



LLD AT&Wn Command

Project	TCS 3.x
Document Type	Technical Documentation
Title	LLD AT&Wn Command
Author	Liyi Yu
Creation Date	08.12.2003
Last Modified	
ID and Version	??
Status	Being Processed

Copyright © 2002-2003 Texas Instruments, Inc. All rights reserved.

Texas Instruments Proprietary Information – Strictly Private

0 Document Control

© Copyright Texas Instruments, Inc. 2002-2003
All rights reserved.

Every effort has been made to ensure that the information contained in this document is accurate at the time of printing. However, the software described in this document is subject to continuous development and improvement. Texas Instruments reserves the right to change the specification of the software. Information in this document is subject to change without notice and does not represent a commitment on the part of Texas Instruments. Texas Instruments accepts no liability for any loss or damage arising from the use of any information contained in this document.

The software described in this document is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement. It is an offence to copy the software in any way except as specifically set out in the agreement. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Texas Instruments.

0.1 Document History

ID	Author	Date	Status
??	Liyi Yu	08.12.2003	Being Processed

0.2 References, Abbreviations, Terms

Table of Contents

0	Document Control	ii
0.1	Document History	ii
0.2	References, Abbreviations, Terms	ii
1	Introduction	2
2	Overview	3
2.1	General	3
2.2	Feature List.....	3
2.3	Basic Principle	4
3	New Functions and Data Structures.....	8
3.1	New Elements.....	8
3.1.1	cmdList []	8
3.1.2	profile[0] []	8
3.1.3	profile[1] []	8
3.2	New Functions.....	8
3.2.1	get_command_id ().....	8
3.2.2	store_command_list.....	9
3.2.3	read_command_list_from_ffs ().....	10
3.2.4	store_profile_to_ffs ()	10
3.2.5	read_profile_from_ffs ()	10
3.2.6	parameter_handler ()	11
3.2.7	sAT_AndW().....	11
3.2.8	sAT_Zn().....	12
4	Test Plan.....	13
4.1	MSC.....	13
4.1.1	Data flow of functions.....	13
4.1.2	Test Cases.....	14
4.2	Test Result Overview	28
4.2.1	Target Test.....	28
4.2.2	Windows Simulation Test.....	31

1 Introduction

G23 is a software package implementing Layers 2 and 3 of the ETSI-defined GSM air interface signalling protocol, and as such represents that part of a GSM mobile station's protocol software which is both, platform and manufacturer independent. Therefore, G23 can be viewed as a building block providing standardised functionality through generic interfaces for easy integration.

The G23 suite of products consists of the following items:

- Layers 2 and 3 for speech & short message services,
- Layers 2 and 3 for fax & data services,
- Application Control Interface,
- Slim MMI [02.30] and
- Test and integration support tools.

2 Overview

2.1 General

This document aims at describing the use [scenario](#) and the implementation of AT command AT&W and ATZ.

AT&W is not explicitly an ETSI/ITU standard command and thus not to be found in the usual AT commands related documentation (ETSI 07.07 / 07.05 and ITU V25ter / T32). However, it is a well-known modem AT command and the definition here will be mostly based on what is usually done by modem manufacturers.

The aim of this command is to provide the user with the possibility to store the current user profile to non-volatile storage. The implementation here supports 2 profiles.

In order to be more flexible, a mechanism of allowing the user to choose user-defined settings commands from a predefined list should be taken into account. E.g. adding a new AT command AT%S. But the implementation here only focuses on implementing the functionality of AT&W. So instead of implementing AT%S, the list of settings should be stored in non-volatile memory manually.

The list of the AT commands should be with the scope of the following commands:

+CGREG;+CLIP;+CLIR;+COLP;+CREG;% ALS;% CGAATT;% CGREG;% CPI;% CREG;

The format is also important (no space in between, using semi-colon at the end of each command). The command list should be stored in the following path in FFS: "/gsm/com/cmdList".

In order to test the functionality of AT&W, the supported values of ATZ shall be increased. This AT command is used for retrieving the user or default settings. So for the test purpose, the implementation of command ATZ is necessary and will be done together with AT&W.

2.2 Feature List

AT&W:

Set command stores the current settings (profile) according to the pre-defined command list which is stored in the non-volatile memory. If the storing of the profile fails, an unknown error message will be returned and the profile has not been stored.

Command	Possible response(s)
AT&W[<prfl>]	ERROR OK

<prfl>:

- 1 - Profile 1.
- 2 - Profile 2.

ATZ[<value>]:

This command instructs the DCE to set all parameters to defaults as specified by the manufacturer or by the user. If <prfl> is greater than 0 a user defined profile will be used.

Command	Possible response(s)
ATZ [<value>]	ERROR OK

<value>: integer type value indicating chosen profile:

- 0 - manufacturer specific default, same as ATZ
- 1 - Profile 1.
- 2 - Profile 2.

2.3 Basic Principle

The command Ids of the pre-chosen command list will be stored in ATI as a reference. The profile of this list of commands will be stored in the non-volatile memory. Three arrays will be created and stored in the non-volatile memory: An array to hold the user defined AT command strings and two other arrays to hold the profiles (parameters of the chosen AT commands). These three arrays are also stored in the ACI so that the window's simulation test can be performed. For the implementation here, the chosen list of settings should be stored in the FFS manually.

The three arrays created in non-volatile memory are as following:

cmdList [] for holding pre-chosen AT commands in strings;
profile[0] [] for holding parameters of profile0;
profile[1] [] for holding parameters for profile 1.

profile[0] [] and profile[1] [] are of data type UBYTE (unsigned char). cmdList [] is a char array.

