the PPP protocol, which triggers the PDP context activation.

.....  
.....

#### 15.1.4. 4G Modules

##### 15.1.4.1. Dial-up & #DUALAPN Command (Platform Version ID 20)

At power on, the PDP context identified by <cid>=1 uses "IP" and "Access\_Point\_Name", as returned by the AT+CGDCONT? read command shown below. For example, use the COM1 serial port to type in the command.

**AT+CGDCONT?**

```
+CGDCONT: 1,"IP","Access_Point_Name","",0,0
OK
```

Check if the module is registered on a 4G network.

**AT+COPS?**

```
+COPS: 0,0,"Network_Operator",7      ← 7 = E-UTRAN
OK
```

Check if the Default EPS Bearer is active

**AT+CGCONTRDP=?**

```
+CGCONTRDP: (1)
OK
```

Check the #DUALAPN mode.

**AT#DUALAPN?**

```
#DUALAPN: 0,0,0,0      ← 0 mode, default.
OK
```

If you assign to <cid>=2 the same APN and PDP type already in use with <cid>=1, the <cid>=2 is routed to the <cid>=1, according to AT#DUALAPN=0 mode (default), refer to chapter 5.7. Therefore, the IP address of both <cid> will be the same, see the following AT commands sequence.

**AT+CGDCONT=2,"IP","Access\_Point\_Name"**

OK

**AT+CGDCONT?**

```
+CGDCONT: 1,"IP","Access_Point_Name","",0,0
+CGDCONT: 2,"IP","Access_Point_Name","",0,0
OK
```

Only the first PDP context is active.

**AT+CGPADDR=**

```
+CGPADDR: 1,"2.192.16.194"
+CGPADDR: 2,""
OK
```

Now, for example, type in AT#PORTCFG=3 command, and reboot the module. To have information on #PORTCFG command refer to documents [4] or [10].

After rebooting, use COM2 port to start a dialup connection using AT2 instance.

The following dial up command establishes a Default EPS Bearer, and uses the <cid>=2, see Fig. 11.

**ATD\*99\*\*\*2#**

```
CONNECT      ← Module is in ONLINE mode.
```

Now, your application can start the PPP protocol on COM2 port, refer to chapter 15.1.1.

Use COM1 port to enter the next command to verify that <cid> 1 and 2 have the same IP address. If the PPP activation is successful, the two context identifiers have the same IP address, as stated before.

```

AT+CGPADDR=
+CGPADDR: 1,"2.192.16.194"
+CGPADDR: 2,"2.192.16.194"
OK
    
```

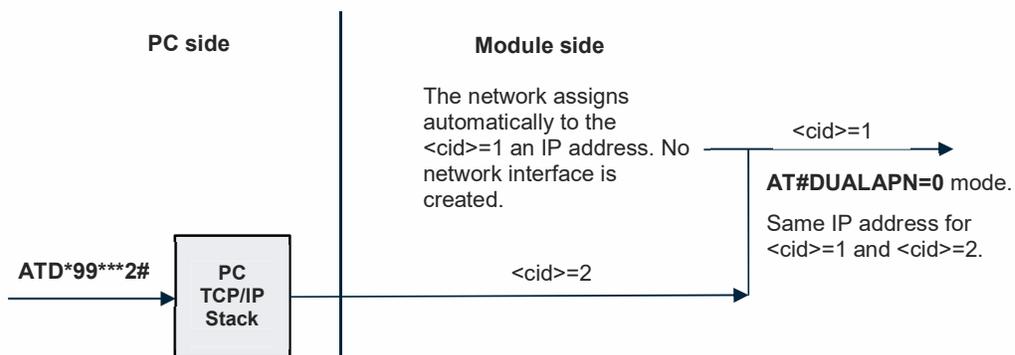


Fig. 11: Dial up connection using <cid>=2

#### 15.1.4.2. Dial-up Command, Platform ID 23

First, remember that:

- PPP is provided on Serial and USB ports. On USB port, PPP is supported only in router mode (AT#NWMODE=0,x,y. See chapter 16.7).
- AT#SGACT and Dial-up cannot run on the same <cid> (as stated in chapter 5.3).

Dial-up can be established using one of the following commands:

**ATD\*99#** default cid (cid 3 for LE866-SV1 and ME866A1-NV1, see chapter 5.2.2.1, or else cid 1, see chapter 5.2.2.2)

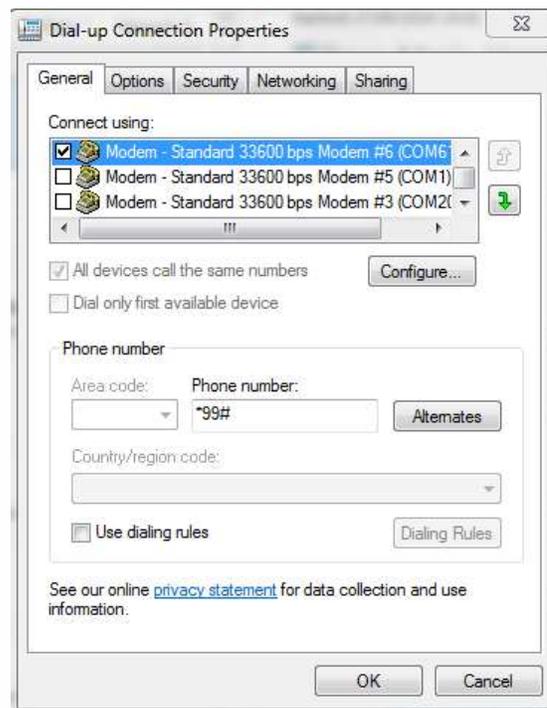
**ATD\*99\*\*1#** default cid (cid 3 for LE866-SV1 and ME866A1-NV1, see chapter 5.2.2.1, or else cid 1, see chapter 5.2.2.2)

**ATD\*99\*\*1\*<cid>#**

**ATD\*99\*\*\*<cid>#**

Assume to use a Windows host: dial-up connection uses Standard 33600 Modem, see the figure below.

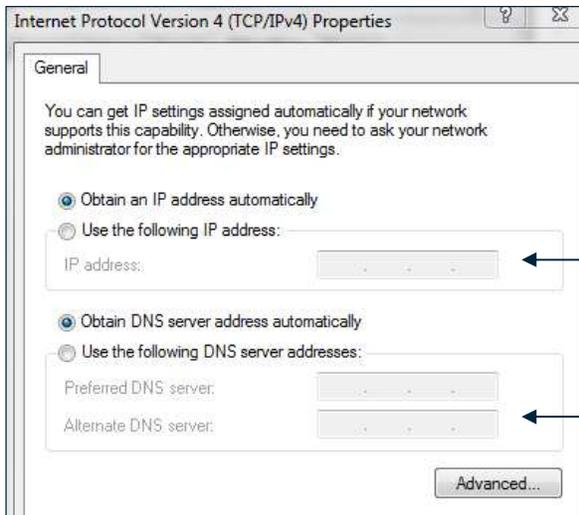
- Control Panel→Phone and Modem→Add Standard 33600 bps Modem selecting the AT interface to be used for Dial-up.
- Control Panel →Network and Sharing Center. Set up a new connection →Connect to Internet →Dial-up.



The IPv4 addresses, on the host side, is managed as described below. For the IPv6 addresses the host uses the global address provided by the network operator.

The PPP background process running on the module establishes a local network between the module and the host.

Property	Value
Device Name	Standard 33600 bps Modem #6
Device Type	modem
Authentication	(unknown)
Compression	(none)
PPP multilink framing	Off
Client IPv4 address	10.0.1.10
Server IPv4 address	2.193.7.157
NAP State	Not NAP-capable
Origin address	(unknown)
Destination address	(unknown)



**Server IP address** is the address, assigned to the module by the network operator, see AT#GPPPCFG command, document [11].

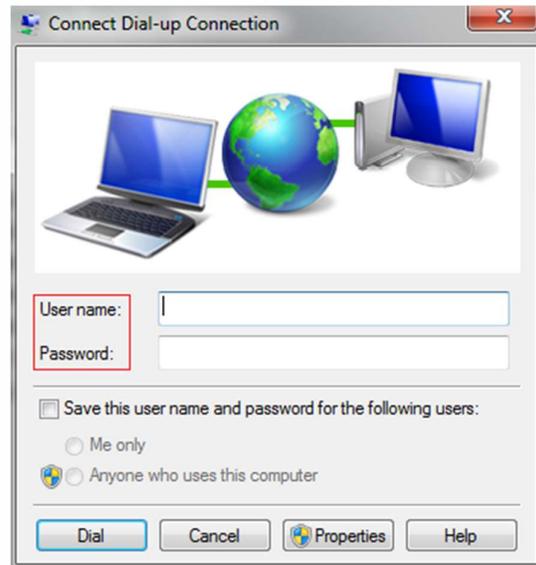
If the host application does not include the host IP address (**Client address**) in the IPCP Configuration Request message, by default it is 10.0.1.10,

If the host application does not include the DNS operator server address in the IPCP Configuration Request message, it is automatically assigned to the host.

## PPP Authentication

The modules support PAP and CHAP authentication.

- PPP authentication type is set through AT#GAUTH.
- PPP username and password should be saved in NVM by AT#GAUTHCFG.
- If PAP or CHAP is used, authentication is successfully managed if the host application uses the same credentials (**username** and **password**) saved in the module and these are accepted by operator during the PDP context activation automatically performed during dial-up.



## 16. APPENDIX

### 16.1. IPv6 Protocol

Tab. 24 shows the Platform Version ID supporting IPV4 (IP), IPV6, and the dual IP stack (IPV4V6, see chapter 16.2). Refer to the Applicability Table to see the correspondence between the Platform Version ID and the products series.

