

XDS Native Command Set Function Library For H.110 Boards

August 2005



American Tel-A-Systems, Inc.

258M013E ©

Printed in U.S.A. All rights reserved.

This page was intentionally left blank.

Contents

1 Common XDS Defines And Return Codes

Library Return Codes	1-3
IOCTL Return Codes	1-4
Board Variables	1-5

2 Generic H.110 Board Functions

xds_id	2-3
xds_msg_off	2-4
xds_msg_on	2-5
xds_msg_send	2-6
xds_msg_receive	2-7
xds_query_receive	2-9
xds_reset_all	2-11
xds_reset_dsp	2-12
xds_board_reset	2-13
xds_set_encoding	2-14
xds_set_type	2-15
xds_set_ct_clock_mode	2-16
xds_h110_led	2-18
xds_pci_resources	2-19
xds_remove_board	2-20
xds_insert_board	2-21
xds_set_config	2-22
xds_num_port	2-24
xds_queue_message	2-25

3 Basic Rate ISDN Common Functions

xds_bri_ctbus_cptones	3-3
xds_bri_bchannel_cptones	3-5
xds_bri_deactivate_layer1	3-6
xds_bri_loopback	3-7
xds_bri_power_feed	3-8
xds_bri_reactivate_layer1	3-10
xds_bri_reset_bchannel	3-11
xds_bri_test	3-12
xds_bru_eoc	3-13
xds_bru_mon8	3-14
xds_reset_port	3-15
xds_set_bri_dn	3-16
xds_set_layer	3-18

xds_tei_assign.....	3-20
xds_tei_check.....	3-21
xds_tei_cnct.....	3-22
xds_tei_disc.....	3-23
xds_tei_fixed.....	3-24
xds_tei_packet.....	3-25
xds_tei_query.....	3-26
xds_tei_remove.....	3-27
xds_tei_verify.....	3-28
xds_set_bri_clock.....	3-29
xds_l3_restart.....	3-30

4 Basic Rate ISDN NI-1 Layer 3 ‘D’ Functions

xds_l3_alert.....	4-3
xds_l3_connect.....	4-5
xds_l3_disconnect.....	4-6
xds_l3_feature_ind.....	4-8
xds_l3_feature_key.....	4-10
xds_l3_hold.....	4-11
xds_l3_hold_ack.....	4-12
xds_l3_hold_rej.....	4-13
xds_l3_info_request.....	4-14
xds_l3_key_hold.....	4-16
xds_l3_key_setup.....	4-17
xds_l3_notify.....	4-19
xds_l3_proceed.....	4-21
xds_l3_progress.....	4-23
xds_l3_query_spid.....	4-25
xds_l3_query_status.....	4-27
xds_l3_rel_com.....	4-28
xds_l3_retrieve.....	4-30
xds_l3_retrieve_ack.....	4-32
xds_l3_retrieve_rej.....	4-33
xds_l3_setup.....	4-35
xds_l3_spid.....	4-38
xds_l3_text.....	4-39
xds_l3_message.....	4-41

5 BRI EURO ISDN Functions

xds_euro_l3_alert.....	5-3
xds_euro_l3_call_forwarding.....	5-4
xds_euro_l3_called_num.....	5-6
xds_euro_l3_connect.....	5-8
xds_euro_l3_disconnect.....	5-10
xds_euro_l3_facility.....	5-12

xds_euro_l3_hold.....	5-14
xds_euro_l3_hold_ack.....	5-15
xds_euro_l3_hold_rej.....	5-16
xds_euro_l3_info.....	5-17
xds_euro_l3_key_fac.....	5-18
xds_euro_l3_notify.....	5-19
xds_euro_l3_proceed.....	5-20
xds_euro_l3_progress.....	5-22
xds_euro_l3_query_dn.....	5-24
xds_euro_l3_query_status.....	5-26
xds_euro_l3_rel_com.....	5-27
xds_euro_l3_resume.....	5-29
xds_euro_l3_resume_ack.....	5-30
xds_euro_l3_resume_rej.....	5-31
xds_euro_l3_retrieve.....	5-32
xds_euro_l3_retrieve_ack.....	5-33
xds_euro_l3_retrieve_rej.....	5-34
xds_euro_l3_setup.....	5-35
xds_euro_l3_suspend.....	5-38
xds_euro_l3_suspend_ack.....	5-40
xds_euro_l3_suspend_rej.....	5-41
xds_euro_l3_text.....	5-42
xds_euro_set_bri_dn.....	5-44