The following diagrams show the format of the command list stored in the cmdList[] and the profile stored in profile[n][].

cmdList []:

+	C	L	I	R	;	%	A	L	S	;	%	C	G	A	A	T	T	;	...
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----

Profile[0][] and profile[1][]:

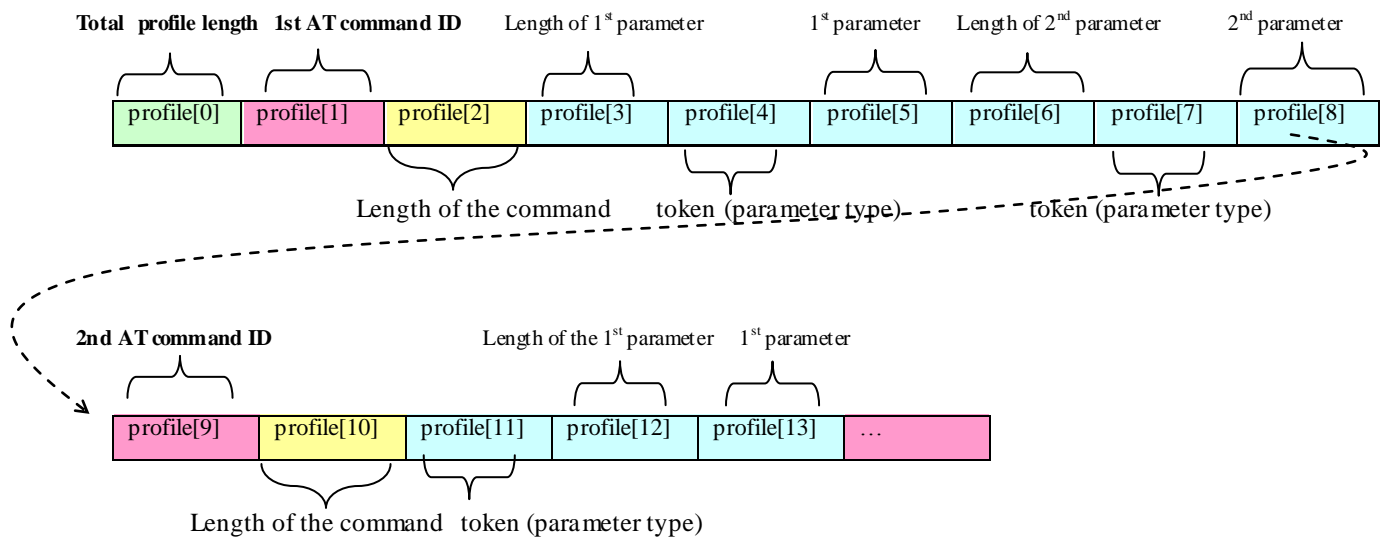
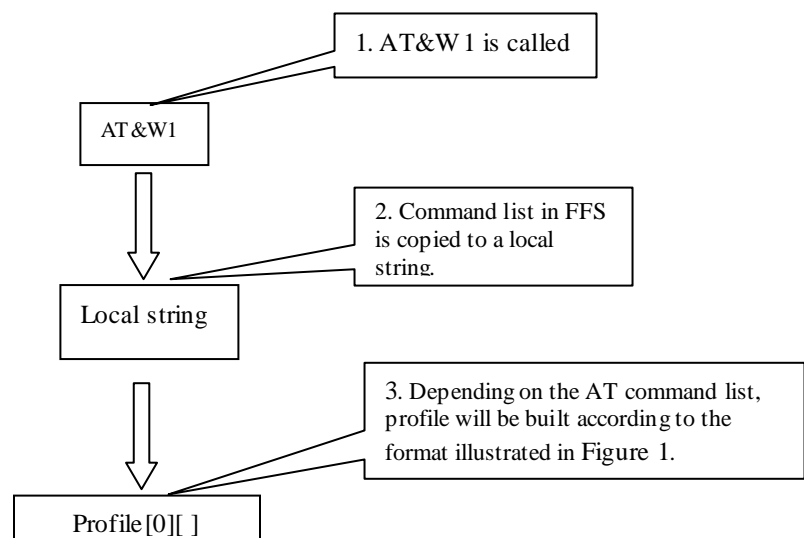


Figure 1 Illustration of data stored in FFS

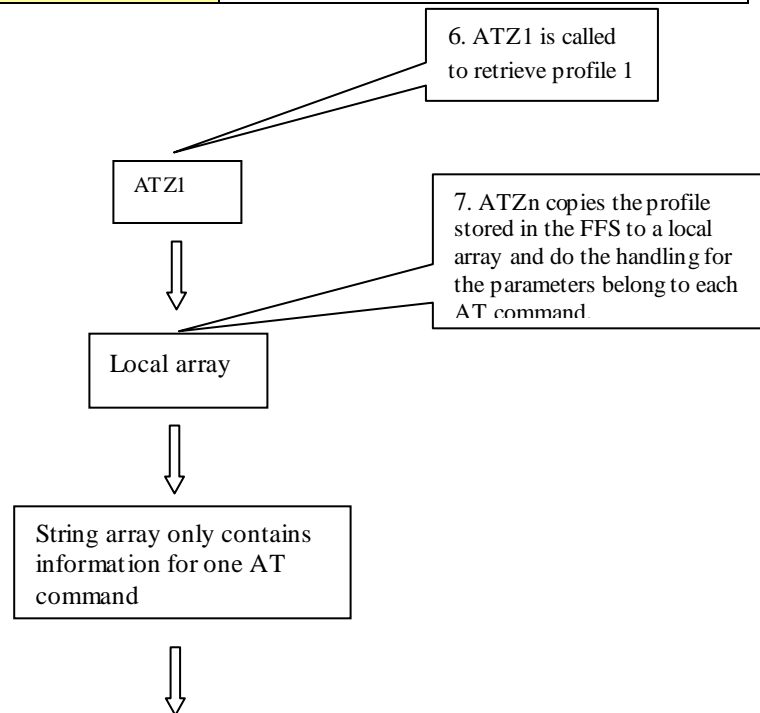
When ATZn is called, profile[n][] stored in FFS will be copied to a local array and handled locally. The local array will then be split up into smaller arrays only contain information for one AT command. Then the smaller arrays will be handled one by one according to the order of AT commands stored in FFS.

Following is a scenario of how AT&Wn is used in connection with ATZn (use command AT+CLIR as an example).

