Technical Document

GSM PROTOCOL STACK

G23

SERIAL INTERFACE DRIVER FOR

FAX & DATA SERVICES

APPLICATION PROGRAMMING INTERFACE

Document Number:	8415.049.99.005
Version:	0.6
Status:	Draft
Approval Authority:	
Creation Date:	1999-Nov-22
Last changed:	2015-Mar-08 by XINTEGRA
File Name:	Serfd_api.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-Nov-22	TSE et al.		0.1		1
1999-Dec-07	TSE et al.		0.2		2
1999-Dec-13	TSE et al.		0.3		3
1999-Dec-15	TSE et al.		0.4		4
2000-Feb-16	MGU et al.		0.5		5
2003-May-20	XINTEGRA		0.6	Draft	

Notes:

1. Initial version
2. Corrections
3. Function parameters updated
4. Restructured
5. revised

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

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 Standard Serial Receiver Transmitter (SER) driver, API. This driver can be used for common communication purposes and includes UART, USART and IRDA devices. This API of the driver is derived from the generic driver interface specification [C_8415.026]. For high performance communication applications, such as fax and data, enhanced serial drivers are used. The serial device driver sets the communication devices a c-cording to the target hardware and the settings in the application software. All SER_fd function call the corresponding UART/USART functions with the pre-configured device identifier.

NOTE: The driver devices need to be configured after initialization.

3 Interface description of the SER driver

The USART driver should be implemented as an interrupt driven module which provides several functions for character oriented asynchronous I/O to the F&D protocol stack.

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
ULONG	unsigned 32 bit integer data type
LONG	signed 32 bit integer data type
BOOL	unsigned 8 bit char data type
T_baudrate	baudrate enum type
T_bitsPerCharacter	Bits per character type
T_stopBits	Stop bits type
T_parity	Parity type
T_flowCtrlMode	Flow control mode type
T_suspendMode	Suspend parameter mode
T_reInstMode	Install mode type
T_FDRET	Type of return values of functions

3.1.1 T_baudrate - baudrate enum type

Definition:

```
typedef enum {
  FD_BAUD_AUTO,
  FD_BAUD_75,
  FD_BAUD_150,
```

```
    FD_BAUD_300,
    FD_BAUD_600,
    FD_BAUD_1200,
    FD_BAUD_2400,
    FD_BAUD_4800,
    FD_BAUD_7200,
    FD_BAUD_9600,
    FD_BAUD_14400,
    FD_BAUD_19200,
    FD_BAUD_28800,
    FD_BAUD_33900,
    FD_BAUD_38400,
    FD_BAUD_57600,
    FD_BAUD_115200,
    FD_BAUD_203125,
    FD_BAUD_406250,
    FD_BAUD_812500
} T_baudrate;
```

Description:

All defined baudrates for the serial device .

3.1.2 T_bitsPerCharacter - Bits per character type

Definition:

```
typedef enum {
    bpc_7,
    bpc_8
} T_bitsPerCharacter;
```

Description:

Defines all supported bits per character parameters.

3.1.3 T_stopBits - Stop bits type

Definition:

```
typedef enum {
    sb_1,
    sb_2
} T_stopBits;
```

Description:

Defines all supported stop bit lengths.

3.1.4 T_parity - Parity type

Definition:

```
typedef enum {
    pa_none,
    pa_even,
    pa_odd,
    pa_space
} T_parity;
```

Description:

Defines all supported parity settings.

3.1.5 T_flowCtrlMode - Flow control mode type

Definition:

```
typedef enum {
   fc_none,
   fc_rts,
   fc_dtr,
   fc_xoff
} T_flowCtrlMode;
```

Description:

Defines all supported flow control modes.

3.1.6 T_suspendMode - Suspend parameter mode

Definition:

```
typedef enum {
   sm_noSuspend,
   sm_suspend
} T_suspendMode;
```

Description:

Defines all supported suspend modes.

3.1.7 T_reInstMode - Install mode type

Definition:

```
typedef enum {
   rm_notDefined,
   rm_reInstall,
   rm_noInstall
} T_reInstMode;
```

Description:

Defines Install mode parameters used in SER_FD_Read and SER_FD_Write.