6 BRI INS NET – 64 Functions

xds_net64_l3_alert.....	6-3
xds_net64_l3_connect.....	6-5
xds_net64_l3_disconnect.....	6-7
xds_net64_l3_disc_ref.....	6-9
xds_net64_l3_feature_act.....	6-11
xds_net64_l3_feature_ind.....	6-12
xds_net64_l3_hold.....	6-14
xds_net64_l3_hold_ack.....	6-15
xds_net64_l3_hold_rej.....	6-16
xds_net64_l3_info.....	6-17
xds_net64_l3_info_text.....	6-19
xds_net64_l3_notify.....	6-20
xds_net64_l3_proceed.....	6-21
xds_net64_l3_progress.....	6-23
xds_net64_l3_query_status.....	6-25
xds_net64_l3_rel_com.....	6-26
xds_net64_l3_rel_com_ref.....	6-28
xds_net64_l3_resume.....	6-30
xds_net64_l3_resume_ack.....	6-31
xds_net64_l3_resume_rej.....	6-32

xds_net64_l3_retrieve.....	6-34
xds_net64_l3_retrieve_ack.....	6-35
xds_net64_l3_retrieve_rej.....	6-36
xds_net64_l3_setup.....	6-37
xds_net64_l3_suspend.....	6-40
xds_net64_l3_suspend_ack.....	6-42
xds_net64_l3_suspend_rej.....	6-43
xds_net64_l3_text.....	6-44
xds_net64_l3_user_alert.....	6-46
xds_net64_l3_user_info.....	6-48
xds_net64_l3_user_roceed.....	6-50

7 BRI AT&T Custom Board Functions

xds_att_l3_assoc.....	7-3
xds_att_l3_assoc_ack.....	7-5
xds_att_l3_call_appearance.....	7-6
xds_att_l3_conference.....	7-8
xds_att_l3_conference_ack.....	7-9
xds_att_l3_conference_rjc.....	7-10
xds_att_l3_drop.....	7-11
xds_att_l3_drop_ack.....	7-12
xds_att_l3_drop_rej.....	7-13
xds_att_l3_feature_ind.....	7-14
xds_att_l3_feature_key.....	7-16
xds_att_l3_hold.....	7-17
xds_att_l3_hold_ack.....	7-18
xds_att_l3_hold_rej.....	7-19
xds_att_l3_mim_spid.....	7-20
xds_att_l3_reconnect.....	7-21
xds_att_l3_reconnect_ack.....	7-22
xds_att_l3_reconnect_rej.....	7-23
xds_att_l3_redirect.....	7-24
xds_att_l3_switchhook.....	7-25
xds_att_l3_text.....	7-26
xds_att_l3_transfer.....	7-28
xds_att_l3_transfer_ack.....	7-29
xds_att_l3_transfer_rej.....	7-30

8 PCI-based MC-3 Board Functions

xds_mc3_ring_mode.....	8-3
xds_ct_listen.....	8-5
xds_ct_pattern.....	8-6
xds_ct_rls.....	8-7
xds_mc3_connect.....	8-8
xds_mc3_listen.....	8-10
xds_mc3_pattern.....	8-11
xds_mc3_loopback_mode.....	8-12
xds_mc3_rls.....	8-13
xds_mc3_2k_tone.....	8-14
xds_mc3_2k_tone_det.....	8-15

9 H.100 Common Line Board Functions

xds_line_ct_listen.....	9-3
xds_line_ct_connect.....	9-4
xds_line_ct_play_call_progress.....	9-5
xds_line_ct_generate_customtone.....	9-7
xds_line_ct_set_transmit.....	9-9
xds_line_ct_reset_lineport.....	9-10
xds_line_ct_set_custom_dialtone.....	9-11
xds_line_ct_set_custom_inf-tone.....	9-13
xds_line_ct_set_gain.....	9-15
xds_line_ct_set_hookflash.....	9-16
xds_em_ct_lead_control.....	9-17
xds_em_ct_quard_tone.....	9-18
xds_loop_ct_port_options.....	9-21

10 H.100 Station Board Functions

xds_station_ct_call_waiting.....	10-3
xds_station_ct_return_polarity.....	10-5
xds_station_ct_reverse_polarity.....	10-6
xds_station_ct_set_calling_name.....	10-7
xds_station_ct_set_date_calling_number.....	10-8
xds_station_ct_set_port_profile.....	10-9
xds_station_ct_set_profile_parms.....	10-11
xds_station_ct_set_visual_message_waiting.....	10-12
xds_station_ct_set_voltage.....	10-13

11 ISA / H.100 / H.110 Common Line Board Functions

xds_line_cptones.....	11-3
xds_line_detect_energy.....	11-5
xds_line_disconnect.....	11-7
xds_line_hold.....	11-8

xds_line_hookflash	11-9
xds_line_disable_output.....	11-10
xds_line_state.....	11-11
xds_line_detect_dtmf.....	11-13
xds_line_seize	11-15
xds_line_send_dial_string.....	11-16

12 Common Enhanced Conference Board Functions

xds_ct_cnf_timeslot	12-3
xds_ct_conference.....	12-4
xds_ct_conference_disable.....	12-6
xds_ct_conference_dtmf.....	12-7
xds_ct_conference_monitor.....	12-9
xds_ct_mix_remove.....	12-11
xds_ct_mix_input.....	12-12
xds_ct_conference_mode.....	12-14
xds_ct_conference_dissolve.....	12-15
xds_ct_conference_param.....	12-16
xds_ct_conference_energy_det.....	12-18
xds_ct_query_energy_table.....	12-19
xds_ct_conference_set_gain.....	12-20
xds_ct_conference_control_gain.....	12-22
xds_ct_conference_distortion_control.....	12-24
xds_ct_conference_dtmf_reporting_mode.....	12-25