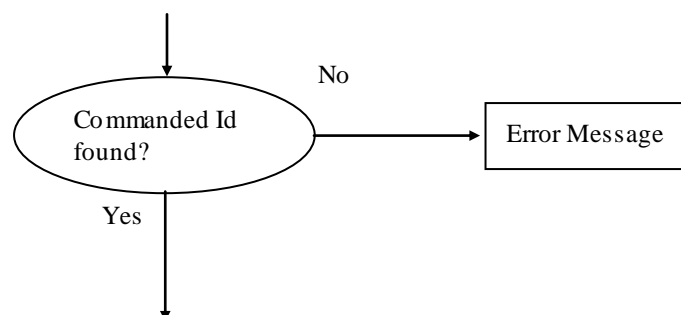


Element No.	Value of Parameter in FFS
0 (Total profile length)	n

1 (1 st command ID)	AT_CMD_CLIR
2 (length of this command)	5
3 (length of the following parameter)	1
4 (type of the following parameter)	SIGNED_VALUE
5 (first parameter)	CLIR_MOD_Enable
6 (2 nd command ID)	AT_CMD_ALS
...	...



1 (1 st command ID)	AT_CMD_CLIR
2 (profile length in bytes of this command)	5
3 (length of the following parameter)	1
4 (type of the following parameter)	SIGNED_VALUE
5 (first parameter)	CLIR_MOD_Enable



sAT_PlusCLIR ((T_ACI_CMD_SRC)srcId, para);

Figure 2 Illustration of an AT&W use scenario

3 New Functions and Data Structures

3.1 New Elements

3.1.1 cmdList []

Definition:

```
T_ACI_AT_CMD atw_cmd_list []
```

Use:

Hold the pre-chosen command strings. The corresponding array in FFS is in “/gsm/com/cmdList”.

```
T_ACI_AT_CMD atw_cmd_list []=  
    { AT_CMD_CGREG, AT_CMD_CLIP, AT_CMD_CLIR,  
      AT_CMD_COLP, AT_CMD_CREG, AT_CMD_ALS, AT_CMD_CGAATT,  
      AT_CMD_P_CGREG, AT_CMD_CPI, AT_CMD_P_CREG, AT_CMD_NONE};
```

3.1.2 atw_profile_simu [0] []

Definition:

```
UBYTE atw_profile_simu [0] [ ]
```

Use:

Holds profile 0. It is for simulation test only. The corresponding array is stored in path “/gsm/com/profile0” in FFS.

3.1.3 atw_profile_simu [1] []

Definition:

```
UBYTE atw_profile_simu [1] [ ]
```

Use:

Holds profile 1. It is for simulation test only. The corresponding array is stored in path “/gsm/com/profile1” in FFS.

3.2 New Functions

3.2.1 get_command_id ()

Prototype:

T_ACI_AT_CMD get_command_id(char *command_str)

Parameters:

command_str AT command string

Return:

T_ACI_AT_CMD

Description:

This function takes in the AT command string as input parameter and returns the corresponding command ID stored in a . If the input AT command is not in the pre-chosen list, AT_CMD_NONE will be returned.

3.2.2 get_command_id_in_wn_list()

Prototype:

LOCAL T_ACI_AT_CMD get_command_id_in_wn_list(CHAR *command_str)

Parameters:

command_str AT command string

Return:

T_ACI_AT_CMD

Description:

This function gets the command Id by calling function get_command_id() for the input command string.

3.2.3 store_command_list()

Prototype:

T_ATI_RSLT store_command_list(void)

Parameters:

Void

Return:

ATI_FAIL	Execution of command failed
ATI_CMPL	Execution of command completed

Description:

This function stores the pre-chosen AT commands for the windows simulation test. This is not needed for target because the list will be stored manually.

3.2.4 read_command_list_from_ffs ()

Prototype:

T_ATI_RSLT read_command_list_from_ffs (UBYTE *cmd_id_read)

Parameters:

cmd_id_read	Holds the command Ids of the AT command IDs
-------------	---

Return:

ATI_FAIL	Execution of command failed
ATI_CMPL	Execution of command completed

Description:

This function reads the AT command strings stored in FFS and passes the corresponding command Ids to the input pointer parameter < cmd_id_read >.

Possible commands to handle:

+CGREG;+CLIP;+CLIR;+COLP;+CREG;% ALS;% CGAATT;% CGREG;% CPI;% CREG;

An unknown command will be ignored.

3.2.5 store_profile_to_ffs ()

Prototype:

LOCAL T_ATI_RSLT store_profile_to_ffs (T_ACI_CMD_SRC srcId, SHORT nProfile)

Parameters:

SrcId	Source Id
nProfile	Which profile to be stored

Return:

ATI_FAIL	Execution of command failed
ATI_CMPL	Execution of command completed

Description:

This function builds and stores the profile in FFS according to the command list read from cmdList .

3.2.6 read_profile_from_ffs ()

Prototype:

```
LOCAL T_ATI_RSLT read_profile_from_ffs (SHORT nProfile, UBYTE *profile_read)
```

Parameters:

nProfile:	Input parameter indicating which profile should be read
parameter_read:	An array to hold the content of <profilen[]>

Return:

ATI_FAIL	Execution of command failed
ATI_CMPL	Execution of command completed

Description:

This function reads the profile stored in FFS and passes the content to the input parameter <profile_read>. Which profile will be read depends on the input parameter <nProfile>.

3.2.7 set_param_to_the_retrieved_value ()

Prototype:

```
LOCAL T_ATI_RSLT set_param_to_the_retrieved_value(UBYTE srcId, UBYTE *string)
```

Parameters:

srcId	Source Id
string	A string containing ATI parameter info to be handled.

Return:

ATI_FAIL	Execution of command failed
ATI_CMPL	Execution of command completed

Description:

This function is called by function ati_zn_retrieve_params () to handle ATI and CMH parameters.

3.2.8 ati_wn_store_params ()

Prototype:

```
LOCAL T_ATI_RSLT ati_wn_store_params (UBYTE srcId, SHORT nProfile)
```

Parameters:

srcId	Source ID;
nProfile	profile to be written.

Return:

ATI_FAIL	Execution of command failed
----------	-----------------------------

ATI_CMPL

Execution of command completed

Description:

Set command stores the current settings. If setting fails, an unknown error message is returned and the profile has not been stored.

3.2.9 ati_zn_retrieve_params ()

Prototype:

LOCAL T_ATI_RSLT ati_zn_retrieve_params (UBYTE srcId, SHORT nProfile)

Parameters:

srcId:	Source ID;
nProfile:	Profile to be written.