3.2 Constants

Name	Description
Return values of functions	
FD_OK	The function has returned successfully
FD_SUSPENDED	The callback is suspended until the buffer or state condition changes
FD_NOT_SUPPORTED	wrong device ID
FD_NOT_READY	a function is called while the callback of the readOutFunc function is activated and still not terminated.
FD_INTERNAL_ERR	The function has returned with an internal error
Line status	
FD_LINE_ON	
FD_LINE_OFF	
Constants for the state parameter in SER_fd_GetLineState and SER_fd_SetLineState.	
CTS	
RTS	
DSR	
DTR	
DCD	
BRK	
ESC	
TXSTP	
RXSTP	
BRKLEN	
RXBLEV	
SA	
SB	
X	
RI	
FD_MAX_BUFFER_SIZE	Size of the circular buffers used in the driver

3.3 Functions

Name	Description
SER_fd_Init	Initialization of the serial communication driver
SER_fd_Enable	The functionalities of the driver are disabled or enabled
SER_fd_SetComPar	Sets up the communication parameters
SER_fd_SetBuffer	Sets up the size of the circular buffers
SER_fd_SetFlowCtrl	Changes the flow control mode of the serial device
SER_fd_SetEscape	Set escape sequence
SER_fd_InpAvail	Returns the number of characters available in the RX buffer
SER_fd_OutpAvail	Returns the number of free characters in TX buffer
SER_fd_ReadData	Read data received via the serial interface
SER_fd_WriteData	Send data via the serial interface
SER_fd_StopRec	tells the terminal equipment that no more data can be received
SER_fd_StartRec	tells the terminal equipment that the receiver is again able to receive more data
SER_fd_GetLineState	Returns the state of the V.24 lines
SER_fd_SetLineState	Sets the states of the V.24 status lines

3.3.1 SER_fd_Init – Driver Initialization

Definition:

T_FDRET SER_fd_Init (void);

Parameters:

Name	Description
-	

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	Wrong device number.
FD_INTERNAL_ERR	Internal problem.

Description

The function initializes the communication device hardware and installs interrupt handlers.

The parameters are set to the default values:

- 19200 baud,
- 8 bits / character,
- no parity,
- 1 stop bit,
- no flow control.

All functionalities of the device driver are disabled.

3.3.2 SER_fd_Enable - The functionalities of the driver are disabled or enabled

Definition:

T_FDRET SER_fd_Enable (BOOL enable)

Parameters:

Name	Description
enable	1: enable the driver 0: disable the driver

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	Wrong device.
FD_INTERNAL_ERROR	Internal problem with the hardware.

Description:

The functions of the communication device driver are disabled or enabled. In the deactivated state, all information about the communication parameters should be stored and recalled if the driver is again enabled. When the driver is enabled, the RX and TX buffers are cleared.

3.3.3 SER_fd_SetComPar - Sets up the communication parameters

Definition:

```
T_FDRET  SER_fd_SetComPar (
        T_baudrate              baudrate,
        T_bitsPerCharacter      bpc,
        T_stopBits              sb,
        T_parity        parity)
```

Parameters:

Name	Description
baudrate	Used baud rate
bpc	Used bits per character
sb	Used stop bits
parity	Used parity

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	The specified parameters do not fit the capabilities of the device or the device ID is wrong.
FD_INTERNAL_ERROR	Internal problem with the hardware.

Description:

Sets up the communication parameters: baud rate, bits per character, number of stop bits, parity. This function should be callable any time after initialization.

3.3.4 SER_fd_SetBuffer - Sets up the size of the circular buffers

Definition:

```
T_FDRET SER_fd_SetBuffer (
        USHORT      bufSize,
        USHORT      rxThreshold,
        USHORT      txThreshold)
```

Parameters:

Name	Description
bufSize	Specifies the size of the circular buffer.
rxThreshold	Amount of received bytes that leads to a call to suspended read-out function which is passed to the function SER_fd_ReadData.
TxThreshold	Amount of bytes in the TX buffer to call the suspended write-in function which is passed to the function SER_fd_WriteData

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	bufSize exceeds the maximal possible capabilities of the driver or the threshold values do not correspond to the bufSize or the device ID is wrong.
FD_INTERNAL_ERROR	Internal problem with the hardware or the function has been called while the serial device is enabled.

Description

Sets up the size of the circular buffers to be used in the serial device. This function may be called only if the device is disabled.

3.3.5 SER_fd_SetFlowCtrl - Changes the flow control mode of the serial device

Definition:

```
T_FDRET SER_fd_SetFlowCtrl (
                T_flowCtrlMode    fcMode,
            USHORT                XON,
            USHORT                XOFF);
```

Parameters:

Name	Description
fcMode	flow control mode (none, DTR/DSR, RTS/CTS, XON/XOFF)
XON	ASCII code of the XON character
XOFF	ASCII code of the XOFF character

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	Device ID number or the flow control mode is not supported.
FD_INTERNAL_ERROR	Internal problem with the hardware.

