



LLD Interface NITZ Feature

Project	TCS 3.1.x
Document Type	Technical Documentation
Title	LLD Interface NITZ Feature
Author	Liyi Yu
Creation Date	23. 04. 2004
Last Modified	
ID and Version	To Be Assigned
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	23.04.2004	Being processed

0.2 References

- [1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
- [2] 3GPP TS 04.08: "Mobile radio interface layer 3 specification".
- [3] 3GPP TS 27.007: "AT command set for UE".

0.3 Abbreviations, Terms

ACI	Application Control Interface (AT Commands Interpreter)
MM	Mobility Management
MMI	Man Machine Interface
MT	Mobile Terminal
NI	Network Identity
NITZ	Network Identity and Time Zone
TZ	Time Zone

Table of Contents

1	Introduction	2
2	Overview	3
2.1	General	3
2.2	Detailed Description	3
2.2.1	Standard Implementation	3
2.2.2	Customized Implementation	3
3	Implementation	5
3.1	Time Zone Automatic Update +CTZU	5
3.1.1	Description	5
3.1.2	Implementation:	5
3.1.2.1	CTZUMode.....	5
3.1.2.2	setatPlusCTZU	5
3.1.2.3	sAT_PlusCTZU()	6
3.1.2.4	queatPlusCTZU ()	6
3.1.2.5	qAT_PlusCTZU()	6
3.2	Time and Date Report %CTZV	7
3.2.1	Description	7
3.2.2	Implementation:	7
3.2.2.1	PCTZVMode	7
3.2.2.2	setatPercentCTZV.....	8
3.2.2.3	sAT_PercentCTZV()	8
3.2.2.4	queatPercentCTZV ()	8
3.2.2.5	qAT_PercentCTZV()	9
3.3	Operator Name Report/Query %CNIV	9
3.3.1	Overview	9
3.3.2	Proposal 1	9
3.3.2.1	Description	9
3.3.2.2	Implementation	10
3.3.2.2.1	setatPercentCNIV	10
3.3.2.2.2	sAT_PercentCNIV()	10
3.3.2.2.3	qAT_PercentCNIV()	11
3.3.2.2.4	queatPercentCNIV ()	11
3.3.3	Proposal 2.....	11
3.4	Interface to Application.....	12
3.4.1	Implementation:	12
3.4.1.1	T_MMR_INFO_IND	12
3.4.1.2	rCI_PlusCTZV () & rAT_PlusCTZV()	13
3.4.1.3	rCI_PercentCTZV () & rAT_PercentCTZV()	13
3.4.1.4	rCI_PercentCNIV () & rAT_PercentCNIV()	14
3.5	MMR_INFO_IND Handling	14
3.5.1	Description	14
3.5.2	psa_mmr_info_ind()	14
4	Test Plan.....	16

Table of Figures

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 is a Low Level Design (LLD) document, which describes the implementation of the NITZ feature related functionalities in ACI.

NITZ is a new, optional feature in GSM Phase 2+. It allows the network operator to transmit different types of information, network name and the current time information like current time and current time zone. The presence of them is optional.

When ACI receives the information from MM, ACI will handle the different types of information accordingly. Detailed description is provided below.

2.2 Detailed Description

2.2.1 Standard Implementation

Recommendation GSM 04.08 sub clause 4.3.6 describes the MM information procedure.

3GPP TS 27.007 describes two NITZ feature related AT commands.

+CTZU:

It is a standard AT command for automatic Time Zone Update. If this feature is set to on, the time zone information of the MT will be automatically updated. The implementation of this AT command is described in section 3.1.

+CTZR:

It is a standard AT command for Time Zone Reporting. If this is set to on, an unsolicited report about the current TZ will be sent to the MT in the following syntax when TZ has been changed.

+CTZV: <tz>

This AT command has been implemented in the ACI and will not be described here.

2.2.2 Customized Implementation

Other than the two standard AT commands, the implementation described here will also support the following customized NITZ related features.

%CTZV:

It is a customized AT command for automatic time and date reporting. For the standard implementation, only the TZ information will be reported to the MT. In the implementation here, the time and date information will also be reported to the MT if this command is set to %CTZV=1. The format of the unsolicited report is:

%CTZV: "yy/mm/dd,hh:mm:ss+/-tz", where letters indicate year (two last digits), month, day, hour, minutes, seconds and time zone.

