LLD - %CPRSM – Pause Receiving Short Messages

Project	G23-GSM Protocol Stack
Document Type	Low Level Design Description
Title	LLD - %CPRSM - Pause Receiving Short Messages
Author	Marc Droste-Franke
Creation Date	06.04.2004 16:43
Last Modified	3/8/2015 8:44:00 PM
ID and Version	8462.732.04.002
Status	Submitted

1 Document Control

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.

1.1 Document History

ID	Author	Date	Status
8462.732.04.001	Marc Droste-Franke	15.04.2004	Submitted
8462.732.04.002	Marc Droste-Franke	19.04.2004	Submitted

1.2 References, Abbreviations, Terms

[1] 7010.801, References and Vocabulary, Texas Instruments.

2 Table of Contents

3 Introduction

To support the customers product it has been requested to provide an AT command for pause and resume the receiving of short messages.

The AT command %CPRSM implements this feature using new primitives that have been added to the MNSMS SAP. The changes of the SAP and modifications in the SMS entity are not in the scope of this document.

3.1 Interface description

3.1.1 AT-command syntax

Command	Possible response(s)
%CPRSM=<set_mode>	+CMS ERROR: <err>
%CPRSM?	%CPRSM: <set_mode> +CMS ERROR: <err>
%CPRSM=?	%CPRSM: (0,1) +CMS ERROR: <err>

Description
The aim of this command is to provide the user the possibility to pause and resume the receiving of short messages. If the user sets the <set_mode> paramter to 1 and the command is preformed successfully the ME will block all incoming messages until the <set_mode> is set to 0. The SMSC will buffer the messages that have not been acknowledged. Receiving the resume the SMSC will immediately start to send the short messages stored in the buffer.
The query command informs the user about the current mode of the %CPRSM.

Defined values
<set_mode >: integer type value indicating:

pause/resume receiving of short messages

1 Pause
0 Resume

3.1.2 Data types

```
typedef enum
{
  CPRSM_MOD_NotPresent    = -1,
  CPRSM_MOD_Resume        =  0,
  CPRSM_MOD_Pause         =  1
} T_ACI_CPRSM_MOD;
```

The possible CPRSM mode parameters are specified by the enumeration T_ACI_CPRSM_MOD.

3.1.3 Functional Interface

3.1.3.1 sat_PercentCPRSM

Provides the functionality for the AT%CPRSM=<set_mode> command

Pototype:
T_ACI_RETURN **sAT_PercentCPRSM** (T_ACI_CMD_SRC srcId,
T_ACI_CPRSM_MOD mode)

Paramters:

srcId	source identifier
mode	mode to which the receiving of SMS should be set to

Return:

AT_EXCT	execution of the command is in progress
AT_FAIL	execution of the command failed
AT_BUSY	execution of the command is blocked due to a busy command handler
AT_CMPL	execution of the command successfully completed

3.1.3.2 qat_PercentCPRSM

Provides the functionality for the AT%CPRSM? command

Prototype:
T_ACI_RETURN **qAT_PercentCPRSM** (T_ACI_CMD_SRC srcId)

Parameters:

srcId	source identifier

Return:

AT_EXCT	execution of the command is in progress
AT_FAIL	execution of the command failed
AT_BUSY	execution of the command is blocked due to a busy command handler

3.1.3.3 rAT_PercentCPRSM

Callback for signaling the result of qAT_PercentCPRSM to the MMI. Implemented as dummy function.

Prototype:
void **rAT_PercentCPRSM** (T_ACI_CPRSM_MOD mode)

Parameters:

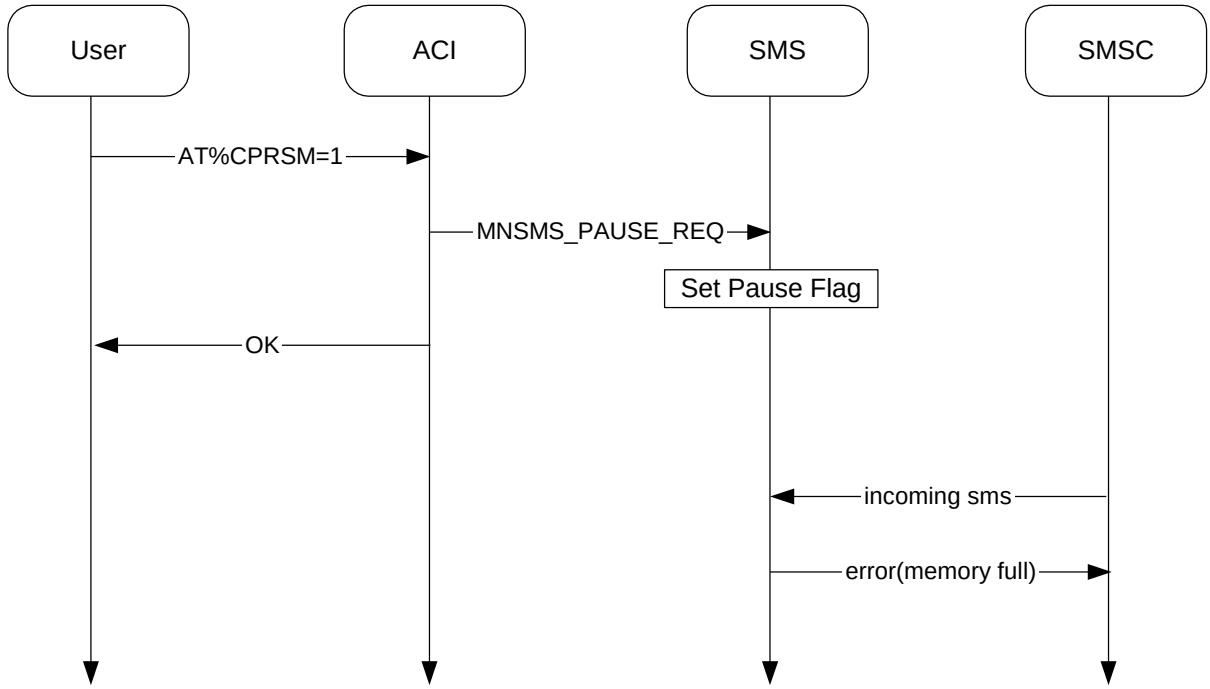
mode	mode to which the receiving of SMS is set

Return:
none

3.2 Interaction Schemes (MSCs)

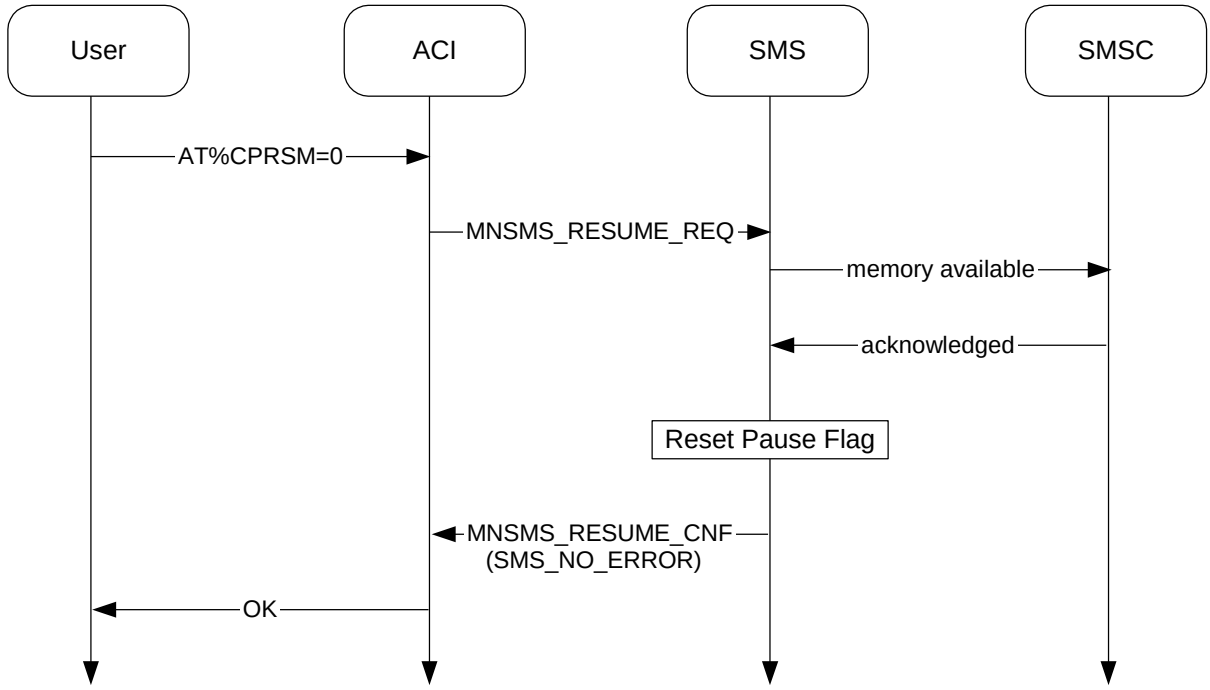
Only the behavior for the TCS 3.1 will be described in the following.