13 Music On Hold Audio Port Functions

xds_audio_disable_music_rcv.....	13-3
xds_audio_disable_music_xmt.....	13-4
xds_audio_enable_music_rcv.....	13-5
xds_audio_enable_music_xmt.....	13-6
xds_audio_music_atten.....	13-7
xds_audio_music_enable.....	13-8

14 H.100 / H.110 MVIP-Compatibility Commands

xds_ct_mvip_conference_det_dtmf.....	14-3
xds_ct_mvip_conference.....	14-4
xds_ct_mvip_conference_output_disable.....	14-6
xds_ct_mvip_conference_output_enable.....	14-7
xds_ct_mvip_conference_output_pattern.....	14-8
xds_ct_mvip_line_det_dtmf.....	14-9
xds_ct_mvip_detect_energy.....	14-10
xds_ct_mvip_send_dial_string.....	14-11
xds_ct_mvip_conference_set_gain.....	14-12
xds_ct_mvip_conference_control_gain.....	14-14
xds_ct_mvip_dial_pulse_detect.....	14-16

xds_ct_mvip_tristate	14-17
----------------------------	-------

15 Received Message Types

Conference Board Errors	15-3
State Change Messages	15-3
Acknowledgments	15-5
Clock Errors	15-6
Argument Errors.....	15-7
Line Board Resource Errors	15-7
Unparseable Messages	15-7
Basic Rate ISDN Board Messages	15-7
Unknown Messages.....	15-11
Query Messages.....	15-11
Queue Errors	15-13

XDS H.110 Native Function Library Reference Manual

Author: Brian Riek

Copyright © American Tel-A-Systems, Inc., August 2005

All rights reserved.

This document and the information herein is proprietary to American Tel-A-Systems, Inc. It is provided and accepted in confidence only for use in the installation, operation, repair and maintenance of Amtelco equipment by the original owner. It also may be used for evaluation purposes if submitted with the prospect of sale of equipment.

This document is not transferable. No part of this document may be reproduced in whole or in part, by any means, including chemical, electronic, digital, xerographic, facsimile, recording, or other, without the express written permission of American Tel-A-Systems, Inc.

The following statement is in lieu of a trademark symbol with every occurrence of trademarked names: trademarked names are used in this document only in an editorial fashion, and to the benefit of the trademark owner with no intention of infringement of the trademark. "H.110" is a registered trademark of the ECTF.

American Tel-A-System, Inc.©

(608) 838-4194

4800 Curtin Drive, McFarland, Wisconsin USA 53558-9424

<http://www.amtelco.com/>

258M013E

This page was intentionally left blank.

Library & IOCTL Common Definitions

This page was intentionally left blank.

Listed in this section of the manual are commonly used function library variables and function return codes.

General Library return codes:

BRD_ERROR	-1	Board Error
ILL_PORT	1	Illegal Port
ILL_SLOT	2	Illegal Timeslot
ILL_ARG	3	Illegal Argument
ILL_HANDLE	4	Illegal Conference Handle
ILL_ATTEN	5	Illegal Attenuation
ILL_THRESHOLD	6	Illegal Threshold
WRONG_QUERY	7	Wrong Query
WRONG_BOARD	8	Wrong Board
NO_UPDATE	9	DPRAM value hasn't changed
ILL_CCA	10	Illegal Conference Handle
ILL_MODE	11	Illegal "mode" value
ILL_TYPE	12	Illegal "type" (BRI) value
ILL_FEATURE	13	Illegal "feature" (BRI) value
ILL_CAUSE	14	Illegal "cause" (BRI) value
ILL_REFERENCE	15	Illegal "reference" (BRI) value
ILL_PROGRESS	16	Illegal "progress" (BRI) value
ILL_SIGNAL	17	Illegal "signal" (BRI) value
ILL_PORT_TYPE	18	Illegal port type

Ioctl return codes:

SUCCESS 0 ioctl() call returned successfully

UNIX only defines:

XDS_MSG_AVAILABLE 0 message available
XDS_BOARD_NOT_PRESENT 1 XDS board not present
XDS_NO_MSG 1 no messages on queue
XDS_BOARD_NOT_RESPONDING 2 XDS board is not responding
XDS_DPRAM_READ_OFF 2 attempt to read at an offset before
the beginning of the board
XDS_DPRAM_BAD_WRITE 2 attempt to write to first 256
bytes of board's DPRAM
XDS_DPRAM_WRITE_LIMIT 3 attempt to write beyond the 2k limit
XDS_DPRAM_READ_LIMIT 3 attempt to read beyond the 2k limit
XDS_BAD_COMMAND 4 non-supported ioctl command

Common OS defines:

XDS_ERR_IOCTL_XMT 256 'XMT' command failed
XDS_ERR_IOCTL_RCV 257 'RCV' command failed
XDS_ERR_IOCTL_RCV_QUERY 258 'RCV_QUERY' command failed
XDS_ERR_IOCTL_WRITE_DPRAM 259 'WRITE_DPRAM' command fail
XDS_ERR_IOCTL_READ_DPRAM 260 'READ_DPRAM' command
XDS_ERR_IOCTL_INVALID_COMMAND 261 Invalid ioctl() command
XDS_ERR_IOCTL_RESET 262 'XDS_RESET' command failed
XDS_ERR_XDS_ID 263 'XDS_BOARD_ID' command failed
XDS_ERR_GET_BUS_DEVICE_NUM 264 'XDS_GET_BUS_DEVICE_NUM'
command failed
XDS_ERR_QUEUE_USER_MSG 265 'XDS_QUEUE_USER_MSG'
command failed

Commonly used variable ranges:

board_number:

ISA	-	0 – 15
H.100	-	16 – 31
H.110	-	1 – 30

port:

MVIP/SCSA BRI (ISA)	-	0 – 11
H.100 BRI	-	0 – 15
H.110 BRI	-	0 – 31
U-interface BRI (All)	-	0 – 3
4 Port BRI (All)	-	0 – 3

b_channel:

MVIP/SCSA (ISA)	-	0 – 23
H.100	-	0 – 31
H.110	-	0 – 63
U-interface (All)	-	0 – 7
4 Port (All)	-	0 – 7
T1 PRI (8 port)	-	0 – 247
T1 PRI (4 port)	-	0 – 119

timeslot:

MVIP	-	0 – 511
SCSA	-	0 – 1023
H.100 bus (MVIP mode)	-	0 – 511
H.100 bus (SCSA mode)	-	0 – 1023
H.100 bus	-	0 – 4095
H.110 bus	-	0 – 4095
MC3 bus	-	0 – 4845

This page was intentionally left blank.

Generic XDS Functions

This page was intentionally left blank.

xds_id

int xds_id(board_number, idstruct);

unsigned char board_number;	a value indicating which board the command is for
struct xdsid *idstruct;	a structure to contain the return information
{	
char board_number;	the board number
char id[5];	an array containing the board ID string
char version[5];	an array containing the firmware version number
int number_ports;	the number of ports on the board
}	

Applicable Boards

All XDS boards

Purpose

This function will return the two, three, or four character ID code (depending on the type of board), the firmware version number, and the number of ports on the board. This information is returned in a structure of the type **xdsid**.

Message Sent

None.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function can be used to obtain information about the configuration of the system, what type of XDS board is associated with each device number, and the number of ports available on each board. It functions by directly reading the Dual-Ported Ram. For the BRI boards, the port number is the number of physical interfaces.

Example

xds_id(16, &idstruct);	returns with the ID information for board 16 in idstruct; if board number 16 was an XDS H.100 MC3 board the structure would contain:
idstruct.board_number = 16;	board number
idstruct.id = "PO3";	board type "PO3" indicates that it is an H.100 MC3 board
idstruct.version = "012e";	firmware version 1.2e
idstruct.number_ports = 0;	no physical or virtual ports setup on the board

xds_msg_off

int xds_msg_off(board_number);
unsigned char board_number; a value indicating which board the command is for

Applicable Boards

All XDS boards

Purpose

This function is used to disable messages from the XDS board. It should be used at the end of an application unless multiple applications are using the same boards.

Message Sent

“IF”

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

If this function is not called at the termination of an application, the XDS board will continue to send messages in response to state changes or error detection. This will cause interrupts and may lead to problems in communicating with the board. Therefore, before closing an application, this function should be called for each XDS board. However, if more than one application is running, only the last application to close should call this function as it prevents messages from being sent by the board. No response message will be sent from the XDS board.

Example

xds_msg_off(1); sends an IF message to board 1 to disable messages

xds_msg_on

int xds_msg_on(board_number);
unsigned char board_number; a value indicating which board the command is for

Applicable Boards

All XDS boards

Purpose

This function is used to enable messages from the XDS board. It should be used at the beginning of an application.

Message Sent

“IN”

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

If this function is not called, the XDS board will not send messages. Therefore, it should be part of the initialization of an application. It will cause the board to respond with an “IA” message.

Example

xds_msg_on(1); sends an IN message to board 1 to enable messages

xds_msg_send

int xds_msg_send(board_number, message);

unsigned char board_number;

a value indicating which board the command is for

char *message;

a pointer to a NULL terminated ASCII string which is the message to be sent

Applicable Boards

All XDS boards

Purpose

This function allows an application to send a message (command string) directly to an XDS board bypassing the library functions.

Message Sent

Determined by the application (user)

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL

an IOCTL was called and returns a return value from the IOCTL call

Comments

The details of the board message sets can be found in the relevant manual for that board.

