Technical Document

GSM PROTOCOL STACK

G23

RX – FIELDSTRENGTH

DRIVER INTERFACE DRAFT

Document Number:	8415.035.99.005
Version:	0.6
Status:	Draft
Approval Authority:	
Creation Date:	1999-Jun-04
Last changed:	2015-Mar-08 by XINTEGRA
File Name:	8415_035.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.

Change History

Date	Changed by	Approved by	Version	Status	Notes
1999-Jun-04	LE et al.		0.1		1
1999-Jul-05	SAB et al.		0.2		2
1999-Nov-18	TSE et al.		0.3		3
1999-Dec-07	TSE et al.		0.4		4
1999-Dec-13	TSE et al.		0.5		5
2003-May-20	XINTEGRA		0.6	Draft	

Notes:

1. Initial version
2. Rx_Status_Type Changed
3. Rx_timeout() added
4. Corrections
5. Function parameters updated

Table of Contents

List of Figures and Tables

List of References

[ISO 9000:2000] International Organization for Standardization. Quality management systems - Fundamentals and vocabulary. December 2000

1.1 References

[C_8415.026] 8415.026.99.012; March 19, 1999
Generic Driver Interface – Functional Specification; Condat

[GSM_05.08] Draft EN 300 911: November 1998 (GSM 5.08 version 6.3.0)
Radio subsystem link control; ETSI

2 Introduction

G23 is a software package implementing Layers 2 and 3 of the ETSI-defined GSM air interface signaling protocol, and as such represents the 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 standardized 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/AT Command Interface,
- MMI and MMI Framework (MFW) and
- Test and integration support tools.

This document describes the functional interface of the G23 fieldstrength management device driver interface. This driver is used to control all fieldstrength related functions. The driver does support multiple devices and therefore no open and close functionality is supported. The driver can be configured to signal different state transitions, for example, fieldstrength level has reached the "fieldstrength low" level. This is done by setting an OS signal or calling a specified call-back function.

3 Interface description of RX driver

3.1 Data types

Name	Description
UBYTE	unsigned 8 bit integer data type
BYTE	signed 8 bit integer data type
USHORT	unsigned 16 bit integer data type
SHORT	signed 16 bit integer data type
T_RX_STATUS	Status Information
T_RX_DCB	Device Control Block

3.1.1 T_RX_STATUS

Definition:

```
typedef struct T_RX_STATUS
{
  UBYTE actLevel;
  UBYTE gsmLevel;
}
```

Description:

This data type represents the driver, respectively the current fieldstrength level. The following table contains a list of elements contained in the status information data type and short descriptions of them.

Data element	Description
ActLevel	Indicates the current fieldstrength level.
GsmLevel	Indicates the current fieldstrength level acc. to the RXLEV value defined in GSM 05.08

3.1.2 T_RX_DCB – Device Control Block

Definition:

```
typedef struct T_RX_DCB
{
  UBYTE Steps
}
```

Description:

The device control block data type contains all parameters used to configure the fieldstrength driver. The following table contains a list of the data elements and short descriptions of them.

Data element	Description
Steps	Value representing the number of fieldstrength levels which can be displayed

3.2 Constants

Name	Description
RX_SIGTYPE_RXLEVEL	Signal type: new RX level
DRV_BUFFER_FULL	The internal buffer is exhausted
DRV_DISABLED	Driver is not enabled
DRV_ENABLED	Driver is enabled
DRV_NOTCONFIGURED	Driver is not configured
DRV_INITFAILURE	Driver initialization failed
DRV_INITIALIZED	Driver is already initialized
DRV_INTERNAL_ERROR	Unspecified internal driver error
DRV_INPROCESS	The requested function is currently being executed
DRV_INVALID_PARAMS	One or more parameters are out of range or invalid
DRV_NOTCONFIGURED	Driver is not configured
DRV_OK	Return value indicating the function completed successfully
DRV_UNKNOWN	Unknown device accessed
DRV_SIGFCT_NOTAVAILABLE	The requested event signaling functionality is not available

3.3 Signals

Signals are used to asynchronously inform the using process about selected events. Signaling is done by passing a signal call-back function to the driver at the time of initialization (see "3.4.1 rx_Init – Driver Initialization"). When no call-back is defined, event signaling cannot be performed.

The contents of the T_DRV_SIGNAL information structures specially defined for this driver are described in the following chapters. The contents of the T_DRV_SIGNAL information structures for the common signals are described in [C_8415.026].

3.3.1 RX_SIGTYPE_RXLEVEL

This signal is indicated when a change of the fieldstrength level has been detected. Aside from being asynchronously informed about the status change, the status may be polled by calling the function rx_GetStatus() (Chapter 3.4.5).