Return:

ATI_FAIL	Execution of command failed
ATI_CMPL	Execution of command completed

Description:

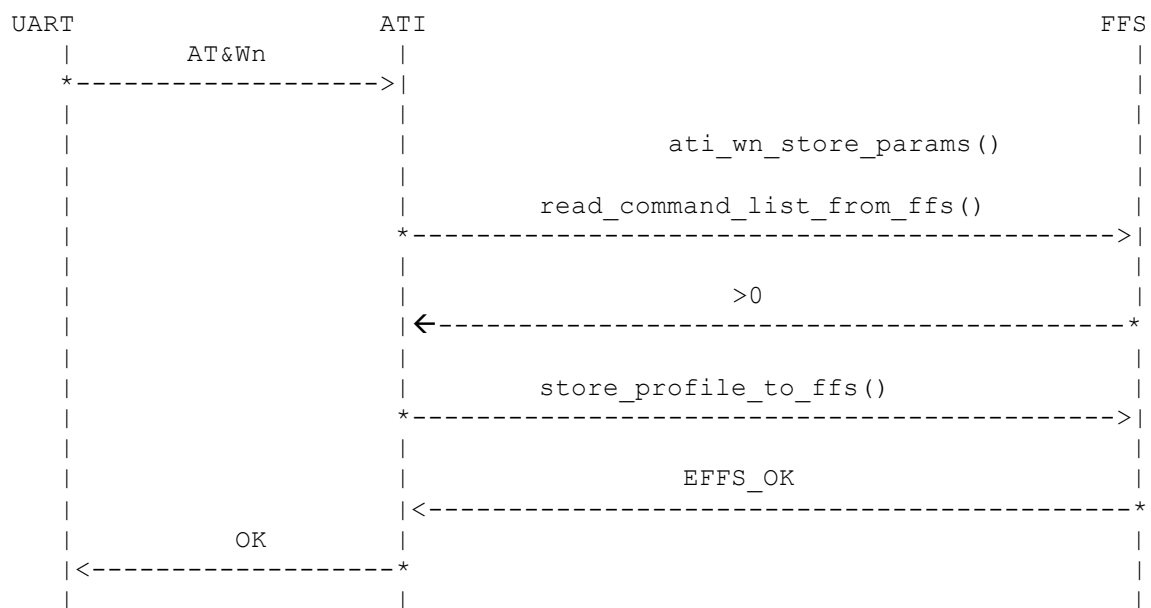
Set command retrieves the settings stored in FFS. It calls function set_param_to_the_retrieved_value () to handle the parameters. If execution fails, an unknown error is returned and the profile has not been retrieved.

4 Test Plan

4.1 MSC

4.1.1 Data flow of functions

(1) AT&Wn:



(2) ATZn:

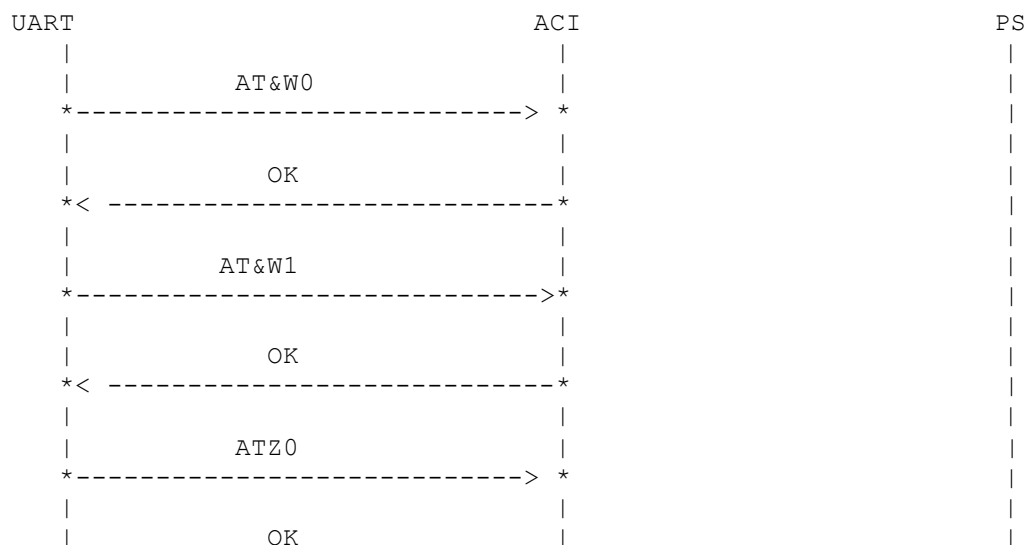


4.1.2 Test Cases

4.1.2.1 T_CASE ACI001

Purpose:

To test the implementation of AT&W and ATZ.



< -----		
ATZ1		
----->		
OK		
< -----		

4.1.2.2 T_CASE ACH002

Purpose:

Test the functionality of storing and retrieving the settings for AT command AT%CREG.

UART	ACI	PS
AT%CREG=0		
----->		
OK		
< -----		
AT&W0		
----->		
OK		
< -----		
AT%CREG=1		
----->		
OK		
< -----		
AT&W1		
----->		
OK		
< -----		
ATZ0		
----->		
OK		
< -----		
AT%CREG?		
----->		
%CREG: 0, 0, , , 2		
----->		
OK		

```

*< -----*
|                                     |
|               ATZ1                 |
|----->*
|                                     |
|               OK                   |
*< -----*
|                                     |
|               AT%CREG?             |
|----->*
|                                     |
|    %CREG: 1, 0, , , 2             |
|----->*
|                                     |
|               OK                   |
*< -----*
|                                     |
|               AT%CREG=2             |
|----->*
|                                     |
|               OK                   |
*< -----*
|                                     |
|               AT&W0                 |
|----->*
|                                     |
|               OK                   |
*< -----*
|                                     |
|               ATZ0                 |
|----->*
|                                     |
|               OK                   |
*< -----*
|                                     |
|               AT%CREG?             |
|----->*
|                                     |
|    %CREG: 2, 0, , , 2             |
|----->*
|                                     |
|               OK                   |
*< -----*
|                                     |

```

4.1.2.3 T_CASEACH1003

Purpose:

Test the functionality of storing and retrieving the settings for AT command AT+CREG.