Example

xds_msg_send(1, "CD00");

this will send the message "CD00" to board 1

xds_msg_receive

int xds_msg_receive(rcv);

```
struct rcvmsg *rcv {  
  unsigned char board_number;      a value indicating which board the command is for  
  char msg_type;                  a value denoting the message type  
  char msg_sub;                   a value denoting the message subtype  
  int port_number;               the port number, if any, involved in the particular  
                                  message. if not port specific, this value should be -1  
  unsigned char argument;        a generalized argument field for the message whose  
                                  meaning is specific to the particular message type, it  
                                  will not have meaning for all message types  
  char msg_str[32];              a character array to pass any message specific  
                                  values that do not fit in the other members of the  
                                  structure  
  unsigned short augRxBLen;       length of Layer 3 message in the auxiliary mailbox  
  unsigned char augRxBMsg[260];  body of Layer 3 message in the auxiliary mailbox  
}
```

Purpose

This function returns a structure with information about messages sent from an XDS board. A single receive queue is used for all XDS boards sharing an interrupt. The returned information includes the board number, the type of message, the subtype of the message, arguments for the port, timeslot, and other information contained in the message, and a character array that may contain any additional information. Query and version responses are received on a separate query queue and may be retrieved using the **xds_query_receive** function. Messages sent by the board may indicate a change in state of a port, an acknowledgement, or some sort of error condition. The message type and subtype can be used to decode this information

Message Sent

None

Returns

If a message is contained in the structure, a 0 will return. If no message was available on the receive queue, a 1 will be returned.

Comments

Message strings starting with ‘V’ or ‘Q’, or which have a ‘Q’ as the second character and do not have an ‘E’ or an ‘S’ for the first character are version request or query request responses and are returned on the query queue and accessed through the **xds_query_receive** function. All other messages are returned on the receive queue and accessed through this function.

On the BRI boards, there is an auxiliary mailbox. This mailbox is used to pass Layer 3 messages. The presence of valid data in this mailbox is indicated by a message in the mailbox that begins with “LC” or “LR”.

Example

`xds_msg_receive(&rcv);` returns the message in the structure **rcv**

if the first message on the queue was the DID number detected message “SD051234” indicating that the DID address digits 1234 have been received on port 05, and the message was from board 2 the receive message structure would contain the following values:

<code>rcv.board_number = 2;</code>	board number 2
<code>rcv.msg_type = 1;</code>	a state change message
<code>rcv.msg_sub = 3;</code>	a DID detect message subtype
<code>rcv.port_number = 5;</code>	port 5 on the board
<code>rcv.argument = 0;</code>	no argument for this message subtype
<code>rcv.msg_str[] = “1234”</code>	the DID number detected
<code>rcv.augRxBLen</code>	not valid data
<code>rcv.augRxMesg</code>	not valid data

xds_query_receive

int xds_query_receive(query);

struct rcvmsg *query {	
unsigned char board_number;	a value indicating which board the command is for
char msg_type;	a value denoting the message type
char msg_sub;	a value denoting the message subtype
int port_number;	the port number, if any, involved in the particular message. if not port specific, this value should be (-1)
unsigned char argument;	a generalized argument field for the message whose meaning is specific to the particular message type, it will not have meaning for all message types
char msg_str[32];	a character array to pass any message specific values that do not fit in the other members of the structure
unsigned short augRxBLen;	length of Layer 3 message in the auxiliary mailbox
unsigned char augRxMesg[260];	body of Layer 3 message in the auxiliary mailbox
}	

Purpose

This function returns a structure with information about query or version request response messages sent from XDS boards. A single query queue is used for all XDS boards sharing an interrupt. The returned information includes the board number, the type of message, the subtype of the message, arguments for the port, timeslot, and other information contained in the message, and a character array that may contain any additional information. Query responses are used for diagnostic purposes. Version responses are used to obtain firmware version information. All query and version request responses are message type 16. The message subtype can be used to decode the message.

Message Sent

None

Returns

If a message is contained in the structure, a 0 will return. If no message was available on the query queue, a 1 will be returned.

Comments

Query response messages are hardware specific and require knowledge of the design of a particular board and the IC's used to implement the design. Version request information may be used to get the board ID and additional information about the board type, firmware version, and number of ports. However, as this requires sending messages in both directions, it is more efficient to use the **xds_id** function to get this information.

Example

`xds_query_receive(&query);` returns the message in the structure **query**

if the first message on the query queue was the query clock mode response message "QC1" and the message was from board 2 the query message structure would contain the following values

<code>query.board_number = 2;</code>	board number 2
<code>query.msg_type = 16;</code>	a query response message
<code>query.msg_sub = 4;</code>	a clock mode query response message subtype
<code>query.port_number = -1;</code>	no port involved
<code>query.argument = 0x31;</code>	clock mode 1, encoded as ASCII
<code>query.msg_str[] = 0</code>	a NULL string
<code>query.augRxLen</code>	not valid data
<code>query.augRxMesg</code>	not valid data

xds_reset_all

int xds_reset_all(board_number);
unsigned char board_number; a value indicating which board the command is for

Applicable Boards

All XDS boards

Purpose

This function can be used to put a board in a known, initialized state. All ports are released, all connections are broken, and all resources are freed. The CT bus is tri-stated with no output. This command does not change the clock mode of the board.

Message Sent

“RA”

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function should be used for all XDS boards when starting an application to put the boards in a known state.

Example

xds_reset_all(1); resets the XDS board number 1

xds_reset_dsp

int xds_reset_dsp(board_number);
unsigned char board_number; a value indicating which board the command is for

Applicable Boards

XDS boards with reset-able DSPs

Purpose

This function should be used for maintenance on XDS boards that have reset able DSPs.