Paramter	Value
SignalType	RX_SIGTYPE_RXLEVEL
DataLength	Sizeof(rx_Status_Type)
UserData	Pointer to a buffer of the type rx_Status_Type

3.4 Functions

Name	Description
rx_Init	Initialization of RX
rx_Exit	Termination of RX
rx_SetConfig	Configure the driver
rx_GetConfig	Retrieve the configuration of the driver
rx_GetStatus	Retrieve the current fieldstrength value

3.4.1 rx_Init – Driver Initialization

Definition:

```
USHORT rx_Init
(
    USHORT              DrvHandle
    T_DRV_CB_FUNC       in_SignalCBPtr
    T_DRV_EXPORT **     DrvInfo
) ;
```

Parameters:

Name	Description
DrvHandle	Unique handle for this driver
in_SignalCBPtr	This parameter points to the function that is called at the time an event that is to be signaled occurs. This value can be set to NULL if event signaling should not be possible.
DrvInfo	Pointer to the driver parameters (see GDI specification document for a description of T_DRV_EXPORT).

Return values:

Name	Description
DRV_OK	Initialization successful
DRV_INITIALIZED	Driver already initialized
DRV_INITFAILURE	Initialization failed

Description

The function initializes the driver's internal data. The function returns DRV_OK in the case of a successful completion.

The function returns DRV_INITIALIZED if the driver has already been initialized and is ready to be used or is already in use. In the case of an initialization failure, i.e. the driver cannot be used, the function returns DRV_INITFAILURE.

The driver exports its properties such as its name, the functions to access driver functionality and a bitfield called flags by the parameter DrvInfo. If the driver is called by ISR, Bit (0) in the bitfield is set, otherwise this bit is cleared.

The driver stores the DrvHandle and passes it via the SignalID to the calling process each time the callback function is called.

3.4.2 rx_Exit – De-initialization of the driver

Definition:

```
void rx_Exit
(
    void
) ;
```

Parameters:

Name	Description
-	

Return values:

Name	Description
-	-

Description

This function is used to indicate to RX that the driver and its functionality are no longer needed.

3.4.3 rx_SetConfig – Set a driver configuration

Definition:

```
USHORT rx_SetConfig
(
    T_RX_DCB *          in_DCBPtr
) ;
```

Parameters:

Name	Description
in_DCBPtr	Pointer to the driver control block

Return values:

Name	Description
DRV_OK	Function successfully completed
DRV_INVALID_PARAMS	One or more values are out of range or invalid in that combination

Description

This function is used to configure the driver. For detailed information about the contents of the driver control block, refer to Chapter 3.1.2.

If any value of the configuration is out of range or invalid in combination with any other value of the configuration, the function returns DRV_INVALID_PARAMS.

Call the rx_GetConfig() function to retrieve the driver's configuration.

3.4.4 rx_GetConfig – Retrieve the driver configuration

Definition:

```
USHORT rx_GetConfig
(
    T_RX_DCB *          out_DCBPtr
) ;
```

Parameters:

Name	Description
out_DCBPtr	Pointer to the driver control block

Return values:

Name	Description
DRV_OK	Function successfully completed
DRV_NOTCONFIGURED	The driver is not yet configured

Description

This function is used to retrieve the configuration of the driver. The configuration is returned in the driver control block to which the pointer provided out_DCBPtr points. For detailed information about the contents of the driver control block, refer to Chapter 3.1.2.

If the driver is not configured, the function returns DRV_NOTCONFIGURED.

Call the rx_SetConfig() function to configure the driver.

3.4.5 rx_GetStatus – Retrieve the Driver Status

Definition:

```
USHORT rx_GetStatus
(
    T_RX_STATUS *       out_StatusPtr

) ;
```

Parameters:

Name	Description
out_StatusPtr	Pointer to the status information buffer

Return values:

Name	Description
DRV_OK	Function successfully completed
DRV_NOTCONFIGURED	The driver is not yet configured
DRV_INVALID_PARAMS	out_StatusPtr is NULL

Description

This function is used to retrieve the status of the driver, respectively the current fieldstrength value.

In the case of a successful completion, the driver returns DRV_OK and the current status of the driver to which the buffer out_StatusPtr points.

If the driver is not yet configured, it returns DRV_NOTCONFIGURED. In this case, the contents of the buffer out_StatusPtr is invalid.

If out_StatusPtr equals NULL, the driver returns DRV_INVALID_PARARMS.

Appendices

A. Acronyms

DS-WCDMA	Direct Sequence/Spread Wideband Code Division Multiple Access

B. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000)	Formerly referred to as FPLMTS (Future Public Land-Mobile Telephone System), this is the ITU's specification/family of standards for 3G. This initiative provides a global infrastructure through both satellite and terrestrial systems, for fixed and mobile phone users. The family of standards is a framework comprising a mix/blend of systems providing global roaming. <URL: http://www.imt-2000.org/>