Platform Version ID	Internet Protocol Version		
	IP Easy environment	Dial UP connection	Easy GSM
	PDP Context Type		GSM Context Type
	AT+CGDCONT=?		AT#GSMCONT=?
10 (2G)	IP, IPV6	IP, IPV6	IP
13 (2G)	IP, IPV6	IP, IPV6	IP
16 (2G)	IP, IPV6	IP, IPV6	IP
12 (3G)	IP, IPV6, IPV4V6	IP, IPV6, IPV4V6	IP
20 (4G)	IP, IPV6, IPV4V6	IP, IPV6, IPV4V6	/
23 (4G)	IP, IPV6, IPV4V6	IP, IPV6, IPV4V6	/
25 (4G)	IP, IPV6, IPV4V6	IP, IPV6, IPV4V6	/

Tab. 24: Platform Version ID & IPV4/IPV6 Addresses

#### 16.1.1. 2G Modules

Assume to use a module having 2G technology and enter the AT+CGDCONT=? test command to get the supported parameters values. The returned message shows that the stack of the module can manage **IPV4** or **IPV6** protocol.

##### AT+CGDCONT=?

```
+CGDCONT: (1-5),"IP",,,(0,1),(0,1)
+CGDCONT: (1-5),"IPV6",,,(0,1),(0,1)
OK
```

#### 16.1.2. 3G Modules

Assume to use a module having 3G technology and enter the AT+CGDCONT=? test command to get the supported parameters values. The returned message shows that the module provides a dual IP stack (**IPV4V6**) that can manage at the same time IPV4 and IPV6 protocols.

##### AT+CGDCONT=?

```
+CGDCONT: (1-5),"IP",,,(0,1),(0,1)
+CGDCONT: (1-5),"IPV6",,,(0,1),(0,1)
+CGDCONT: (1-5),"IPV4V6",,,(0,1),(0,1)
OK
```

#### 16.1.3. 4G Modules

Assume to use a module having 4G technology and enter the AT+CGDCONT=? test command to get the supported parameters values. The returned message shows that the module provides a dual IP stack (**IPV4V6**) that can manage at the same time IPV4 and IPV6 protocols. Here is an example for LE910-EU1 module having Platform Version ID 20.

##### AT+CGDCONT=?

```
+CGDCONT: (1-15),"IP",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
+CGDCONT: (1-15),"IPV6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
```

```
+CGDCONT: (1-15),"IPV4V6",,,0,(0-4),(0,1),(0,1),(0,1),(0,1)
OK
```



Platform Version ID 23:  
the modules must be configured in bridge mode (see #NWMODE command, chapter 16.7) to work with IPV6.

## 16.2. IPV4, IPV6 and Dual Stack

### 16.2.1. 2G/3G/4G Modules

Referring to Fig. 4.

Define PDP context <cid>=1 using IPV4 ("IP").

```
AT+CGDCONT=1,"IP","Access_Point_Name"
```

OK

```
AT#SGACT = 1,1
```

← activate PDP context <cid>=1

```
#SGACT: 31.157.55.95
```

OK

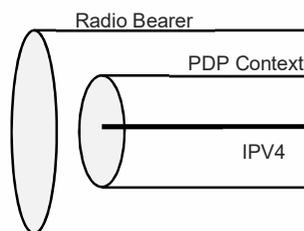


Fig. 12: IPV4

#SL command is listening only on IPV4 connections.

Referring to Fig. 13.

Define PDP context <cid>=2 using IPV6.

```
AT+CGDCONT=2,"IPV6","Access_Point_Name"
```

OK

```
AT#SGACT=2,1
```

← activate the PDP

```
#SGACT: 42.1.1.152.2.76.193.109.0.0.0.5.17.176.201.1
```

OK

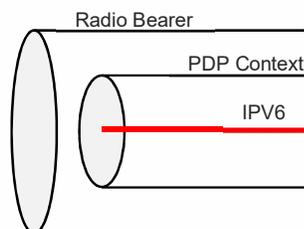


Fig. 13: IPV6

#SL command is listening only on IPV6 connections.

## 16.2.2. 3G/4G Modules

Referring to Fig. 6.

Define PDP context <cid>=1 using IPV4V6.

```
AT+CGDCONT=1,"IPV4V6","Access_Point_Name"
```

OK

```
AT#SGACT=1,1
```

← activate the PDP context 1

```
#SGACT: 31.157.55.95,42.1.1.152.2.76.193.109.0.0.0.5.17.176.201.1
```

OK

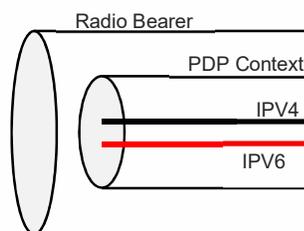


Fig. 14: Dual IP Stack IPV4V6

Using dual IP stack, #SL command is listening on IPV4 and IPV6. If the first received packet is an IPV4 packet, the connection will continue to use only IPV4 packets, and IPV6 packets are discarded. Conversely, if the first received packet is an IPV6 packet.

If you need to filter out some IP addresses use the firewall command, see chapter 7.4.

## 16.3. IPV6 Notation

The AT commands supporting the IPV6 address accept two IPV6 notation.