3.2.1 Using +CPRSM for setting Receiving of SMS to Pause, returning OK



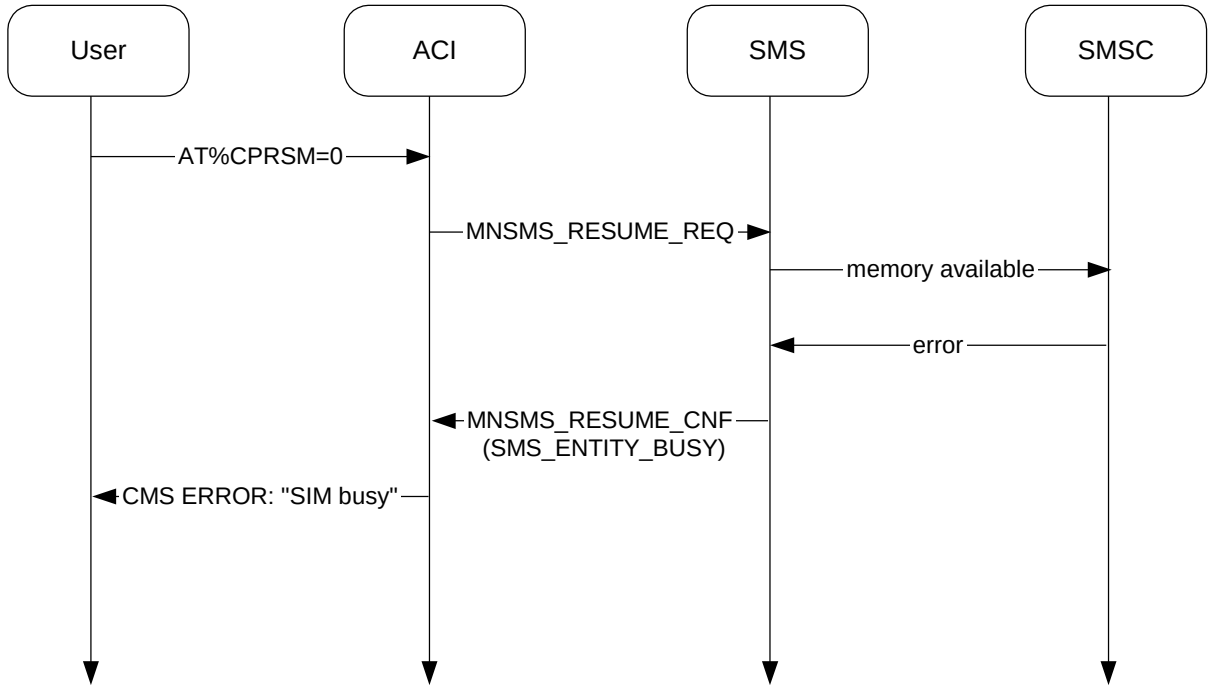
Scheme 1: After the Pause Flag has been set in the SMS entity, incoming SMS are rejected by responsing an error: memory full to the SMSC. The SMSC will buffer all SMS that have not been acknowledged by the ME. The SMS entity also sets the SMS full flag on the SIM card.

