



Technical Document

**LLD SPECIFICATION. CALLED LINE
IDENTIFICATION PRESENTATION +CDIP.**

Document Number:	20_04_04_02519
Version:	0.1
Status:	Approved
Approval Authority:	
Creation Date:	2005-May-19
Last changed:	2015-Mar-08 by x0001198
File Name:	lld_aci_cdip_xt6.doc

Important Notice

Texas Instruments Incorporated and/or its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products, software and services at any time and to discontinue any product, software or service without notice. Customers should obtain the latest relevant information during product design and before placing orders and should verify that such information is current and complete.

All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI products, software and/or services. To minimize the risks associated with customer products and applications, customers should provide adequate design, testing and operating safeguards.

Any access to and/or use of TI software described in this document is subject to Customers entering into formal license agreements and payment of associated license fees. TI software may solely be used and/or copied subject to and strictly in accordance with all the terms of such license agreements.

Customer acknowledges and agrees that TI products and/or software may be based on or implement industry recognized standards and that certain third parties may claim intellectual property rights therein. The supply of products and/or the licensing of software does not convey a license from TI to any third party intellectual property rights and TI expressly disclaims liability for infringement of third party intellectual property rights.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products, software or services are used.

Information published by TI regarding third-party products, software or services does not constitute a license from TI to use such products, software or services or a warranty, endorsement thereof or statement regarding their availability. Use of such information, products, software or services may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

No part of this document may be reproduced or transmitted in any form or by any means, electronically or mechanically, including photocopying and recording, for any purpose without the express written permission of TI.

Change History

Date	Changed by	Approved by	Version	Status	Notes
2005-Aug-29	x0001198		0.1	Approved	1

Notes:

1. Initial version

Table of Contents

1	Introduction	4
2	Interface changes	5
3	Proposed Low Level Design.	6
3.1	Interface Changes	6
3.2	ATI Modifications.....	6
3.2.1	New Functions:	6
3.2.2	Affected Global variables:	6
3.2.3	Description of the changes:	7
3.3	CMH Modications	7
3.3.1	New Functions	Error! Bookmark not defined.
3.3.2	New Callback:.....	7
3.4	BAT.....	9
3.4.1	BAT Testing.....	9
4	Simulation Tests	10
5	List of Reference s.....	11

1 Introduction

This +CDIP command is related to a network service that provides "multiple called numbers (called line identifications) service" to an MT. This command enables a called subscriber to get the called line identification of the called party when receiving a mobile terminated call.

2 Interface changes

For AT I this will be as defined in ref [1] section 7.9:

+CDIP parameter command syntax

Command	Possible response(s)
+CDIP= [<n>]	OK ERROR
+CDIP?	+CDIP: <n>, <m> OK
+CDIP=?	+CDIP: (list of supported <n>s)

This command enables a called subscriber to get the called line identification of the called party when receiving a mobile terminated call. Set command enables or disables the presentation of the called line identifications at the TE.

When the presentation of the called line identification at the TE is enabled,
+CDIP: <number>, <type>[, <subaddr>, <satype>] response is returned only once after the first RING.

Defined values

<n> (parameter sets/shows the result code presentation status in the MT/TA):

0 disable

1 enable

<m> (parameter shows the subscriber "multiple called numbers" service status in the network):

0 "multiple called numbers service" is not provisioned

1 "multiple called numbers service" is provisioned

2 unknown (e.g. no network, etc.)

<number>: string type phone number of format specified by <type>

<type>: type of address octet in integer format (refer GSM 04.08 [8] subclause 10.5.4.7)

<subaddr>: string type subaddress of format specified by <satype>

<satype>: type of subaddress octet in integer format (refer GSM 04.08 [8] subclause 10.5.4.8)

As we do not know anything about the provisioning of "multiple called numbers service", the
+CDIP: <n>, <m> result code will have an <m> parameter value of 2 for "unknown".

This command is comparable to the +CLIP command and its related unsolicited result code.

3 Proposed Low Level Design.

3.1 Interface Changes

The +CDIP command will be defined as described above.

3.2 ATI Modifications.

3.2.1 New Functions:

-setatPlusCDIP() will handle the +CDIP set command.
-queatPlusCDIP() will handle the +CDIP query command

3.2.2 Affected Global variables:

ati_cmd.c:

The ATI array "cmds" will be updated with the new functions as follow:

setatPlusCDIP()

queatPlusCDIP()

For the test command echo the following string will be added to the ATI array "cmds" to be handled via test_gen.

"%s: (0, 1)"

aci_cmd.h:

Addition of the ID AT_CMD_CDIP will be necessary to type T_ACI_AT_CMD for call to the functions above.

<m> Typedef T_ACI_CDIP_STAT will be defined to represent this parameter which shows provisioning of "multiple called numbers service" supported by the network. Parameter value of 2 for "unknown" will be returned as we are not aware of any such network provisioning.

```
typedef enum          /* +CDIP parameter <stat> */
{
    CDIP_STAT_NotPresent = -1,
    CDIP_STAT_NotProv,
    CDIP_STAT_Prov,
    CDIP_STAT_Unknown
}
T_ACI_CDIP_STAT;
```

aci_cmd.h:

<n> Sets/shows the result code presentation status in the MT/TA.

CDIP_stat will be added to type T_ATI_USER_OUTPUT_CFG for the global variable
ati_user_output_cfg[CMD_SRC_MAX] for set/query functionality of the +CDIP command.