Dotted-decimal notation:

- xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx  
Where xxx is a number having the range: 000 - 255

Hexadecimal notation:

- yyyy:yyyy:yyyy:yyyy:yyyy:yyyy:yyyy:yyyy  
Where yyyy is expressed in hexadecimal format.

The AT commands return the IPV6 addresses always in the dotted-decimal notation.

### 16.3.1. +CGPIAF Command

4G modules provide the +CGPIAF command to select the format used to display the IPV6 address. The command syntax is:

```
AT+CGPIAF=[<IPV6_AddressFormat>,<IPV6_SubnetNotation>  
[,<IPV6_leadingZeros>[,<IPV6_compressZeros>]]]
```

To have information on the parameters meaning refer to documents [4], or [11] in accordance with the module you are using.

## 16.4. Examples

### 16.4.1. +CGDCONT Command & IPV6 Notation

Assume to use a 2G module and enter the +CGDCONT test command.

**AT+CGDCONT=?**

+CGDCONT: (1-5),"IP",,,(0,1),(0,1)

+CGDCONT: (1-5),"IPV6",,,(0,1),(0,1)

OK

Define PDP context <cid>=1 using IPV6.

**AT+CGDCONT=1,"IPV6","Access\_Point\_Name"**

OK

Activate the PDP Context. The command returns the IPV6 address assigned by the network to the module in dotted-decimal notation.

**AT#SGACT=1,1**

+IP: 42.1.1.152.2.76.193.109.0.0.0.5.17.176.201.1

OK

### 16.4.2. #SD Command & IPV6 Notation

Configure the socket.

**AT#SCFG=1,1,300,90,600,50**

OK

Open a socket connection using IPV6 in dotted-decimal notation.

**AT#SD=1,0,80,"32.1.7.0.0.1.46.0.0.0.0.15.112",0**

CONNECT

.....

Open a socket connection using IPV6 in hexadecimal notation.

**AT#SD=1,0,80," 2001:700:0:12e::f70",0**

CONNECT

.....

### 16.4.3. #FTPOPEN Command & IPV6 Notation

Open a FTP connection using IPV6 in dotted-decimal notation.

**AT#FTPOPEN="32.1.7.0.0.1.46.0.0.0.0.15.112",,,,,,1**

OK

Open a FTP connection using IPV6 in hexadecimal notation.

**AT#FTPOPEN="2001:700:0:12e::f70",,,,,,1**

OK

### 16.4.4. #PING Command & IPV6 Notation

Ping remote host using IPV6 in dotted-decimal notation.

```
AT#PING="32.1.7.0.0.0.1.46.0.0.0.0.0.15.112"
#PING: 01," 32.1.7.0.0.0.1.46.0.0.0.0.0.15.112",47,237
...
OK
```

Ping remote host using IPV6 in hexadecimal notation.

```
AT#PING="2001:700:0:12e::f70"
#PING: 01," 32.1.7.0.0.0.1.46.0.0.0.0.0.15.112",47,237
...
OK
```

#### 16.4.5. #DNS Command & IPV6 Notation

Set primary and secondary DNS server. Use the hexadecimal notation for the IPV6 addresses.

```
AT#DNS=1,"2001:4860:4860::8888","2001:4860:4860::8844"
OK
```

Check the DNS addresses set manually. The IPV6 addresses are returned in dotted-decimal notation.

```
AT#DNS?
#DNS:
1,"32.1.72.96.72.96.0.0.0.0.0.0.0.136.136","32.1.72.96.72.96.0.0.0.0.0.0.0.136.68"
OK
```

#### 16.4.6. #FRWLIPV6 Command & IPV6 Notation

Add an ACCEPT chain. Use the hexadecimal notation for the IPV6 addresses.

```
AT#FRWLIPV6=1,"2001:700:0:12e::f70","ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff"
OK
```

Check all the ACCEPT chains. The command returns the IPV6 addresses in dotted-decimal notation.

```
AT#FRWLIPV6?
#FRWLIPV6:"32.1.7.0.0.0.1.46.0.0.0.0.0.15.112","255.255.255.255.255.255.255.255.255.255.255.255.255.255"
OK
```

## 16.5. #NCM Command

#NCM command sets up the Network Control Model protocol. The command provides two modes to establish a Default EPS Bearer:

- mode=1 (see chapter 16.5.1)
- mode=2 (see chapter 16.5.2)

#NCM command does not belong to the IP Easy environment and is available on modules having Platform Version ID 20, see documents [4], and [9].

### 16.5.1. #NCM Command, Mode = 1

Assume to use the #NCM command configured in NCM mode =1 (manual PDP context activation). In addition, #NCM command configuration does not use authentication parameters. To set the authentication parameters use the #PDPAUTH command.

At power on, assume that the PDP context identified by <cid>=1 has not defined APN.

**AT+CGDCONT?**

+CGDCONT: 1,"IPV4V6","", "",0,0

OK

After attach procedure, the PDP context identified by <cid>=1 is automatically associated to the Default EPS Bearer assigned by the Network Operator in response of the not defined APN.

Read the PDP context dynamic parameters.