3.2.2 Using +CPRSM for setting Receiving of SMS to Resume, returning OK



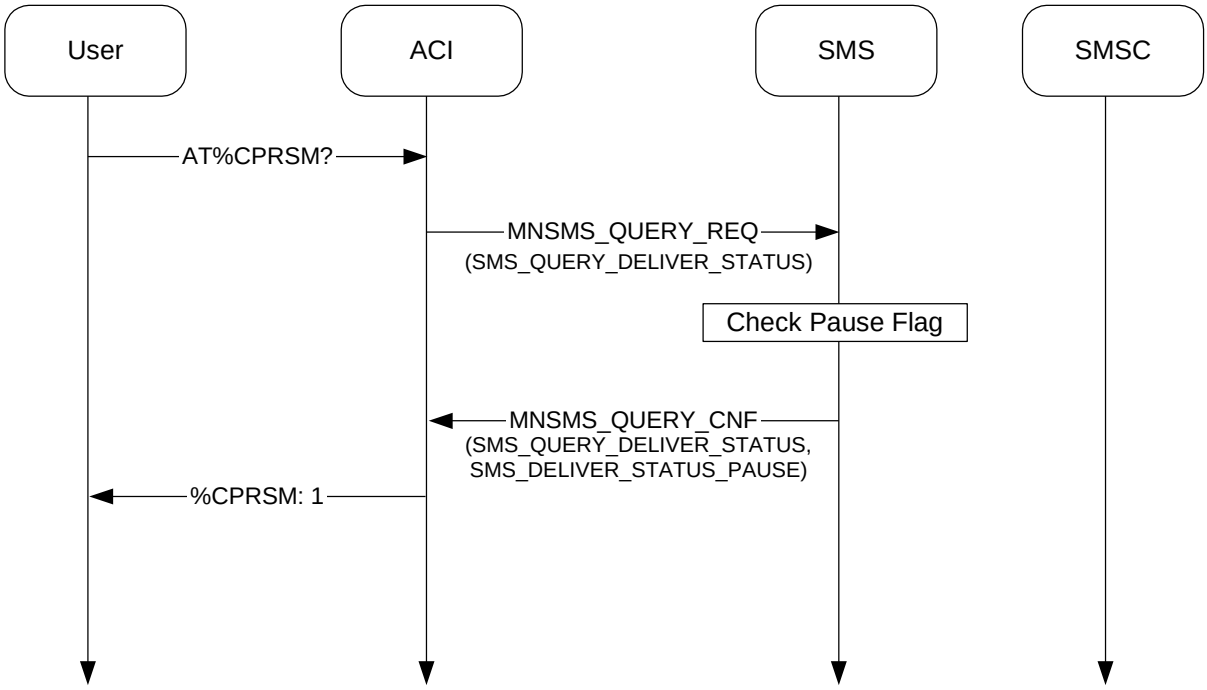
Scheme 2: The OK is given in the MNSMS_RESUME_CNF. After the SMS entity has indicated that the memory is now available, the SMSC starts to resend the buffered SMS.