Description

Changes the flow control mode of the serial device. If a flow control is activated, DTR is activated or XOFF is sent, if the RX buffer is not able to store the received characters,DTR is deactivated or XON is sent.

3.3.6 SER_fd_SetEscape - Set escape sequence

Definition:

```
T_FDRET SER_fd_SetEscape (
                char        escChar,
                USHORT      guardPeriod);
```

Parameters:

Name	Description
escChar	ASCII character which could appear three times as an escape sequence.
guardPeriod	Denotes the minimal duration of the rest before the first and after the last character of the escape sequence, and the maximal receiving duration of the entire escape string. This value is expressed in ms.

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	Wrong Device ID.
FD_INTERNAL_ERROR	Internal problem with the hardware.

Description

To return to the command mode at the ACI while a data connection is established, an escape sequence must be detected. To distinguish between user data and the escape sequence, a defined guard period is necessary before and after this sequence.

3.3.7 SER_fd_InpAvail - Returns the number of characters available in the RX buffer

Definition:

T_FDRET SER_fd_InpAvail (void);

Parameters:

Name	Description
-	

Return values:

Name	Description
>= 0	The returned value is the amount of data in the RX buffer.
FD_NOT_SUPPORTED	Wrong Device ID.
FD_NOT_READY	The function is called while the call-back of the readOutFunc function is activated and still not terminated.
FD_INTERNAL_ERROR	Internal problem with the hardware.

Description

Returns the number of characters available in the RX buffer of the serial device. If the device is disabled the function returns 0.

3.3.8 SER_fd_OutpAvail - Returns the number of free characters in TX buffer

Definition:

T_FDRET SER_fd_OutpAvail (void);

Parameters:

Name	Description
-	

Return values:

Name	Description
>= 0	The returned value is the amount of data in the TX buffer.
FD_NOT_SUPPORTED	Wrong Device ID.
FD_NOT_READY	The function is called while the callback of the writeInFunc function is activated and still not terminated.
FD_INTERNAL_ERROR	Internal problem with the hardware.

Description

Returns the number of free characters in TX buffer of the device. If the device is disabled, the function returns 0.

3.3.9 SER_fd_ReadData - Read data from the driver

Definition:

```
T_FDRET SER_fd_ReadData (
        T_suspendMode suspend,
        void (readOutFunc (BOOL cldFromIrq,
                T_reInstMode *reInstall,
                UBYTE nsource,
                UBYTE *source[],
                USHORT size[],
                ULONG state)))
```

Parameters:

Name	Description
suspend	mode of suspension if RX buffer empty
readOutFunc	Call-back function.
cldFromIrqThe	driver sets this parameter to 1 if the call-back function is called from an interrupt service routine.
reInstall	The call-back function sets this parameter to rm_reInstall if the driver must call again, the call-back function when the RX threshold level is reached. Otherwise, it will be set to rm_noInstall. Before calling the readOutFunc function, this parameter is set to rm_notDefined.
nsource	Informed the callback function about the number of fragments which are ready to copy from the circular RX buffer.
source	Array which contains the addresses of the fragments.
size	Array which contains the sizes of each fragments.
state	The state parameter is the status of the V.24 lines and the break / escape detection. The state parameter is described in the specification of SER_fd_GetLineState.

Return values:

Name	Description
>= 0	Successful operation. Amount of processed bytes.
FD_SUSPENDED	The call-back is suspended until the buffer or state condition changed.
FD_NOT_READY	The function is called while the callback is activated and still not terminated.
FD_NOT_SUPPORTED	Wrong device ID.
FD_INTERNAL_ERROR	Internal problems with the hardware.