**AT+CGCONTRDP=1**

+CGCONTRDP:

1,

5,

"APN\_Assigned\_by\_Network",

"100.87.125.10.255.0.0.0",

"100.87.125.11",

"80.201.237.239",

"0.0.0.0","0.0.0.0",

"0.0.0.0"

OK

Set up the new PDP Context identified by <cid>=3.

**AT+CGDCONT=3,"IPV4V6","Access\_Point\_Name"**

OK

Set the PDP authentication parameters on PDP context <cid>=3.

**AT#PDPAUTH=3,1,"UserName","PassWord"**

OK

Activate the PDP Context identified by <cid>=3.

**AT+CGACT=1,3**

OK

Check the PDP contexts dynamic parameters assigned by the Network Operator.

**AT+CGCONTRDP=**

```
+CGCONTRDP:
1,                ←PDP context activated during attach procedure
5,
"APN_Assigned_by_Network",
"100.114.19.108.255.0.0.0",
"100.114.19.109",
"80.201.237.239",
"0.0.0.0",
"0.0.0.0",
"0.0.0.0"
```

```
+CGCONTRDP:
3,                ← PDP context activated by +CGACT command
6,
"Access_Point_Name.....",
"10.113.0.66.255.0.0.0",
"10.113.0.67",
"8.8.4.4",
"0.0.0.0",
"0.0.0.0",
"0.0.0.0"
OK
```

You must configure your PC to use the NCM command, refer to document [9].

Use NMC mode = 1, and assign the NCM protocol to PDP context identified by <cid> = 3.

**AT#NCM=1,3**

OK

Use the AT+CGDATA=? test command to check the protocols supported by the module.

**AT+CGDATA=?**

+CGDATA: ("PPP","M-HEX","M-RAW\_IP"),(1-15)

OK

Activate the NCM protocol, refer to document [9].

**AT+CGDATA="M-RAW\_IP",3**

CONNECT

OK

## 16.5.2. #NCM Command, Mode = 2

This example focuses on the use of:

- I. #NCM with authentication parameters. If you use the AT#PDPAUTH command, and then use the AT#NCM command with new authentication parameters, the authentication process is performed using the last entered parameters.
- II. #NCM command in NCM mode = 2 (automatic PDP context and NCM activation).

At power on, assume that the PDP context identified by <cid>=1 has not defined APN.

**AT+CGDCONT?**

+CGDCONT: 1,"IPV4V6","", "",0,0

OK

After attach procedure, the PDP context identified by <cid>=1 is automatically associated to the Default EPS Bearer assigned by the Network Operator in response of the not defined APN (see +CGDCONT? command above).

Read the PDP context dynamic parameters.

**AT+CGCONTRDP=1**

+CGCONTRDP:

1,

5,

"APN\_Assigned\_by\_Network",

"100.80.28.204.255.0.0.0","100.80.28.205",

"80.201.237.239",

"0.0.0.0","0.0.0.0",

"0.0.0.0"

OK

Set up the new PDP Context associated to <cid>=1.

**AT+CGDCONT=1,"IPV4V6","Access\_Point\_Name"**

OK

Verify the entered PDP Context.

**AT+CGDCONT?**

+CGDCONT: 1,"IPV4V6","Access\_Point\_Name","",0,0

OK

Set the PDP authentication parameters on PDP context <cid>=1. This authentication parameters will be overwritten by the #NCM command used at the end of this example.

**AT#PDPAUTH=1,1,"UserName","PassWord"**

OK

Detach the module from the network.

**AT+CGATT=0**

OK

Verify if the module is detached.

**AT+CGATT?**

+CGATT: 0

OK

You must configure your PC to use the NCM command, refer to document [9].

Using the NCM mode = 2, the command activates automatically the PDP context identified by the <cid> (in this case <cid>=1). The authentication parameters override the previous User Name and Password entered through the #PDPAUTH command.

**AT#NCM=2,1,0,"UserName","PassWord"**

OK

Verify the state of the PDP Context.

**AT+CGACT?**

+CGACT: 1,1

OK

Check the PDP context dynamic parameters assigned by the Network Operator in response of the user defined PDP context activated by the #NCM command.

**AT+CGCONTRDP=1**

+CGCONTRDP:

1,

5,

"Access\_Point\_Name.....",

"10.113.0.66.255.0.0.0",

"10.113.0.67",

"8.8.4.4",

"0.0.0.0",

"0.0.0.0",

"0.0.0.0"

OK

## 16.6. #MBIMCFG Command

#MBIMCFG command selects the <cid> that will be used by MBIM standard when a connection will be established. The host (PC) is connected to the device (module) through a USB port. The device uses the assigned <cid> and starts the connection. The command syntax is:

**AT#MBIMCFG=<cid>**

#MBIMCFG command does not belong to the IP Easy environment and is available on modules having Platform Version ID 20, see document [4].

### 16.6.1. Modules: LE910 Series

Modules: LE910-EU1, B1-EU, -JN1, -NA1, B1-NA, B1-SA, -NA V2, B4-NA, -EU V2, AU V2

Check the available <cid> for the MBIM standard.