UART
|

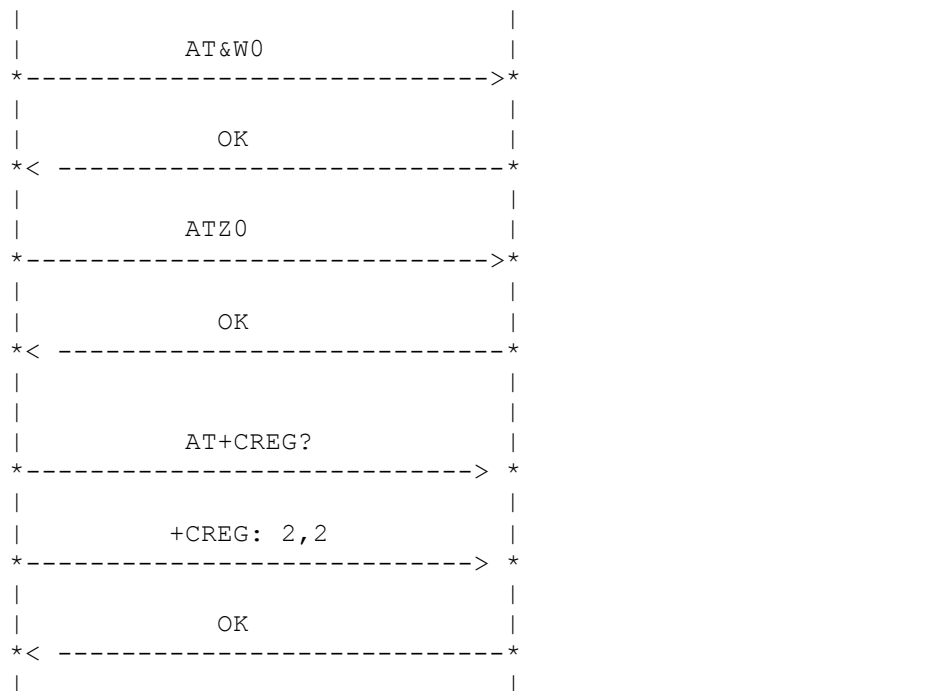
ACI
|

PS
|

```

|           AT+CREG=0           |
|----->|
|           OK                   |
|<-----|
|           AT&W0                 |
|----->|
|           OK                   |
|<-----|
|           AT+CREG=1             |
|----->|
|           OK                   |
|<-----|
|           AT&W1                 |
|----->|
|           OK                   |
|<-----|
|           ATZ0                 |
|----->|
|           OK                   |
|<-----|
|           AT+CREG?              |
|----->|
|           +CREG: 0,2            |
|----->|
|           OK                   |
|<-----|
|           ATZ1                 |
|----->|
|           OK                   |
|<-----|
|           AT+CREG?              |
|----->|
|           +CREG: 1,2            |
|----->|
|           OK                   |
|<-----|
|           AT+CREG=2             |
|----->|
|           OK                   |
|<-----|

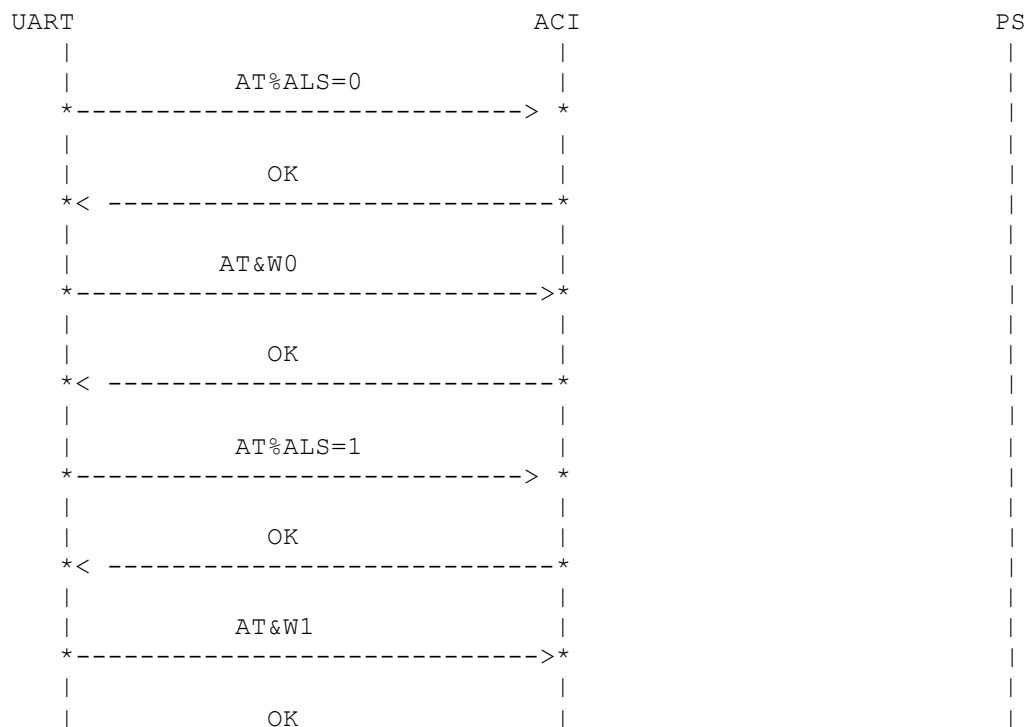
```