3.2.3 Using +CPRSM for setting Receiving of SMS to Resume, returning error



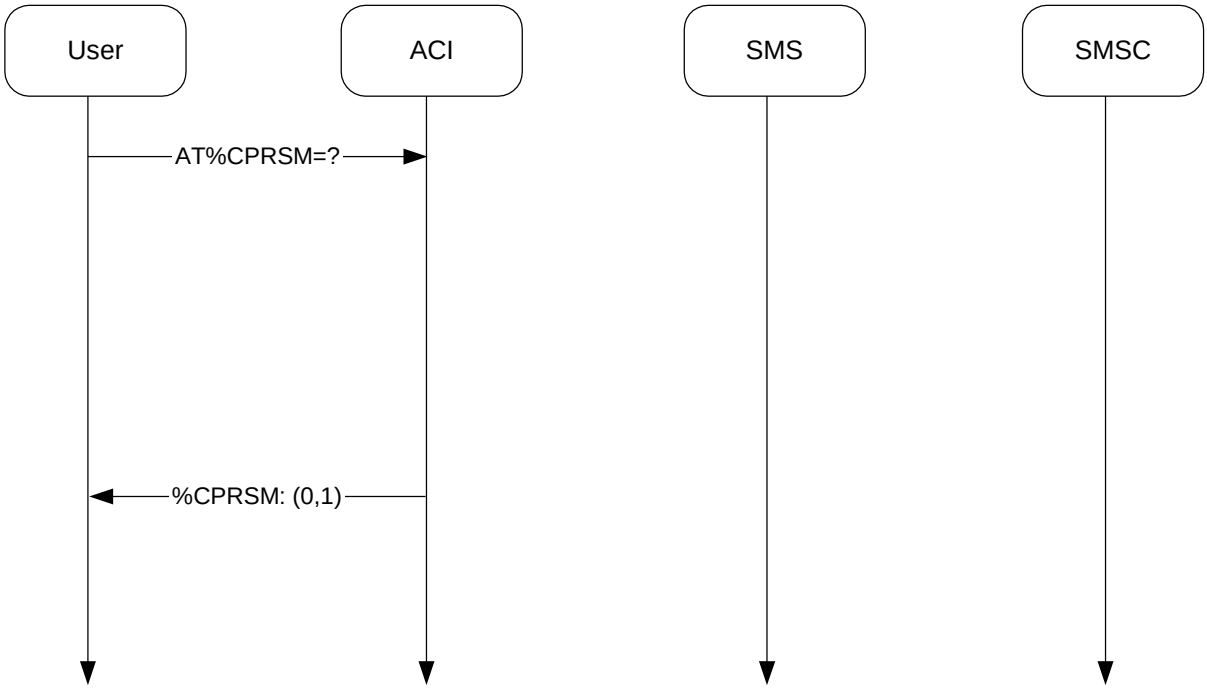
Scheme 3: The ERROR cause is given in the MNSMS_RESUME_CNF. The error that is indicated by the CMS ERROR: <err>.

3.2.4 Using +CPRSM? for checking the status of Receiving SMS



Scheme 4: The MNSMS_QUERY_REQ includes the requested information of the delivery status, respectively the status of receiving SMS.

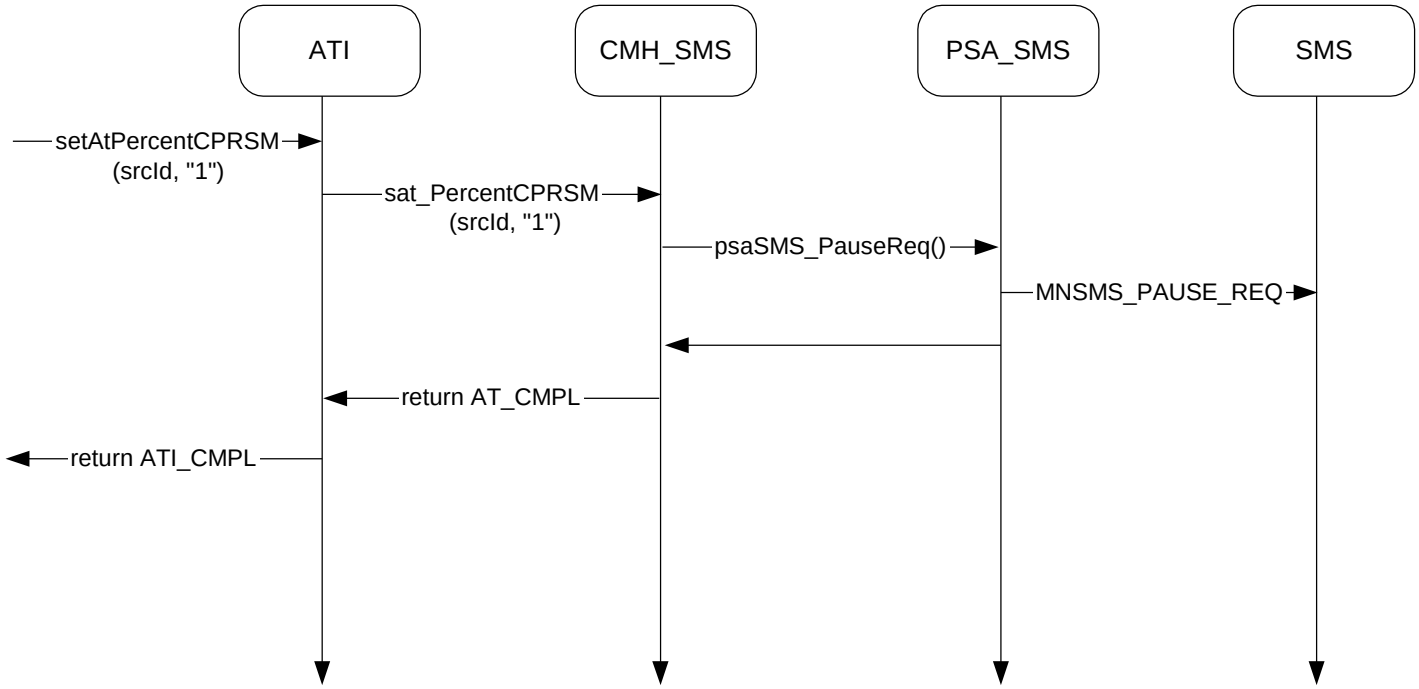
3.2.5 Using +CPRSM=? for testing the parameters of this command