**AT#MBIMCFG=?**  
#MBIMCFG: (1-15)  
OK

Check the current <cid> assigned to the MBIM standard

**AT#MBIMCFG?**  
#MBIMCFG: 15  
OK

### 16.6.2. Modules: LE910-SV1, -SVL, SV V2

Check the module identification.

**AT+CGMM**  
LE910-SV1  
OK

In this product, only <cid> = 3 can be used by MBIM standard.

**AT#MBIMCFG=?**  
#MBIMCFG: (3)  
OK

Check the current <cid> assigned to the MBIM standard

**AT#MBIMCFG?**  
#MBIMCFG: 3  
OK

## 16.7. #NWMODE Command

#NWMODE command sets the network configuration on USB port, on which is running the RNDIS/ECM protocols. After entering this command, the module reboots. The command syntax is:

**AT#NWMODE=<mode>,<dhcp>,<dmz>**

#NWMODE command does not belong to the IP Easy environment, and is available on modules having Platform Version ID 23, see document [11].

Here is how to use the PPP protocol, and configure the module in router or bridge mode.

- In router mode, the module and the host establish a local network. If is used
    - IPv4 protocol, the host has a unique local address.
    - IPv6 protocol, the host may have also a global address.
  - In bridge mode, the host has the addresses assigned directly by the network operator.
- The PPP protocol is available only in router mode.

**#NWMODE=0,0,0** default configuration, module set for no inbound data to the host;

<mode>=0: router mode;

<dhcp> =0: DHCP on USB-RNDIS/ECM is disabled.  
Host IPv4 address is set manually at 10.0.0.10. To the host is also assigned a link-local IPv6 address.  
**Note:** the 10.0.1.10 address (see chapter 15.1.4.2) is active when PPP is running. Therefore, the host may have both addresses 10.0.1.10 and 10.0.0.10 if PPP is running, and USB active.

<dmz> =0: module server services are accessible from outside. The host cannot be accessed from outside (no inbound data to the host).  
The start of the PPP protocol triggers the PDP context activation, see chapter 15.1.1.  
If PPP protocol is not running, the host can use the USB-RNDIS/ECM connection to reach the external network. The PDP context must be active.

**#NWMODE=0,0,2** module set for inbound data to the host;

<mode>=0: router mode;

<dhcp> =0: DHCP on USB-RNDIS/ECM is disabled.  
Host IPv4 address is set manually at 10.0.0.10. To the host is also assigned a link-local IPv6 address.  
**Note:** the 10.0.1.10 address (see chapter 15.1.4.2) is active when PPP is running. Therefore, the host may have both addresses 10.0.1.10 and 10.0.0.10 if PPP is running, and USB active.

<dmz> =2: host server services are accessible from outside through PPP protocol (inbound data to the host).  
The start of the PPP protocol triggers the PDP context activation, see chapter 15.1.1.  
If PPP protocol is not running, the host can use the USB-RNDIS/ECM connection to reach the external network. The PDP context must be active.

**#NWMODE=0,1,0** network operator assigns to the host the global address IPv6;

<mode>=0: router mode;

<dhcp> =1: DHCP on USB-RNDIS/ECM is enabled as in a local network.  
Host IPv4 address is automatically set at 10.0.0.10.  
To the host are also assigned the global and link-local IPv6 addresses.  
Global address prefix is allocated by the network operator.

<dmz> =0: module server services are accessible from outside. The host cannot be accessed from outside (no inbound data to the host),  
The host can use the USB-RNDIS/ECM connection to reach the external network. The PDP context must be active.

➤ The PPP protocol is not available in bridge mode.

**#NWMODE=1,1,1** network operator assigns to the host the IPv4, and the global address IPv6. The module works like a modem;

<mode>=1: bridge mode;

<dhcp> =1: DHCP on USB-RNDIS/ECM is enabled, in this configuration the host sees the module like a modem, it means that #SD, #HTTP, #FTP, and #SSL AT commands are not available.  
Host IP address is automatically set using the IP address provided by the network operator. To the host are also assigned the global and link-local IPv6 addresses. Global address prefix is assigned by the network operator.

<dmz> =1: server services in the host are accessible from outside via USB-RNDIS/ECM connection (inbound data to the host). <dhcp> must be set to 1. PDP context must be active.

## 16.8. Obsolete Commands

Historically, IP Easy Extension is an improvement of the first IP Easy environment.

To underline the differences between IP Easy and IP Easy Extension environments, here is the correspondence table between the obsolete commands of the IP Easy environment and the new commands of the IP Easy Extension environment. It is strongly recommended to use only the commands set of the IP Easy Extension environment, and do not mix obsolete and new commands. In chapter 3, IP Easy term indicates the IP Easy Extension.