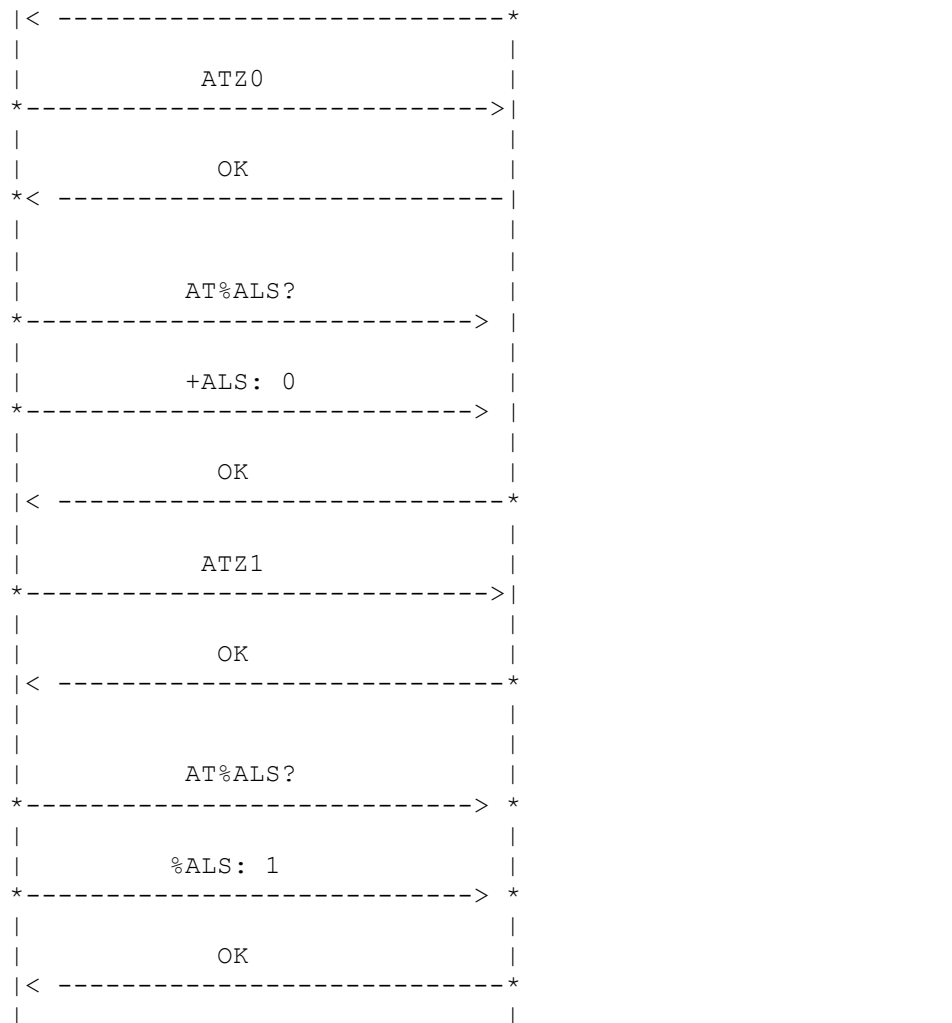


4.1.2.4 T_CASE ACI004

Purpose:

Test the functionality of storing and retrieving the settings for AT command AT%ALS.

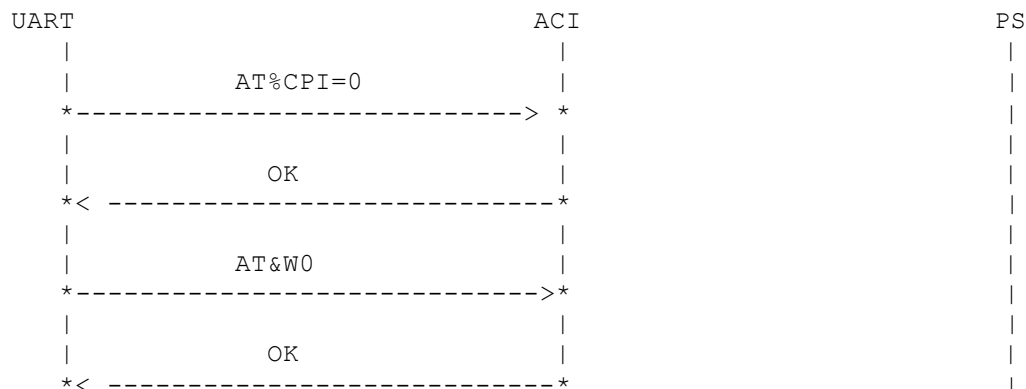




4.1.2.5 T_CASE ACH005

Purpose:

Test the functionality of storing and retrieving the settings for AT command AT% CPI.



```

|                                     |
|      AT%CPI=1                      |
|-----> *                           |
|                                     |
|      OK                            |
|<-----*                           |
|                                     |
|      AT&W1                         |
|-----> *                           |
|                                     |
|      OK                            |
|<-----*                           |
|                                     |
|      ATZ0                          |
|-----> *                           |
|                                     |
|      OK                            |
|<-----*                           |
|                                     |
|      AT%CPI?                      |
|-----> *                           |
|                                     |
|      %CPI: 0                      |
|-----> *                           |
|                                     |
|      OK                            |
|<-----*                           |
|                                     |
|      ATZ1                          |
|-----> *                           |
|                                     |
|      OK                            |
|<-----*                           |
|                                     |
|      AT%CPI?                      |
|-----> *                           |
|                                     |
|      %CPI: 1                      |
|-----> *                           |
|                                     |
|      OK                            |
|<-----*                           |
|                                     |
|      AT%CPI=2                      |
|-----> *                           |
|                                     |
|      OK                            |
|<-----*                           |
|                                     |
|      AT&W0                         |
|-----> *                           |
|                                     |
|      OK                            |
|<-----*                           |
|                                     |
|      ATZ0                          |
|-----> *                           |