%CNIV:

It is a customized AT command for automatic network identity reporting. Depends on the different usage of the network ID information there are two different options to implement this:

1. ACI provides an unsolicited report about the network identity (NI) to MT when NI has changed
2. ACI provides an unsolicited report about the NI to MT and stores the up to date NI in the non-volatile memory so that the NI can be access any time via %CNIV.

According to the different options two proposals are provided in section 3.3.

3 Implementation

3.1 Time Zone Automatic Update +CTZU

3.1.1 Description

Command	Possible response(s)
+CTZU = <onoff>	+CME ERROR: <err>
+CTZU?	+CTZU: <onoff> +CME ERROR: <err>
+CTZU=?	+CTZU: (list of supported <onoff>s) +CME ERROR: <err>

Set command enables and disables automatic time zone update via NITZ. If setting fails in an MT error, +CME ERROR: <err> is returned.

Read command returns the current settings in the MT.

Test command returns supported on- and off-values.

<on/off>: integer type value indicating:

- 0 – Disable automatic time zone update via NITZ (default).
- 1 – Enable automatic time zone update via NITZ.

3.1.2 Implementation:

3.1.2.1 CTZUMode

Definition:

typedef enum

```
{
    CTZU_MODE_OFF = 0,
    CTZU_MODE_ON
} T_ACI_CTZU_MODE;
```

Use:

This is the CTZU flag, which indicates if the automatic TZ update is on or off. This mode is set by calling the setting function of +CTZU, can be queried by the query function of +CTZU.

3.1.2.2 setatPlusCTZU

Prototype:

GLOBAL T_ATI_RSLT setatPlusCTZU (char *cl, UBYTE srcId)

Parameters:

cl	command line;
----	---------------

srcId	source Id.
-------	------------

Returns:

T_ATI_RSLT

Description:

This is the implementation of +CTZU AT command in the ATI side, which will actually call the functional counterpart sAT_PlusCTZU() in ACI.

3.1.2.3 sAT_PlusCTZU()

Prototype:

GLOBAL T_ACI_RETURN sAT_PlusCTZU (T_ACI_CMD_SRC srcId, T_ACI_CTZU_MODE mode)

Parameters:

SrcId	Source Id;
-------	------------

Mode CTZU mode.

Returns:

T_ACI_RETURN

Description:

This is the functional counterpart to the `+CTZU AT` command, which sets the status of CTZU mode, indicating whether time zone automatic update feature is enabled or disabled.

3.1.2.4 queatPlusCTZU ()

Prototype:

GLOBAL T_ATI_RSLT queatPlusCTZU (char *cl, UBYTE srcId)

Parameters:

cl	command line;
----	---------------

srcId source Id.

Returns:

T_ATI_RSLT

Description:

This is the implementation of +CTZU query AT command in the ATI side, which will actually call the functional counterpart qAT_PlusCTZU() in A CI.

3.1.2.5 qAT_PlusCTZU0

Prototype:

GLOBAL T_ACI_RETURN qAT_PlusCTZU (T_ACI_CMD_SRC srcId, T_ACI_CTZU_MODE *mode)

Parameters:

SrcId Source Id;
Mode CTZU mode.

Returns:

T_ACI_RETURN

Description:

This is the functional counterpart to the +CTZU AT command, which will query the status of CTZU mode, indicating whether TZ automatic update feature is enabled or disabled.

3.2 Time and Date Report %CTZV

3.2.1 Description

Command	Possible response(s)
%CTZV = <onoff>	+CME ERROR: <err>
%CTZV?	%CTZV: <onoff> +CME ERROR: <err>
%CTZV=?	%CTZV: (list of supported <onoff>s) +CME ERROR: <err>

Set command enables and disables automatic time and date report to the MT when time and date information is received from the network. If setting fails in an MT error, +CME ERROR: <err> is returned.

Read command returns the current settings in the MT.

Test command returns supported on- and off-values.

<on/off>: integer type value indicating:

- 0 – Disable automatic time and date reporting (default).
- 1 – Enable automatic time and date reporting.