Description

To read the received characters out of the RX buffer, the address of a function is passed. If characters are available, the driver calls this function and passes the address and the amount of readable characters. Because the RX buffer is circular, the call-back function may be called with more than one address of buffer fragment. The readOutFunc function modifies the contents of the size array to return to the driver the number of processed characters. Each array entry is decremented by the number of bytes read in the fragment. If the SER_fd_ReadData is called while the RX buffer is empty, it depends on the suspend parameter to suspend the call-back or to leave without any operation. In case of suspension, the return value of SER_fd_ReadData is FD_SUSPENDED. A delayed call-back will be performed if:

- the RX buffer reaches the adjusted threshold (rxThreshold of SER_fd_SetBuffer),

- the state of an input line has changed,

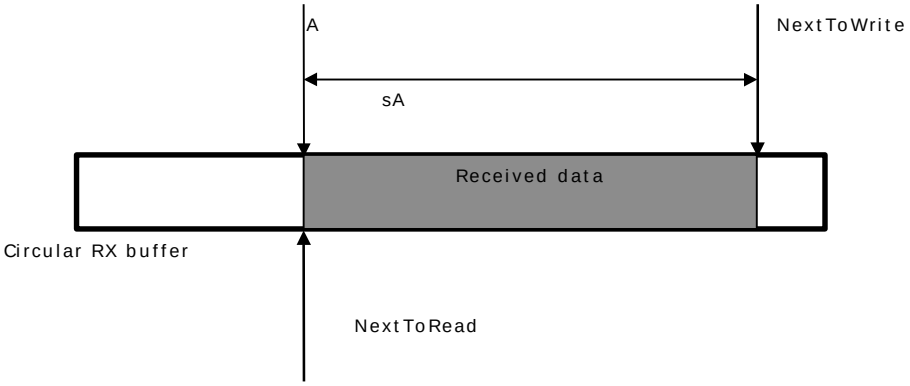
- a break is detected,

- an escape sequence is detected.

If no suspension is necessary the function returns the number of processed bytes.

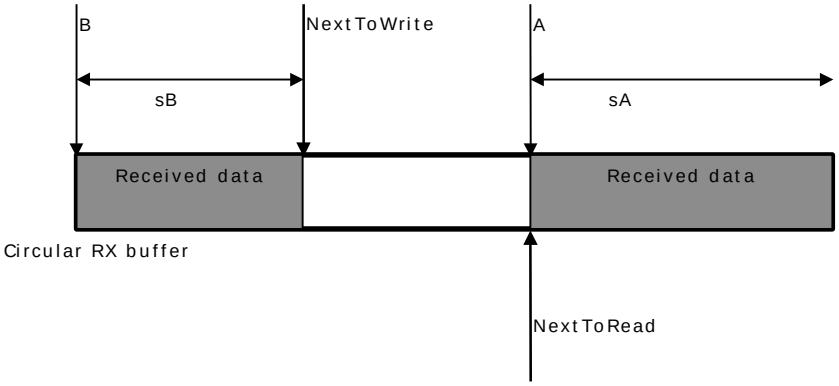
Example

This examples show the two cases of callback parameters for a circular RX buffer.



The readOutFunc should be called with nsource = 1, source[0] = A, size[0] = sA, reInstall = rm_notDefined.

The NextToRead pointer should be increased by (sA - size[0]) as soon as the content of the reInstall variable contains a value which is not equal to rm_notDefined.



The readOutFunc should be called with nsource = 2, source[0] = A, size[0] = sA

source[1] = B, size[1] = sB, reInstall = rm_notDefined

The NextToRead pointer should be increased by ((sA - size[0]) + (sB - size[1])) as soon as the content of the reInstall variable contains a value which is not equal to rm_notDefined.

After incrementing the NextToRead pointer it should be corrected with a circular algorithm.

3.3.10 SER_fd_WriteData – Transmit data via the serial interface

Definition:

```
T_FDRET SER_fd_WriteData (
        T_suspendMode suspend,
        void (writeInFunc (BOOL cldFromIrq,
                T_reInstMode *reInstall,
                UBYTE ndest,
                UBYTE *dest[],
                USHORT size[])))
```

Parameters:

Name	Description
suspend	mode of suspension in case TX buffer is full
writeInFunc	Callback function.
cldFromIrq	The driver sets this parameter to 1 if the call-back function is called from an interrupt service routine.
reInstall	The callback function sets this parameter to rm_reInstall if the driver must call the callback function when the TX threshold level is reached again. Otherwise it will be set to rm_noInstall. Before calling the readOutFunc function, this parameter is set to rm_notDefined.
ndest	Informed the callback function about the number of fragments which are available in the TX buffer.
dest	Array which contains the addresses of the fragments.
size	Array which contains the sizes of each fragments.