```

```

*----->*
|
|           OK
|-----*
|
|           AT%CPI?
|----->*
|
|           %CPI: 2
|----->*
|
|           OK
|-----*
|

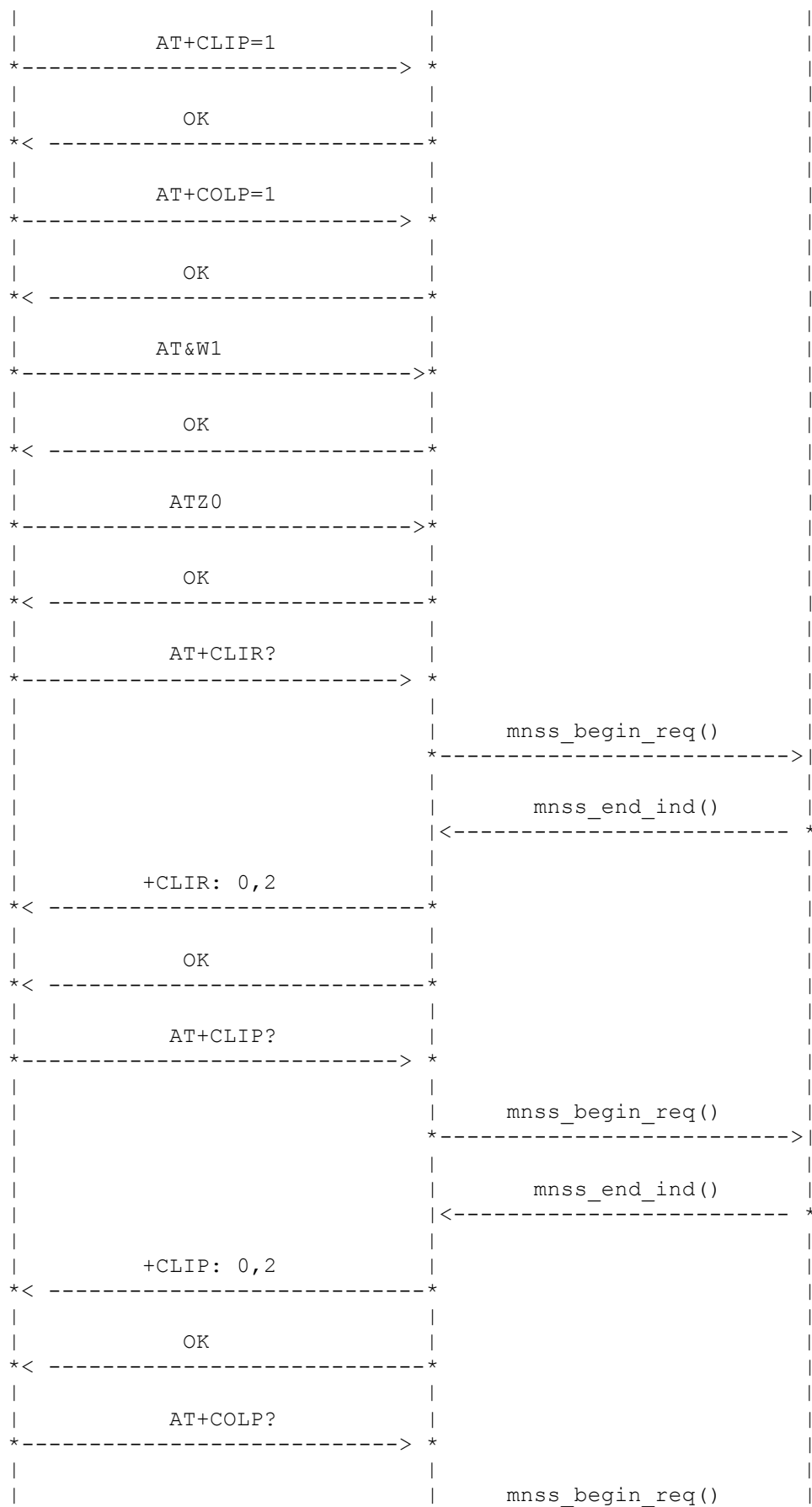
```

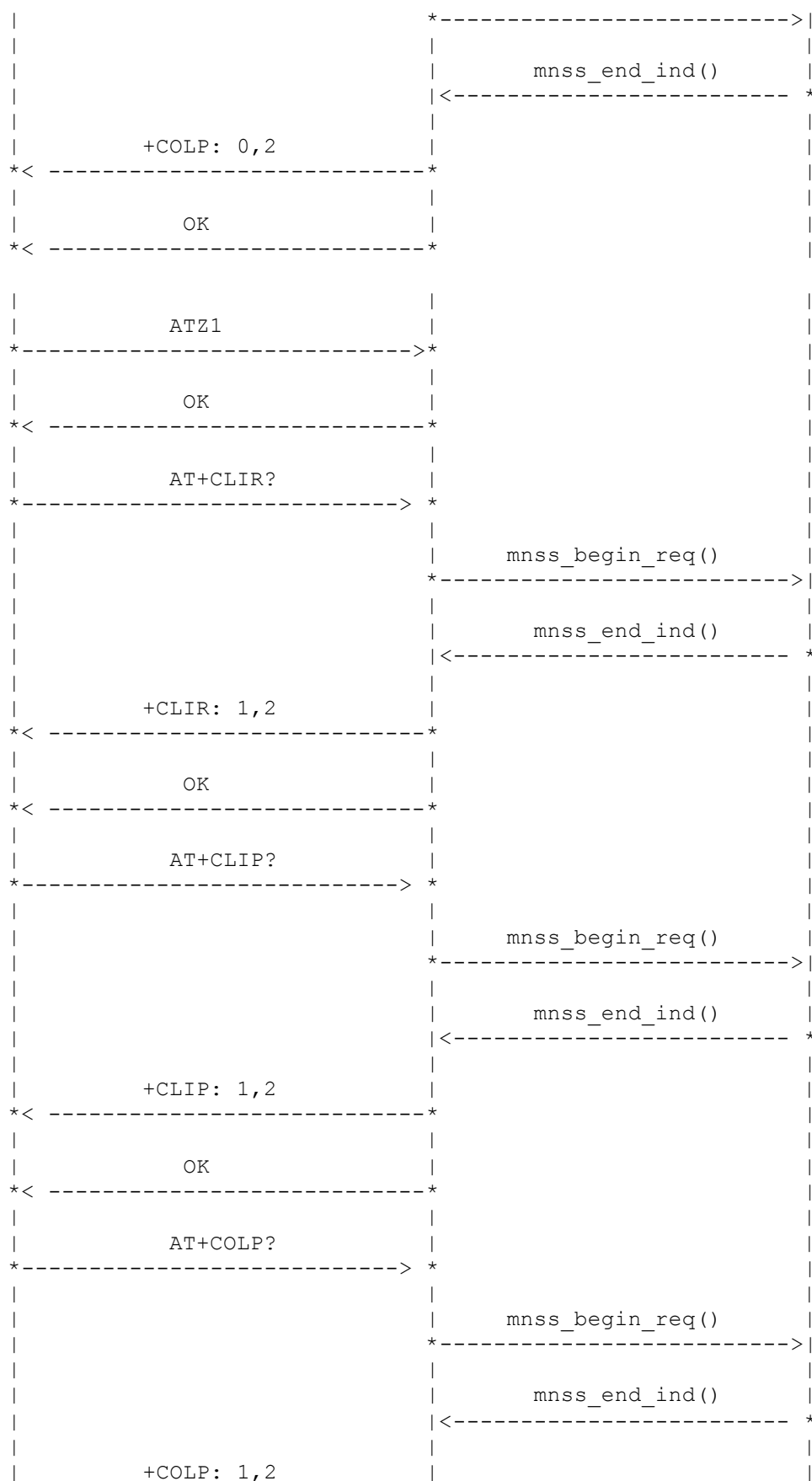
4.1.2.6 T_CASEACI007

Purpose:

Test the functionality of AT&W and ATZn by storing the parameters of +COLP, +CLIR, +CLIP into profiles and retrieving them later.