3.2.2 Implementation:

3.2.2.1 PCTZVMode

Definition:

typedef enum

```
{
    PCTZV_MODE_OFF = 0,
    PCTZV_MODE_ON
```

```
} T_ACI_PCTZV_MODE;
```

Use:

This is the %CTZV flag, which indicates if the automatic time and date report is on or off. This mode is set by calling the setting function of %CTZV, can be queried by the query function of %CTZV.

3.2.2.2 setatPercentCTZV

Prototype:

```
GLOBAL T_ATI_RSLT setatPercentCTZV (char *cl, UBYTE srcId)
```

Parameters:

cl	command line;
srcId	source Id.

Returns:

T_ATI_RSLT

Description:

This is the implementation of %CTZV AT command in the ATI side, which will actually call the functional counterpart sAT_PercentCTZV() in ACI.

3.2.2.3 sAT_PercentCTZV()

Prototype:

```
GLOBAL T_ACI_RETURN sAT_PercentCTZV (T_ACI_CMD_SRC srcId, T_ACI_PCTZV_MODE mode)
```

Parameters:

SrcId	Source Id;
Mode	PCTZV mode.

Returns:

T_ACI_RETURN

Description:

This is the functional counterpart to the %CTZV AT command, which sets the status of %CTZV mode, indicating whether time and date report feature is enabled or disabled.

3.2.2.4 queatPercentCTZV()

Prototype:

```
GLOBAL T_ATI_RSLT queatPercentCTZV (char *cl, UBYTE srcId)
```

Parameters:

cl	command line;
----	---------------

srcId source Id.

Returns:

T ATI RSLT

Description:

This is the implementation of %CTZV query AT command in the ATI side, which will actually call the functional counterpart qAT_PercentCTZV() in A CI.

3.2.2.5 qAT_PercentCTZV()

Prototype:

```
GLOBAL T_ACI_RETURN qAT_PercentCTZV ( T_ACI_CMD_SRC srcId, T_ACI_PCTZV_MODE *mode)
```

Parameters:

SrcId	Source Id;
Mode	%CTZV mode.

Returns:

T ACI RETURN

Description:

This is the functional counterpart to the %CTZV AT command, which will query the status of %CTZV mode, indicating whether time and date report feature is enabled or disabled.

3.3 Operator Name Report/Query %CNIV

3.3.1 Overview

There are two different ways to make use of the NITZ network name information. One possibility is to just report the operator name and the related information to the MT when NI is changed. The other possibility is to report the operator related information to MT and store the name in the non-volatile memory and provide a query AT command to access the NI name any time. According to these two possibilities, two proposals are provided below.

3.3.2 Proposal 1

3.3.2.1 Description

This proposal is a simple way of making use of the operator name information. When ACI receives the NITZ primitive MMR_INFO_IND, ACI checks the CNIV mode. If the CNIV mode is activated, ACI will provide an unsolicited report to the MT in the following format:

%CNIV: “<full name>”, “<short name>”, “<plmn>”

<full name>:

Full network name in string, up to 40 characters

<short name>:

Short network name in string, up to 40 characters

$\langle plmn \rangle$:

The plmn of the network in digits, has the following structure: (country code digit 3)(country code digit 2)(country code digit 1)(network code digit 2)(network code digit 1)

%CNIV related AT commands are listed below:

Command	Possible response(s)
%CNIV = <onoff>	+CME ERROR: <err>
%CNIV?	%CNIV: <onoff> +CME ERROR: <err>
%CNIV=?	%CNIV: (list of supported <onoff>s) +CME ERROR: <err>

Set command enables and disables network identity report to the MT when NI is received from the network. If setting fails in an MT error, +CME ERROR: <err> is returned.

Read command returns the current settings in the MT.

Test command returns supported on- and off-values.

<on/off>: integer type value indicating:

- 0 – Disable NI reporting (default).
- 1 – Enable NI reporting.

3.3.2.2 Implementation

3.3.2.2.1 **setatPercentCNIV**

Prototype:

GLOBAL T_ATI_RSLT setatPercentCNIV (char *cl, UBYTE srcId)

Parameters:

cl	command line;
----	---------------

srcId	source Id.
-------	------------

Returns:

TABLE 1

Description:

This is the implementation of %CNIV AT command in the ATI side, which will actually call the functional counterpart sAT_PercentCNIV() in ACL.