IP Easy	IP Easy Extension	
Obsolete commands	New commands	Operation description
AT#SKTOP	AT#SGACT; AT#SD	socket open
AT#SKTD	AT#SD	socket dial
AT#SKTL	AT#SL	socket listen
AT#SKTSET	not required	
AT#SKTSAV	not required	
AT#GPRS	AT#SGACT	activation of context
+++ after AT#SKTD	+++; AT#SH	socket close
+++ after AT#SKTOP	+++; AT#SH; AT#SGACT	
AT#USERID	AT#SGACT	authentication
AT#PASSWD	AT#SGACT	
AT#PKTSZ	AT#SCFG	socket configuration
AT#DSTO	AT#SCFG	
AT#SKTTO	AT#SCFG	
AT#SKTCT	AT#SCFG	

Tab. 25: Obsolete and New AT Commands

### Obsolete commands in Multi-socket environment

#SKTD and #SKTL obsolete commands (IP Easy) are available also on modules providing the Multi-socket feature (IP Easy Extension). They work in the old-fashioned: when a connection is opened with #SKTD, it cannot be suspended and the escape sequence (+++) closes definitively the connection.

#SKTOP must be not use with the new Multi-socket commands because it deactivates the context when the connection is closed, this could close the suspended sockets.

### CMUX and obsolete commands

Using CMUX tool, the module provides three virtual serial lines connected to three AT parsers, refer to document [6]. Assume that one HTTP (or SMTP, FTP) session is already opened on a CMUX virtual port. If you try to open a second HTTP client connection on another CMUX virtual port, the HTTP command returns an error message. When the old HTTP connection is closed, the new one can be opened.

CMUX protocol allows the use of three virtual port to enter AT commands, refer to document [6]. #SKTD allows opening three simultaneous connections. The connections are closed by means of the escape sequence (+++).

#SKTOP is a mono-socket command and works only with <cid>=1 and <connId>=1, therefore it can be used only on one virtual port of CMUX.

## 17. GLOSSARY AND ACRONYMS

	Description
APN	Access Point Name
AT	Attention commands
CHAP	Challenge-Handshake Authentication Protocol
CSD	Circuit Switched Data
DHCP	Dynamic Host Configuration Protocol
ECM	Ethernet Control Model
FTP	File Transfer Protocol
GBR	Guaranteed Bit Rate
GPRS	General Radio Packet Service
GSM	Global System for Mobile communication
HTML	Hyper Text Mark-up Language
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISP	Internet Service Provider
MBIM	Mobile Broadband Interface Model
NCM	Network Control Model
NTP	Network Time Protocol
PAP	Password Authentication Protocol
PDN	Packet Data Network
PDP	Packet Data Protocol
PDU	Protocol Data Unit

PPP	Point to Point Protocol
PSD	Packet Switching Data
QoS	Quality of Service
RNDIS	Remote Network Driver Interface Specification
SIM	Subscriber Identity Module
SMTP	Simple Mail Transfer Protocol
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security

## 18. DOCUMENT HISTORY

Revision	Date	Changes
0	2007-02-01	Initial release
1	2007-03-14	2.3.2 Easy GPRS – Email sending application: added new examples
2	2007-09-03	updated applicability table new disclaimer
3	2007-11-29	This document has been integrated with Multi-socket User Guide and is valid from the 7.02.03 SW release
4	2008-07-16	Added new features related to sw release 7.03.00 or 7.02.05 such as: command mode connections and Easy GPRS over GSM
5	2009-07-20	Added new features related to sw release 7.03.01 or 7.02.06 such as: FTP in command mode (par. 3.3.4.2, 3.3.5, 3.3.6)
6	2009-07-31	Applied new layout – shifted paragraphs accordingly Added explanation about escape sequence guard time
7	2009-08-24	Added new features related to sw release 7.03.01 or 7.02.06 such as: - automatic context activation, - direct control on TCP/IP settings, - Listen auto-response, UDP Listen, - command mode data sending in Hex format, - ICMP/PING handling.
8	2010-07-26	Added new features related to sw release 7.03.02 or 7.02.07 such as: - added new socket configuration parameters (see new #SCFGEXT2 command) - added new command #SSENDEXT for sending data in command mode Added new features related to sw release 7.03.02 ,7.02.07 or 10.0.02 such as: - AT#PADFWD and AT#PADCMD commands Removed parameter no more present in #EMAILD command from the example (page 61) Added new features related to SW release 10.0.03 such as: - AT#BASE64 command - AT#SGACTCFGEXT command  Added note in Easy GSM chapter, about COPS and CSURV commands
9	2010-10-04	Added GL865-DUAL to applicability table
10	2011-03-09	Added new features related to sw release 10.0.04 such as:

Revision	Date	Changes
		<p>- AT#SMTPCL command, it allows to send an email with different types of attachments. #SMTPCL handles attachment, managing MIME headers and encoding if required.</p> <p>Updated all references from "GPRS" to "GPRS/UMTS/HSPA"(or removed when not required)</p> <p>Correction on #SSENDEXT referenced wrongly as "#SSEND" at page 78; par. 5.2.3</p> <p>Correction on par. 5.2.1(pag.74), reference to #SD / cmd mode (was wrongly indicated as "Atem")</p> <p>Added general note regarding availability of commands on different SW version</p> <p>Changed title from EASY GPRS User Guide to IP Easy User Guide</p>
11	2011-10-17	Updated software versions Corrected mistake in note at par 2.1.3
12	2011-10-20	Update related to differences relating to SW versions starting from 12.00.xx1
13	2012-10-09	Updated Applicability and SW Versions tables. Modified values range of the <srMode> parameter of the AT#SCFGEXT command. Added new commands #FTPAPPEXT, #SLASTCLOSURE
14	2013-02-07	Removed a note in chapter 3.5.1. Updated Applicability Table.
15	2013-02-19	Updated Applicability Table and chapter 1.4 Added IPV6 parameter value relating +CGDCONT command in chapter 3.1 Modified max number of bytes from 1024 to 1500 in chapter 7.3. Added chapter 16.1.
16	2014-03-06	<p>Chapter 10.5.4: Changed sentence from "...three different procedures..." to "...two different procedure...". Changed sentence from "(in blue you can find ... , in violet the one received from host)" to "(in bold you can find the data sent by us, in regular the one received from host)".</p> <p><u>Products added:</u> GE910-QUAD AUTO / 13.00.xx5 GE910-GNSS / 13.00.xx4 GL865-QUAD V3 / 16.00.xx3 GE910-QUAD V3 / 16.00.xx3 GE866-QUAD / 16.00.xx3</p>

Revision	Date	Changes
		UE910-EUR / EUD / 12.00.xx4 UE910-NAR / NAD / 12.00.xx4 UL865-EUR/EUD / 12.00.xx4 UL865-NAR/NAD / 12.00.xx4 UL865-N3G / 12.00.xx4
17	2014-04-18	Added chapters 8 and 10 Chapter 3.7.1.- "IP Easy- HTTP client application" was renamed to "IP Easy HTTP client with #SD" Chapter 3.7.2. – "IP Easy - EMAIL sending application" was renamed to "IP Easy - EMAIL sending with #SD" and part with AT#SMTPCL / AT#EMAILD was removed Chapter 3.7.3 - "IP Easy -EMAIL receiving application" was renamed to "IP Easy -EMAIL receive with #SD" IPv6 table updated in chapter 6.- "Notes concerning IPV6 Implementation"
18	2014-07-14	Chapter 6.1.1: correct the syntax of the AT#SL command listed in the first table of the chapter.
19	2014-09-03	<u>Products added:</u> UE910-EU V2 / 19.10.xx2 UE910-NA V2 / 19.10.xx2 UE910-EU V2 AUTO / 19.10.xx2 HE910-EU V2 / 14.22.xx1 HE910-EUG V2 / 14.22.xx1 HE910-NA V2 / 14.22.xx1 HE910-NAG V2 / 14.22.xx1 HE920-EU AUTO / 14.12.xx1 HE920-NA AUTO / 14.12.xx1
20	2014-10-21	Chapter 3.2.3: changed the two footnotes
21	2014-11-17	Added the note "... open a socket with #SD ..." in chapter 3.2
22	2015-11-10	Updated the table in chapter 6. SW version 12.00.xxx supports IPV6 in IP Easy
23	2017-10-24	The document is fully revised, and chapters are reorganized. A new template is used.  Product series removed: GC864, GE864, GT86x, UC864, UE910 V2, and HE910 V2.  Product series added: LE910 Cat1, LE910 V2, LE866, and ME866A1.  Revised chapter 6.1 Firewall Setting.  Added HTTP POST and PUT examples.

Revision	Date	Changes
		Added: chapters 15.1.4.2 Dial-up Command, Platform ID 23; 16.7 #NWMODE.
24	2019-07-10	Product series added: LE910Cx



# SUPPORT INQUIRIES

Link to [www.telit.com](http://www.telit.com) and contact our technical support team for any questions related to technical issues.

[www.telit.com](http://www.telit.com)



Telit Communications S.p.A.  
Via Stazione di Prosecco, 5/B  
I-34010 Sgonico (Trieste), Italy

Telit Wireless Solutions Inc.  
3131 RDU Center Drive, Suite 135  
Morrisville, NC 27560, USA

Telit Wireless Solutions Ltd.  
10 Habarzel St.  
Tel Aviv 69710, Israel

Telit IoT Platforms LLC  
5300 Broken Sound Blvd, Suite 150  
Boca Raton, FL 33487, USA

Telit Wireless Solutions Co., Ltd.  
8th FL., Shinyoung Securities Bld.  
6, Gukjegeumyung-ro8-gil, Yeongdeungpo-gu  
Seoul, 150-884, Korea

Telit Wireless Solutions  
Tecnologia e Servicos Ltda  
Avenida Paulista, 1776, Room 10.C  
01310-921 São Paulo, Brazil

Telit reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. The information contained herein is provided "as is". No warranty of any kind, either express or implied, is made in relation to the accuracy, reliability, fitness for a particular purpose or content of this document. This document may be revised by Telit at any time. For most recent documents, please visit [www.telit.com](http://www.telit.com)

Copyright © 2016, Telit

Mod. 0809 2016-08 Rev.7