Message Sent

“RD”

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function should be used for maintenance on XDS boards that have reset-able DSPs. Consult the hardware reference manual to check for this option.

Example

xds_reset_dsp(1); resets the XDS board number 1

xds_board_reset

int xds_board_reset(board_number);

unsigned char board_number;

a value indicating which board the command is for

Applicable Boards

All Infinity Series H.100/H.110 Boards

Purpose

This function will reset the entire board without having to have the user shut down the system.

Message Sent

“IF”, then the hardware reset

Returns

This function will return a 0 if successful. If it is not successful, it will return a 1.

Comments

This function could be useful if a board becomes congested. After the board is active again, the user will need to resend a “IN” to re-enable board interrupts (if they choose).

Example

xds_board_reset(1);

resets board number 1

xds_set_encoding

int xds_set_encoding(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
char mode; a character indicating the encoding mode, 'A' for A-Law,
 'M' for Mu-law

Applicable Boards

All XDS boards

Purpose

This function is used to set the encoding mode for an XDS board. Two standards apply, A-Law which is used in Europe and Asia, and Mu-Law which is used in North America and Japan.

Message Sent

“SEm” where **m** is the mode.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE an invalid value for the encoding was chosen

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

There are two standards for the conversion between analog and digital and digital and analog signals used in telephony. These are A-Law, which is used in Europe and Asia, and Mu-Law, which is used in North America and Japan. These two standards are incompatible. Because of this, all boards in a system must use the same standard. If any outside digital lines such as T1, E1, or ISDN are used, the standard used must be compatible with that used by the digital line. Boards that only pass through digital signals such as the MVIP/SCSA Bridge Board do not require this function. However, any board, even if all digital that processes signals, such as with conferencing, must be set to the appropriate encoding standard. The default standard for XDS boards is Mu-Law.

Example

xds_set_encoding(1, 'M'); sets the encoding to Mu-Law board

xds_set_type

int xds_set_type(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
char *types; a pointer to a character array indicating port types

Applicable Boards

XDS boards with physical ports (interfaces)

Purpose

This function will set the behavior types of each port on the board. The allowed types of behavior will depend on the specific board types. As an example, setting a port to type 'P' or phone on a station board will allow that port to be rung. Setting the port type is important for the port to exhibit the correct behavior.

Message Sent

“ST(p * ports)” where **p** is the port type selected line boards and all non-H.110 boards.

“STL(p * ports)” where **p** is the mode – lower 16 ports (H.110).

“STH(p * ports)” where **p** is the mode – upper 16 ports (H.110).

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_ARG	an empty string of port types was passed to the function
ILL_TYPE	an invalid port type was selected for the board
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

As there are many port types for the different boards, you will need to consult the board reference manual to determine the valid types.

Example

xds_set_type(1, “NNUU”);	sets ports 0 and 1 to “NT” and ports 3 and 4 to “Undefined” (BRI 4-port board)
--------------------------	--

xds_set_ct_clock_mode

xds_set_ct_clock_mode(board_number, mode, submode, arg1, arg2, arg3);

unsigned char board_number;	a value indicating which board the command is for
char mode;	a value between 0 and 5 selecting the clock mode
char submode;	a value between 0 and 5 selecting the clock submode
char arg1;	a value between 0 and 3 specifying a clock mode argument
char arg2;	a value between 0 and 5 specifying a clock mode argument
char arg3;	a value between 0 and 3 specifying a clock mode argument

Applicable boards

All Infinity Series boards

Purpose

This function is used to set the clock mode for XDS H.100 and H.110 boards. Because the clocks for the MC3 bus and H.100 bus are related, this function may affect clocks on both busses.

Message Sent

“Scmsabb(c)” where **m** is the clock mode, **s** is the submode, **a** is arg1, **bb** is arg2, and **c** is agr3.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

SUCCESS	if successful
ILL_ARG	an invalid clock mode or sub-mode was passed
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

In a system of PCs tied together using the MC3 boards, there must be only one source of the master clock. This clock may be generated by any board in the system and may be created by a local oscillator on that board or derived from an outside source such as a T1 span. All other boards in the system must be slaved to this clock. The MC3 Board in the same chassis with the master clock must derive the MC3 bus clock from the H.100/H.110 bus and generate the clocks for the MC3 bus. Other MC3 Boards in the system must derive their clocks from the MC3 bus and supply the clocks to the H.100/H.110 bus in their chassis. It is possible for a MC3 Board to supply all the clocks to both buses from an on board oscillator.

Because all boards must be set to compatible clock modes, this function must be used with care. A more detailed discussion of the clocks available to each board may be found in any of the Infinity Series board technical manuals.

Example

```
xds_set_ct_clock_mode(16, 2, 3, 0, 2);
```

this will set board 16 to derive the clock from the Ring 0 of the MC3 bus and provide the clock to the H.100/H.110 bus.

xds_h110_led

int xds_h110_led(board_number, led_mode);
unsigned char board_number; a value indicating which board the command is for
int led_mode; the L.E.D. mode to use

Applicable Boards

XDS H.110 BRI boards

Purpose

This function will control the AUX LED on the front panel of the XDS BRI board.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

SUCCESS if successful
ILL_ARG an empty (NULL) or invalid port type string was passed
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Message Sent

“SIX” where **x** is the LED mode to use.