3.3.2.2.2 sAT PercentCNIV()

Prototype:

GLOBAL T ACI RETURN sAT PercentCNIV (T ACI CMD SRC srcId,T ACI CNIV MODE mode)

Parameters:

SrcId	Source Id;
-------	------------

Mode CNIV mode.

Returns:

T_ACI_RETURN

Description:

This is the functional counterpart to the %CNIV AT command, which sets the status of %CNIV mode, indicating whether NI report feature is enabled or disabled.

3.3.2.2.3 qAT_PercentCNIV()

Prototype:

GLOBAL T_ACI_RETURN qAT_PercentCNIV (T_ACI_CMD_SRC srcId, T_ACI_CNIV_MODE *mode)

Parameters:

SrcId	Source Id;
Mode	CNIV mode.

Returns:

T_ACI_RETURN

Description:

This is the functional counterpart to the %CNIV AT command, which will query the current status of CNIV mode.

3.3.2.2.4 queatPercentCNIV ()

Prototype:

GLOBAL T_ATI_RSLT queatPercentCNIV (char *cl, UBYTE srcId)

Parameters:

cl	command line;
srcId	source Id.

Returns:

T_ATI_RSLT

Description:

This is the implementation of %CNIV query AT command in the ATI side, which will actually call the functional counterpart qAT_PercentCNIV() in ACI.

3.3.3 Proposal 2

Proposal 2 includes proposal 1 but with the following different functionalities in the query function.

The query of the AT command %CNIV will look into the non-volatile memory where the most up to date NI name is stored and then return the NI name together with the status of the CNIV mode. The network operator name will be stored every time when ACI receives the new network name via the NITZ primitive MMR_INFO_IND.

Command	Possible response(s)
%CNIV?	%CNIV:on/off, name of NI +CME ERROR: <err>

Proposal 1 was chosen for the current implementation after discussion.

3.4 Interface to Application

3.4.1 Implementation:

3.4.1.1 T_MMR_INFO_IND

```
#define MMR_MAX_TEXT_LEN 40
#define SIZE_MCC 3
#define SIZE_MNC 3
typedef struct
{
    U8                v_plmn;           /*valid flag*/
    U8                mcc[SIZE_MCC];    /*mobile country code*/
    U8                mnc[SIZE_MNC];    /*mobile network code*/
    U8                _align0;          /*alignment*/
} T_plmn;

typedef struct
{
    U8                v_name;           /*valid flag*/
    U8                dcs;              /*data coding scheme*/
    U8                add_ci;           /*add ci indicator*/
    U8                num_spare;        /*num spare bits*/
    U8                c_text;           /*counter*/
    U8                text[MMR_MAX_TEXT_LEN]; /*name*/
    U8                _align0;          /*alignment*/
    U8                _align1;          /*alignment*/
    U8                _align2;          /*alignment*/
} T_full_name;

typedef T_full_name T_short_name;

typedef struct
{
    U8                v_tz;             /*timezone valid*/
    U8                tz;               /*timezone*/
    U8                _align0;          /*alignment*/
    U8                _align1;          /*alignment*/
} T_ntz;

typedef struct
{
    U8                v_time;           /*time valid*/
```

```

U8          year;          /*year*/
U8          month;         /*month*/
U8          day;           /*day*/
U8          hour;          /*hour*/
U8          minute;        /*minute*/
U8          second;        /*second*/
U8          _align0;       /*alignment*/
} T_time;

typedef struct
{
    T_plmn          plmn;          /*plmn identification */
    T_full_name     full_name;     /*Network name,long format*/
    T_short_name    short_name;    /*Network name, short format*/
    T_ntz           ntz;           /*Network time zone*/
    T_time          time;          /*Network time*/
} T_MMR_INFO_IND;

```

3.4.1.2 rCI_PlusCTZV () & rAT_PlusCTZV()

Prototype:

If the flag FF_TIMEZONE is on:

GLOBAL void rCI_PlusCTZV (S32 time zone)

GLOBAL void rAT_PlusCTZV (S32 time zone)

Otherwise:

GLOBAL void rCI_PlusCTZV (UBYTE* time zone)

GLOBAL void rAT_PlusCTZV (UBYTE* time zone)

Parameters:

time zone time zone

Returns:

void

Description:

This is the implementation of +CTZV call back function. It reports the time zone information to the application. The time zone here is the local time zone.