Return values:

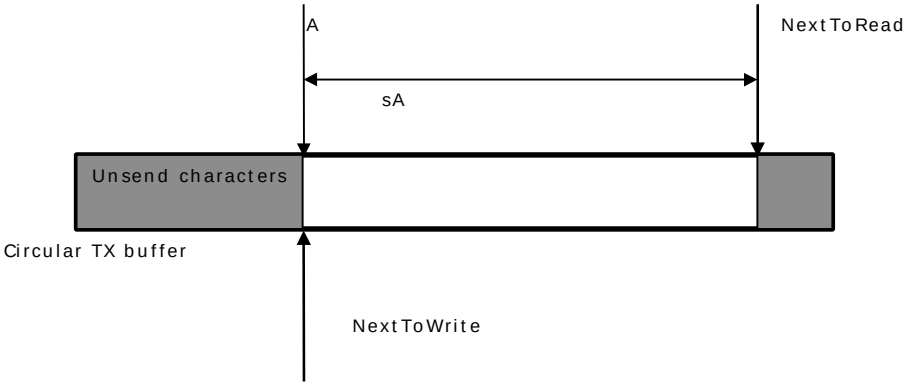
Name	Description
>= 0	Successful operation. Amount of processed bytes.
FD_SUSPENDED	The call-back is suspended until the buffer condition changed.
FD_NOT_READY	The function is called while the callback is activated and still not terminated.
FD_NOT_SUPPORTED	Wrong device ID.
FD_INTERNAL_ERROR	Internal problems with the hardware.

Description

To write characters into the TX buffer, the address of a function is passed. If free space is available in the buffer, the driver calls this function and passes the destination address and the amount of space. Because the RX buffer is circular, the call-back function may be called with more than one address of buffer fragment. The writeInFunc function modifies the contents of the size array to return the driver the number of processed bytes. Each array entry is decremented by the number of bytes written in this fragment. If the SER_fd_WriteData function is called while the TX buffer is full, it depends on the suspend parameter to suspend the call-back or to leave this function without any operation. In the case of suspension, the returned value of the SER_fd_WriteData is FD_SUSPENDED. A delayed call-back will be performed if the TX buffer reaches the adjusted threshold (txThreshold of SER_fd_SetBuffer). If no suspension is necessary, the function returns the number of processed bytes.

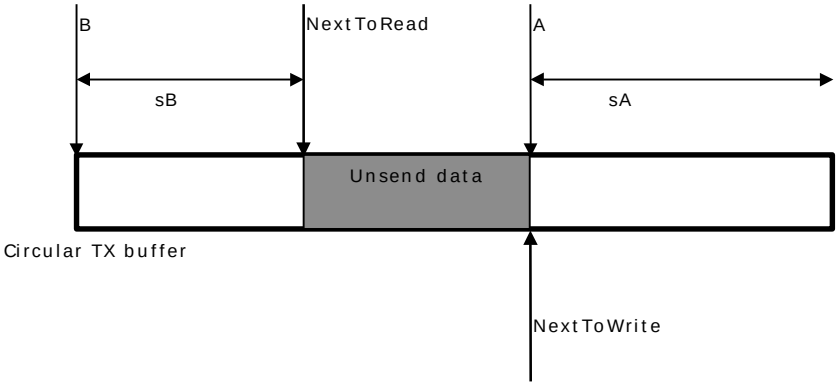
Example

These examples show the two cases of callback parameters for a circular TX buffer.



The writeInFunc should be called with ndest = 1, dest[0] = A, size[0] = sA, reInstall = rm_notDefined.

The NextToWrite pointer should be increased by (sA - size[0]) as soon as the content of the reInstall variable contains a value which is not equal to rm_notDefined.



The writeInFunc should be called with ndest = 2, dest[0] = A, size[0] = sA

dest[1] = B, size[1] = sB, reInstall = rm_notDefined

The NextToWrite pointer should be increased by ((sA - size[0]) + (sB - size[1])) as soon as the content of the reInstall variable contains a value which is not equal to rm_notDefined.

After incrementing the NextToWrite pointer, it should be corrected with a circular algorithm.

3.3.11 SER_fd_StopRec - tells the terminal equipment that no more data can be received

Definition:

T_FDRET SER_fd_StopRec (void);

Parameters:

Name	Description
-	

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	Wrong Device ID.
FD_INTERNAL_ERROR	Internal problem with the hardware.

Description

If a flow control mode is set, this function tells the terminal equipment that no more data can be received.

XON/XOFF: XOFF is sent.

DTR/DSR : DTR is deactivated.

RTS/CTS : RTS is deactivated.