Comments

The user should check the hardware reference manual to see whether or not this command is supported by the board’s firmware.

Mode	Message Sent	LED Behavior
0	“SIX”	normal mode
1	“SI0”	LED off
2	“SI2”	LED to orange
3	“SI4”	LED to flashing orange
4	“SI1”	LED to green
5	“SI3”	LED to flashing green

Example

xds_h110_led(1, 5); sets the AUX LED to flash green

xds_pci_resources

```
int xds_pci_resources(board_number, info);
unsigned char board_number;          a value indicating which board the command is for
struct xdsid *info;                 a structure to contain the return information
{
  unsigned char board_number;        a value indicating which board the command is for
  char id[5];                       an array containing the board ID string
  char version[5];                  an array containing the firmware version number
  int number_ports;                 the number of ports on the board
  unsigned char pci_bus_number;      the PCI bus number for the board
  unsigned char pci_slot_number;     the PCI slot number for the board
}
```

Applicable Boards

All Infinity Series boards

Purpose

This function will return the PCI bus and slot number for a given XDS PDI/cPCI board.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

SUCCESS if successful
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Message Sent

None

Comments

The PCI resource for each XDS board is assigned by the operating system.

Example

```
xds_pci_resources(1, &info);       returns the PCI bus and slot number for board 1 in info

info.board_number = 1;             board number is 1
info.pci_bus_number = 2;           PCI bus number for board 1 is 2
info.pci_slot_number = 7;          PCI slot (function) number for board 1 is 7
```

xds_remove_board

int xds_remove_board(board_number);

unsigned char board_number; a value indicating which board the command is for

Applicable Boards

All XDS cPCI H.110 boards

Purpose

This function is used to prepare the H.110 board for removal from the system.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

SUCCESS if successful

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Message Sent

“IF” to the board before removal and an “HR” to the application level.

Comments

This command is intended for cPCI H.110 XDS boards only. This will disable the board interrupts and PLX chip before activating the blue hot-swap LED on the board. Once the blue LED is on, the user may physically remove the board from the system.

If the XDS interface is a two-board set, like the BRI S/T board, the user will remove the front board first, and the rear I/O board last. The user will then replace the rear I/O board first, and the front board last.

Example

xds_remove_board(10);

removes board 10

xds_insert_board

```
int xds_insert_board(board_number);  
unsigned char board_number;          a value indicating which board the command is for
```

Applicable Boards

All XDS cPCI H.110 boards

Purpose

This function is to prepare the H.110 board for insertion back into a system.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

SUCCESS if successful
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Message Sent

“HA” to the application level.

Comments

This command is intended for cPCI H.110 XDS boards only. When administering a board, the user must replace the board removed with another board of the same type (ie: MC3 board must be replaced with another MC3 board) into the same slot in which the original board was removed from.

If the XDS interface is a two-board set, like the BRI S/T board, the user will remove the front board first, and the rear I/O board last. The user will then replace the rear I/O board first, and the front board last.

The user may not insert a new board into a different (new) slot while the system is running. The boards must be present at boot-up for the system to receive the board resources from the BIOS.

Example

```
xds_insert_board(10);                inserts board 10
```

xds_set_config

xds_set_config(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
char mode; a character indicating the mode of the configuration
 command, allowed values are 'C', 'L', and 'S'

Applicable boards

All XDS BRI boards

Purpose

This function is used to clear, load, or save data in the EEAROM. This information consists of port type, protocol layer, directory numbers, and SPIDs. If this information is saved in the EEAROM, it is automatically loaded when the board reboots, eliminating the need for the application to download this configuration information.

Message Sent

“SMm” where **m** is the mode, valid modes are:

- 'C' - clear EEAROM data
- 'L' - load EEAROM data into RAM memory
- 'S' - save configuration information in RAM in the EEAROM

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE invalid configuration mode
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function is used to save the current configuration information held in RAM in the on board EEAROM. This information will then be loaded into RAM every time the board reboots. This saves the application from having to download the configuration information every time the board reboots. It is only necessary to save information when the configuration of the board changes. If a number of changes are made at a time, it is only necessary to save the information after the last change has been made.

The information saved consists of the port types, the support layer selected, the clock mode, and the directory numbers and SPIDs associated with each BRI interface.

Example

`xds_set_config(0, 'S');` this will cause board 0 to save configuration information

xds_num_port

int xds_num_port(board_number);

unsigned char board_number; a value indicating which board the command is for

Applicable Boards

All XDS boards

Purpose

This function is used to query the number of ports (interfaces) on an XDS board

Message Sent

NONE

Returns

This function will return the number of ports that are present.

Comments

PCI-based Multi-chassis boards, such will return a (0) for the number of ports, unless the user assigns the number of ports, using the “SPst” command.