3.4.1.3 rCI_PercentCTZV () & rAT_PercentCTZV()

Prototype:

GLOBAL void rCI_PercentCTZV (T_MMR_INFO_IND *mmr_info_ind, S32 time zone)

GLOBAL void rAT_PercentCTZV (T_MMR_INFO_IND *mmr_info_ind, S32 time zone)

Parameters:

mmr_info_ind an MMR primitive indication, which contains time zone info and time info

Returns:

void

Description:

This is the implementation of %CTZV call back function. It passes the time zone and time information to the application. The time zone here is the local time zone. The time reported here is the GMT network time.

3.4.1.4 rCI_PercentCNIV () & rAT_PercentCNIV()

Prototype:

```
void rCI_PercentCNIV (T_MMR_INFO_IND *mmr_info_ind)
```

```
void rAT_PercentCNIV(T_MMR_INFO_IND*mmr_info_ind)
```

Parameters:

mmr_info_ind defined in 3.4.1.1

Returns:

void

Description:

This is the implementation of %CNIV call back function. It passes the entire information to the application.

3.5 MMR_INFO_IND Handling

3.5.1 Description

In ACI the functional interface for this primitive has been implemented but the new functionality should be added. Since this primitive can contain three different types of information: network name, time zone, universal time and time zone and they are optional. According to the three types of information, the handling of the primitive is implemented as described below.

3.5.2 psa_mmr_info_ind()

Prototype:

```
GLOBAL const void psa_mmr_info_ind ( T_MMR_INFO_IND *mmr_info_ind )
```

Parameters:

mmr_info_ind	This parameter is a struct which includes information about time, time zone and network name.
--------------	---

Returns:

Void id

Changes:

Since the time information in the MM information is universal time, and in the RTC module the local time is stored. So it is necessary to store the time zone information in the non-volatile memory for later reference in case only time zone is received.

According to the information passed in by primitive MMR_INFO_IND, the following scenarios will be covered:

1. Time Zone Information Only:

When the primitive is received, the CTZU mode will be checked, if it is set to on, ACI will reset the time zone stored in RTC by using the RTC API: RTC_SetCurrentTZ(T_RTC_TZ currentTimeZone). If the CTZU is off nothing will be done.

The CTZR mode will also be checked to see if the non-solicited report should be provided. If the CTZR mode is set to on, ACI will give a non-solicited report to the MT about the time zone information in the following syntax:

+CTZV: <tz>

2. Universal Time and Time Zone Information:

When the primitive is received, the CTZU mode will be checked, if it is set to on, ACI will reset the time zone stored in RTC by using the RTC API: RTC_SetCurrentTZ(T_RTC_TZ currentTimeZone).. The new time is based on the network time because it is more accurate than the RTC time. If the CTZU is off nothing will be done.

The CTZR mode will be checked to see if the non-solicited report should be provided. If the CTZR mode is set to on, ACI will give a non-solicited report to the MT about the time zone information in the following syntax:

+CTZV: <tz>

The PCTZV mode will also be checked. If it is set to on, ACI will supply a non-solicited report to the MT about the time and date in the following syntax:

%CTZV: "yy/mm/dd,hh:mm:ss+/-tz", where letters indicate year (two last digits), month, day, hour, minutes, seconds and time zone.

3. Network Name Information:

If the CNIV mode is set to on, for proposal 1, the following non-solicited information will be sent to the MT:

%CNIV: long alphanumeric <full_name>,short alphanumeric <short_name>,numeric <plmn>, where the first parameter is the long network name, the second parameter is the short network name and the third parameter is the plmn of the network.

For proposal 2, the following non-solicited information will be sent to the MT:

%CNIV: <name>

And then the operator name will be stored in the non-volatile memory for later reference. Since the FFS has very limited memory, a look up table in ACI will be considered.

Please note that the NITZ feature only applies for Calyps o+ and P2 sample.

4 Test Plan

TBD