UART	ACI	PS
AT+CLIR=0		
----->		
OK		
----->		
AT+CLIP=0		
----->		
OK		
----->		
AT+COLP=0		
----->		
OK		
----->		
AT&W0		
----->		
OK		
----->		
AT+CLIR=1		
----->		
OK		
----->		





4.1.2.7 T_CASE GACI830

Purpose:

Save parameters for AT commands +CGAATT into profile.

```

sequenceDiagram
    participant UART
    participant ACI
    participant PS

    PS->>PS: PS is high
    UART->>ACI: AT%CGAATT=1,1
    ACI->>UART: OK
    UART->>ACI: AT&W0
    ACI->>UART: OK
  
```

```

| AT&W1 |
|*----->|
| | |
| OK |
|*<-----|
| | |
| AT%CGAATT=0,1 |
|*----->|
| | |
| OK |
|*<-----|
| | |
| ATZ0 |
|*----->|
| | |
| OK |
|*<-----|
| | |
| AT%CGAATT? |
|*----->|
| | |
| %CGAATT: 1,1 |
|*<-----|
| | |
| OK |
|*<-----|
| | |
| ATZ1 |
|*----->|
| | |
| OK |
|*<-----|
| | |
| AT%CGAATT? |
|*----->|
| | |
| %CGAATT: 1,1 |
|*<-----|
| | |
| OK |
|*<-----|

```

Purpose:

```

sequenceDiagram
    participant UART
    participant ACI
    participant PS

    Note over UART: AT+CGREG=1
    UART-->>ACI: *
    ACI-->>UART: OK
    Note over ACI: *
    
```

```

|           AT&W0           |
|----->|
|           OK           |
|<-----|
|           AT&W1           |
|----->|
|           OK           |
|<-----|
|           AT+CGREG=0       |
|----->|
|           OK           |
|<-----|
|           ATZ0           |
|----->|
|           OK           |
|<-----|
|           AT+CGREG?       |
|----->|
|           +CGREG: 1,0     |
|<-----|
|           OK           |
|<-----|
|           ATZ1           |
|----->|
|           OK           |
|<-----|
|           AT+CGREG?       |
|----->|
|           +CGREG: 1,0     |
|<-----|
|           OK           |
|<-----|

```

4.1.2.9 T_CASEGACI832

Purpose:

Save parameters for AT commands %CGREG into profile.

UART

ACI

PS

|

|

|

```

|           AT%CGREG=1           |
|----->|
|           OK                    |
|<-----|
|           AT&W0                 |
|----->|
|           OK                    |
|<-----|
|           AT&W1                 |
|----->|
|           OK                    |
|<-----|
|           AT%CGREG=0           |
|----->|
|           OK                    |
|<-----|
|           ATZ0                 |
|----->|
|           OK                    |
|<-----|
|           AT%CGREG?            |
|----->|
|           %CGREG: 1,0          |
|<-----|
|           OK                    |
|<-----|
|           ATZ1                 |
|----->|
|           OK                    |
|<-----|
|           AT%CGREG?            |
|----->|
|           %CGREG: 1,0          |
|<-----|
|           OK                    |
|<-----|

```

4.2 Test Result Overview

4.2.1 Target Test

Target build for ACI only, MFW/BMI, Goliteis done. ACIA built not possible.

Feature related target test on D-Sample is done with the following traces.

Trace:

at+clir=1

OK

at+cgreg=1

OK

at+clip=1

OK

at+colp=1

OK

at+creg=1

OK

at%als=1

ERROR

at%cgaatt=1

OK

at%cgreg=1

OK

at%cpil=1

OK

at%creg=1

OK

at&w1

OK

at+cgreg=0

OK

at+clip=0

OK

at+clir=0

OK

at+colp=0

OK

at+creg=0

OK

at%als=0

OK

at%cgaatt=0

OK

at%cgreg=0

OK

at%cpi=0

OK

at%creg=0

OK

at&w0

OK

atz1

OK

at+cgreg?

+CGREG: 1,0

OK

at+clip?

+CLIP: 1,2

OK

at+clir?

+CLIR: 1,2

OK

at+colp?

+COLP: 1,2

OK

at+creg?

+CREG: 1,2

OK

at%als?

%ALS: 0

OK

at%cgaatt?

%CGAATT: 1,1

OK

at%cgreg?

%CGREG: 1, 0

OK

at%cpil?

%CPI: 1

OK

at%creg?

%CREG: 1, 2, , , 0

OK

atz0

OK

at+cgreg?

+CGREG: 0,0

OK

at+clip?

+CLIP: 0,2

OK

at+clir?

+CLIR: 0,2

OK

at+colp?

+COLP: 0,2

OK

at+creg?

+CREG: 0,2

OK

at%als?

%ALS: 0

OK

at%cgaatt?

%CGAATT: 0,1

OK

at%cgreg?

%CGREG: 0, 0

OK

at%cpi?

%CPI: 0

OK

at%creg?

%CREG: 0, 2, , , 0

OK

4.2.2 Windows Simulation Test

Build successfully for the following GPRS stacks:

G23_SMI_GPRS.exe

G23_SMI_GPRS_WAP.exe

Build successfully for the following GSM stacks:

MS_TI_FD.exe

MS_TI_WAP.exe

Windows simulation test is done for:

ACI_test with MS_TI_FD.exe

GACI_test with G23_SMI_GPRS.exe

Test was done for GACI test, no new fails were introduced, but for ACI test even the baseline was not ok, so ACI test has not been done.