3.2.3 Description of the changes:

Ati_cc.c:

+CDIP Set and Query functions will do the following:

setatPlusCDIP() will call GenAtCR_C(cl, srcId, AT_CMD_CDIP).

queatPlusCDIP() will display the current settings of parameters <n> and <m>.

GenAtCR_C () will be modified to include case for AT_CMD_CDIP where ati_user_output_cfg[] will be updated as shown below for the AT command parameter <n>.

```
case( AT_CMD_CDIP ):
    ati_user_output_cfg[srcId].CDIP_stat = x;
```

3.3 CMH Modifications

3.3.1 New Callback:

A Callback is required for when the presentation of the called line identification at the TE is enabled, +CDIP:<number>,<type>[,<subaddr>,<satype>]

Here as with all other unsolicited messages we make use of the RAT_ macro to display the result code.

aci_cmh.h

RAT_CDIP will be added to type RAT_ID.

Additionally the prototype will be added as below:

```
EXTERN void rAT_PlusCDIP ( CHAR      * number,
                          T_ACI_TOA   * type,
                          CHAR        * subaddr,
                          T_ACI_TOS   * satype);
```

The rAT_PlusCDIP function definition will be implemented either in mfw_cm.c (like rAT_PlusCLIP) or in a dummy functions module. All it needs to do is return.

Ati_ret.c

Here a new ATI function rCI_PlusCDIP (as below) will be implemented which corresponds to the rAT_PlusCDIP function. The setting of ati_user_output_cfg[srcId].CDIP_stat will be used to generate the +CDIP: result code (comparable to rCI_PlusCLIP).

```
GLOBAL void rCI_PlusCDIP ( CHAR      * number,
                          T_ACI_TOA   * type,
                          CHAR        * subaddr,
                          T_ACI_TOS   * satype)
```

cmh_ssr.c:

Additional Case for +CDIP, AT_CMD_CDIP will be added where R_AT macro will also be called. R_AT macro will be called with RAT_CDIP.

Cmh_f.c:

New element for +CDIP will be added to the array RATJumpTbl

```
CB_VC( PlusCDIP ), /* RAT_CDIP */
```

Cmh_ccr.c:

In function `cmhCC_IncomingCall()`, call `R_AT` macro with the following parameters (similar as for `RAT_CLIP`):

```
R_AT ( RAT_CDIP, idx )  
      (psaCC_ctbCldAdr2Num( cld, numBuf, sizeof (numBuf) ),  
       cmhCC_ctbGetCldNumTyp( cld, &toaBuf ),  
       psaCC_ctbCldAdr2Sub( cld, subBuf ),  
       cmhCC_ctbGetCldSubTyp( cld, &tosBuf ));
```


3.4 BAT

For BAT the equivalent is:

BAT_RES_UNSPUS_CDIP T_BAT_res_unspus_cdip:

```
typedef struct
{
    U8          c_number;                /* counter */
    U8          number[BAT_MAX_CDIP_NUMBER_LEN]; /* number */
    U8          type;                    /* Type of address */
    U8          v_subaddr;               /* valid-flag */
    U8          c_subaddr;               /* counter */
    U8          subaddr[BAT_MAX_SUBADDR_LENGTH]; /* Subaddress */
    S16         satype;                  /* Type of subaddress */
} T_BAT_res_unspus_cdip;
```

The BAT_RES_UNSPUS_CDIP will follow the BAT_RES_UNSPUS_CLIP. There will be no application control over the generation of this unsolicited result code (i.e. no need to enable and disable with a BAT_CMD_SETPLUS_CDIP equivalent to AT+CDIP=[<n>] as it is always enabled with BAT).

3.4.1 BAT Testing

The BAT_RES_UNSPUS_CDIP results code will occur on every incoming call.

T_BAT_res_unspus_cdip structure in section 3.4 above will convey the number and sub address information as explained below:

```
c_number      /* Calling number count */
number        /* Calling number */
type          /* Type of address */
v_subaddr     /* valid-flag = TRUE if subaddress exists otherwise NoSubaddress = FALSE */
c_subaddr     /* Subaddress counter */
subaddr       /* Calling party Subaddress */
satype        /* Type of subaddress */
```

4 Simulation Tests

The following simulation test cases will be added:

-Testcase (ACICCC250F) - +CDIP Test Command
+CDIP=?
+CDIP: (0, 1)

-Testcase (ACICCC250E) - Obtain Called Line ID Presentation state
+CDIP?
+CDIP: 0, 2

-Testcase (ACICCC250B) - Modify Called Line ID Presentation state - Disable +CDIP
+CDIP=0 /* This will disable unsolicited results code */
OK,+CME ERROR

Incoming call...
RING
No +CDIP Unsolicited results code is received

-Testcase (ACICCC250C) - Modify Called Line ID Presentation state - Enable +CDIP
+CDIP=1 /* This will enable unsolicited results code (Default in BAT) */
OK,+CME ERROR

Incoming call...
RING

+CDIP: "01234567",129 /* +CDIP results code with no Subaddress Information */

-Testcase (ACICCC250D) - Modify Called Line ID Presentation state - Enable +CDIP
+CDIP=1 /* This will enable unsolicited results code (Default in BAT) */
OK,+CME ERROR

Incoming call...
RING

+CDIP: "+4901234567",145,"00491239",248
/* This example shows +CDIP results code with subaddress and subaddress type */

Similar functionality will be tested in the target.

5 List of References

- [1] 3GPP TS 27.007 v3.13.0