3.3.12 SER_fd_StartRec - receiver is again able to receive more data

Definition:

T_FDRET SER_fd_StartRec (void);

Parameters:

Name	Description
-	

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	Wrong Device ID.
FD_INTERNAL_ERROR	Internal problem with the hardware.

Description

If a flow control mode is set, this function tells the terminal equipment that the receiver is again able to receive more data. If the buffer has already reached the high water mark, the driver sends the signal only if the buffer drains to a low water mark.

XON/XOFF: XON is sent.

DTR/DSR : DTR is activated.

RTS/CTS : RTS is activated.

3.3.13 SER_fd_GetLineState - Returns the state of the V.24 lines

Definition:

T_FDRET SER_fd_GetLineState (ULONG *state);

Parameters:

Name	Description
state	State of the V.24 lines, the flow control state and the result of the break/escape sequence detection process as a bit field. The format of the bit-field is described in the following table. Bits in this bitfield are read with the SER_fd_GetLineState function. Bits are set with the function SER_fd_SetLineState

Signal	Bitpos	Length	Set/Get	Meaning
CTS	0	1	Set	State of the V.24 line CTS
RTS	1	1	Get	State of the V.24 line RTS
RI	1	1	Set	State of the V.24 line RI
DSR	2	1	Set	State of the V.24 line DSR
DTR	3	1	Get	State of the V.24 line DTR
DCD	4	1	Set	State of the V.24 line DCD
BRK	5	1	Set/Get	Break received/to be send
ESC	6	1	Get	Escape sequence de-tected
TXSTP	7	1	Get	Transmitter is stopped
RXSTP	8	1	Get	Receiver is stopped
BRKLEN	9-16	8	Set/Get	Length of the break signal in characters
RXBLEV	17-28	12	Get	Number of free bytes in the RX buffer
SA	29	1	Set/Get	State of SA bit
SB	30	1	Set/Get	State of SB bit
X	31	1	Set/Get	State of X bit

The three bits SA, SB and X are mapped onto the corresponding lines. They are provided to give the user of the driver a more abstract view of the status lines. The mapping depends on the mode of flow control used:

signal		mode of flow control			
		None	Rts	dtr	Xon
SA	set	CT107 = DSR	CT107 = DSR	ignore	CT107 = DSR
	get	CT108 = DTR	CT108 = DTR	ON = 0	CT108 = DTR

SB	set	CT109 = DCD	CT109 = DCD	CT109 = DCD	CT109 = DCD
	get	CT105 = RTS	ON = 0	CT105 = RTS	CT105 = RTS
X	set	Ignore	CT106 = CTS	CT107 = DSR	mapped onto XON/XOFF
	get	ON = 0	CT105 = RTS	CT108 = DTR	

The SA bit is always mapped onto DSR/DTR unless the mode of flow control is dtr. In this case, the driver ignores any setting of the SA bit and returns a 0 (ON) as SA bit.

The SB bit is always mapped onto DCD/RTS unless the mode of flow control is rts. In this case RTS is used for the flow control and the driver returns always a 0 (ON) as SB bit.

The X bit is either mapped on CTS/RTS or DSR/DTR in the case of outband flow control or on XON/XOFF in the case of inband flow control. When no flow control is used, the driver ignores any setting of the X bit and returns a 0 (ON) as X bit.

Any unused line must be held in ON condition (0) by the driver.

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	Wrong Device ID.
FD_NOT_READY	The function is called while the callback of the readOutFunc function is activated and still not terminated.
FD_INTERNAL_ERROR	Internal problem with the hardware

Description

Returns the state of the V.24 lines, the flow control state and the result of the break/escape detection process as a bit field.

3.3.14 SER_fd_SetLineState - Sets the states of the V.24 status lines

Definition:

```
T_FDRET SER_fd_SetLineState (
        ULONG       state,
        ULONG       mask);
```

Parameters:

Name	Description
state	Bit field. Only the signals which are marked with the 'set' access can be used to change the state of the signal. The state bitfield has the same format as described in SER_fdGetLineState.
Mask	Bit field with the same structure as state. Each bit in state corresponds to a bit in mask. Settable bits marked by a 1 are manipulated by the driver.

Return values:

Name	Description
FD_OK	Successful operation.
FD_NOT_SUPPORTED	Wrong Device ID.
FD_INTERNAL_ERROR	Internal problem with the hardware.

Description

Sets the states of the V.24 status lines according to the bit field of the parameter state.

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/>