Scheme 4: The allowed parameter set is generated within ACI.

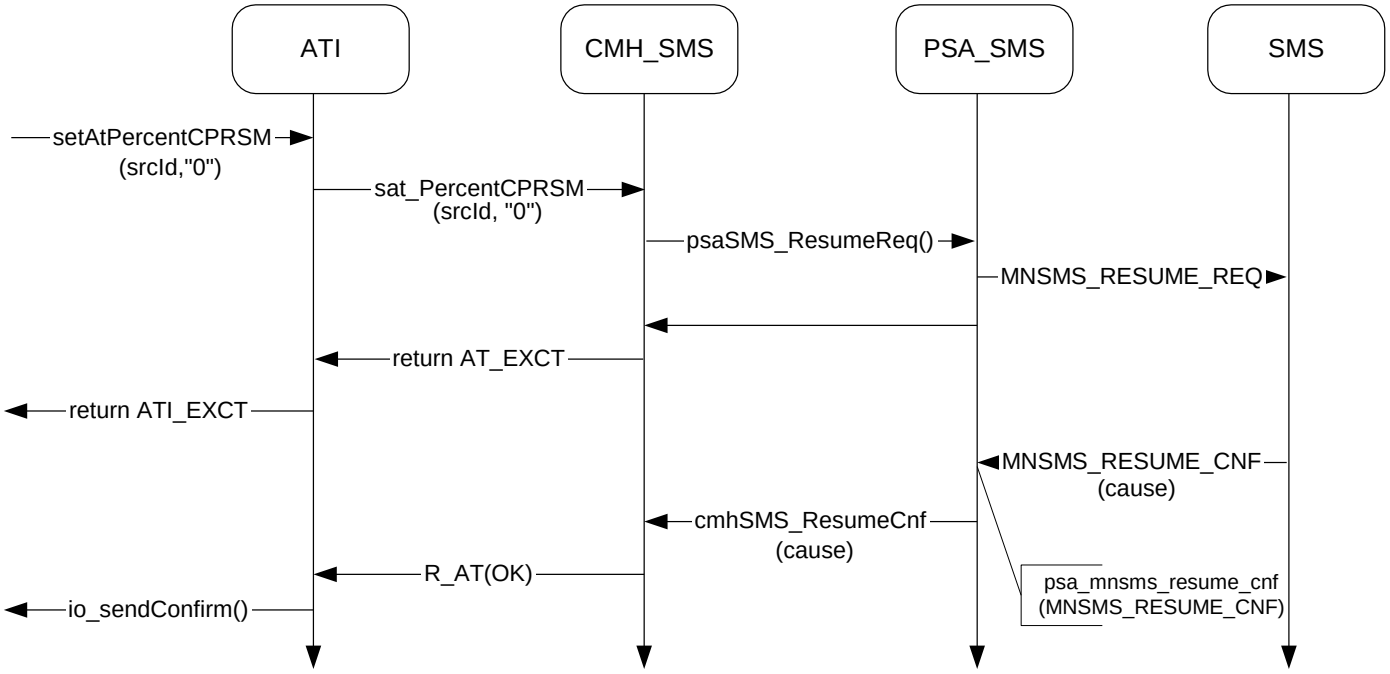
3.3 ACI Flowcharts

3.3.1 Using +CPRSM for setting Receiving of SMS to Pause, returning okay



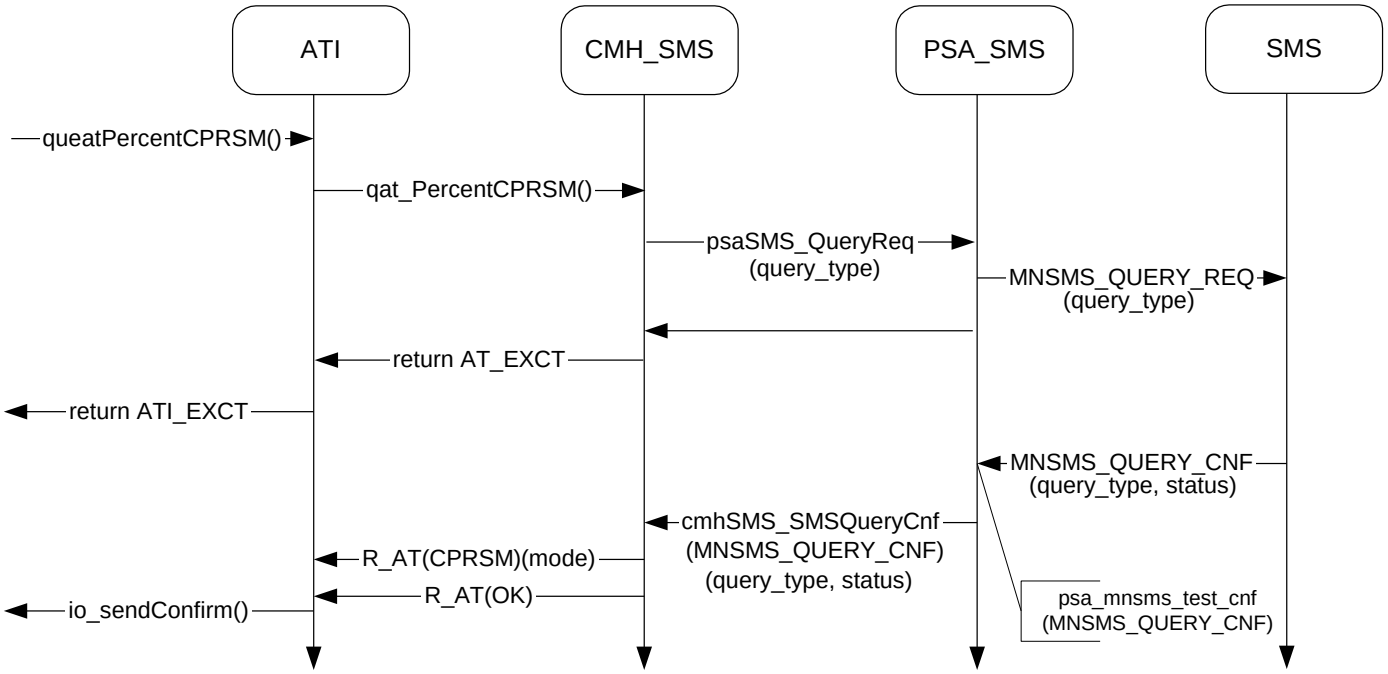
Scheme 5: Describes the ACI function flow for scenario described in scheme 1.

3.3.2 Using +CPRSM for setting Receiving of SMS to Resume, returning okay



Scheme 6: Describes the ACI function flow for scenario described in scheme 2.

3.3.3 Using +CPRSM? for requesting the status, returning okay



Scheme 7: Describes the ACI function flow for scenario described in scheme 4. The MNSMS_TEST_REQUEST is a general primitive for requesting SMS parameters. The requested parameter has to be specified by query_type. It provides it by sending a MNSMS_QUERY_CNF containing the kind of requested value and the value.

3.3.4 Using +CPRSM=? for getting possible command parameters

This response to tesat_PercentCPRSM is automatically generated within the cmd[] table.

4 Modifications in non-ACI modules

4.1 MFW

4.1.1 mfw_cb.c

Changes:

Added callback function for qAT_PercentCPRSM: rAT_PercentCPRSM for signaling the received mode value to the MMI. Implemented as dummy function.

4.2 SMI

4.2.1 smi_dmy.c

Changes:

added rAT_PercentCPRSM dummy function.

5 Modifications in ACI modules

5.1 ATI

5.1.1 ati_cmd.c

Changes:

- Added prototypes for setatPercentCPRSM and queatPercentCPRSM
- Added %CPRSM command into command table, including the response for the test command.

5.1.2 ati_ret.c

Changes:

- Added rCI_PercentCPRSM callback for converting the result to command interpreter output

5.1.3 ati_ret.c

Changes:

- Added body for setatPercentCPRSM and queatPercentCPRSM

5.2 CMH

5.2.1 cmh.h

Changes:

- Added RAT_CPRSM into RAT_ID enumeration

5.2.2 cmh_f.c

Changes:

- Added CB_VC(PercentCRSM) entry into R_AT jump table RATJmpTbl

5.2.3 cmh_sms.h

Changes:

- Added prototypes of cmhSMS_SMSResumeCnf and cmhSMS_SMSQueryCnf
- Added prototype of auxiliary function cmhSMS_convertDeliverStatusToACI.

5.2.4 cmh_smsf.c

Changes:

- Added body of cmhSMS_convertDeliverStatusToACI function. This function is used to convert the SMS receiving mode to ACI's T_ACI_CPRSM_MOD format.

5.2.5 cmh_smsr.c

Changes:

- Added body of cmhSMS_SMSResumeCnf and cmhSMS_SMSQueryCnf function for processing either a MNSMS_RESUME_CNF or a MNSMS_QUERY_CNF received from network.

5.2.6 cmh_smss.c

Changes:

- Added bodies of the functional inface sAT_PercentCPRSM and qAT_PercentCPRSM

5.2.7 aci.h

Changes:

- Added definitions and prototypes for PSA functions psa_mnsms_resume_cnf and psa_mnsms_query_cnf

5.2.8 aci_cmh.h

Changes:

- Added AT_CPRSM = 217 to T_ACI_AT_CMD enumeration
- Added T_ACI_CPRSM_MOD enumeration
- Added prototypes for functional interfaces sAT_PercentCPRSM and qAT_PercentCPRSM
- Added two rAT_PercentCPRSM prototypes (void and with parameter).
- Added two rCI_PercentCPRSM prototypes (void and with parameter).

5.2.9 aci_pei.c

Changes:

- Added mapping of PSA functions psa_mnsms_resume_cnf and psa_mnsms_query_cnf to respective primitives.

5.3 PSA

5.3.1 psa_sms.h

Changes:

- Added prototypes for psaSMS_PauseReq, psaSMS_ResumeReq and psaSMSQueryReq.

5.3.2 psa_smsp.c

Changes:

- Added bodies for PSA functions psa_mnsms_resume_cnf and psa_mnsms_query_cnf

5.3.3 psa_smss.c

Changes:

- Added bodies for psaSMS_PauseReq, psaSMS_ResumeReq and psaSMSQueryReq. These functions are responsible for sending the respective primitives to SMS.

6 Tests

6.1 Simulation test

Since the MNSMS SAP provides new primitives for supporting the %CPRSM command, the communication of ACI using this primitives is simulated by the following testcases:

6.1.1 Set command

The simulation testcases ASC620, ASC621 and ASC624 simulate the set command

6.1.1.1 ASC620

The "AT%CPRSM=1" command sets the receiving of SMS successfully to pause.

6.1.1.2 ASC621

The "AT%CPRSM=0" command sets the receiving of SMS successfully to resume. The command requires a MNSMS_RESUME_CNF primitive to be acknowledged by SMS.

6.1.1.3 ASC624

This testcase executes the ASC621 after ASC620.

6.1.2 Test command

6.1.2.1 ASC622

Simulates the "AT%CPRSM=?" command to get the parameter set that can be used by the set command.

6.1.3 Query command

6.1.3.1 ASC623

The "AT%CPRSM?" command requests the current setting from SMS. The parameters are provided by a MNSMS_QUERY_CNF primitive and have to be displayed to the user.

6.1.4 Mixed commands

6.1.4.1 ASC625

This testcase executes ASC621 after ASC623

6.1.4.2 ASC626

This testcase executes ASC623 after ASC621