Example

xds_num_port(1);

returns the number of ports for XDS board
number 1

xds_queue_message

xds_queue_message (board_number, message);

unsigned char board_number; a value indicating which board the command is for
char *message; a character string indicating the message string to be
 queued

Applicable boards

All XDS boards

Purpose

This function is used to place an XDS message string on the message receive queue for a requested board.

Message Sent

None

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Example

xds_queue_message (16, "SH00"); this will place the "SH00" message on the receive
 queue

This page was intentionally left blank.

Basic Rate ISDN Common Functions

This page was intentionally left blank.

xds_bri_ctbus_cptones

int xds_bri_ctbus_cptones(board_number, tone, stream, timeslot);

unsigned char board_number; a value indicating which board the command is for
int tone; a value between 0x20 and 0x25 or 0x40 and 0x45,
 depending on type of board, indicating which call progress
 tone to play
int stream; a value between 0x0 and 0xF (ISA), or 0x0 and 0x1F
 (H.100 & H.110) indicating the stream the tone should be
 output on
int timeslot; a value between 0x0 and 0x3F (ISA), or 0x0 and 0x7F
 (H.100 & H.110) indicating the timeslot on that stream

Applicable boards

All XDS BRI boards except for the MVIP (ISA) version.

Purpose

This function transmits a call progress tone to the CT bus. Certain Layer 3 commands may indicate that a call progress tone should be sent to the listener. This command allows this if the board to which the listener is connected does not have call progress facilities.

Message Sent

“CSxxsst” where **xx** is the tone, **ss** is the stream, and **tt** is the timeslot.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG an out of range call progress tone was selected
WRONG_BOARD an MVIP BRI board was selected (not supported)
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The purpose of this command is to allow for the playing of a call progress tone onto the CT bus when a received Layer 3 message indicates that this is appropriate with a signal information element. As not all MVIP ISA boards are capable of generating call progress tones, provisions were made for this facility on the other forms of the BRI board. The stream and timeslot value are the actual stream and timeslot on the CT bus.

The six tones produced by this function are selected with the tone argument. The tones produced are:

SCSA ISA and H.100 PCI board-

0x20 - dial tone

0x21 - reorder/ congestion tone

0x22 - busy tone

0x23 - audible ringback

0x24 - digital milliWatt, a 1004 Hz. steady tone

0x25 - silence

H.110 cPCI board-

0x40 - dial tone

0x41 - reorder/ congestion tone

0x42 - busy tone

0x43 - audible ringback

0x44 - digital milliWatt, a 1004 Hz. steady tone

0x45 - silence

The first four tones are as defined in RS-464.

Example

`xds_bri_ctbus_cptones(16, 0x21, 2, 3);` this will play “reorder tone” on the CTbus stream 2, timeslot 3 timeslot of (H.100) board 16

xds_bri_bchannel_cptones

int xds_bri_bchannel_cptones(board_number, tone);

unsigned char board_number; a value indicating which board the command is for
int tone; a value between 0x20 and 0x25 or 0x40 and 0x45,
 depending on type of board, indicating which call progress
 tone to play

Applicable boards

All XDS BRI boards

Purpose

This function gives a call progress tone to a specified B-channel.

Message Sent

“CP xy ” where xx is the B-channel and y is the tone.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range
ILL_ARG an out of range call progress tone was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The purpose of this command is to allow for the playing of a call progress tone onto the B-channel when a received Layer 3 message indicates that this is appropriate with a signal information element.

The six tones available to this function are selected with the tone argument. They are:

- 0 - dial tone
- 1 - reorder/ congestion tone
- 2 - busy tone
- 3 - audible ringback
- 4 - digital milliWatt, a 1004 Hz. steady tone
- 5 - silence

Example

xds_bri_bchannel_cptones(16, 2, 1); this will play “reorder tone” on B-channel 2, board
16

xds_bri_deactivate_layer1

xds_bri_deactivate_layer1(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface

Applicable boards

All XDS BRI boards, except the ISA S/T boards

Purpose

This function is used to deactivate Layer 1 on an NT or LT type port.

Message Sent

“RIpp” where **pp** is the port number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function will cause Layer 1 to be deactivated on an S/T port set to type NT or a U-Interface port set to type LT. With Layer 1 deactivated, no communication including the B-channels is possible until Layer 1 is reactivated. This function is present in S/T boards with firmware version 0.7 or later or EuroISDN firmware and U-interface boards with firmware version 0.0h or later.

Example

xds_bri_deactivate_layer1(16, 0x4); this will deactivate port 4 on board 16

xds_bri_loopback

int xds_bri_loopback(board_number, port, mode);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int mode; a value between 0 and 3 indicating the loopback mode

Applicable boards

All XDS BRI boards

Purpose

This function is used to set the loopback mode on the BRI interface specified with the port argument. Setting the loopback mode can be used for diagnostic purposes.

Message Sent

“XLxxm” where xx is the BRI interface number and l is the loopback mode

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range
ILL_MODE the mode specified is outside the range 0x0 to 0x3
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to place a BRI interface into loopback at the S/T interface. The possible modes are:

- 0 - no loopback
- 1 - loopback on channel B1
- 2 - loopback on channel B2
- 3 - loopback on both channels

This command should only be used for diagnostic purposes. The interface will not operate properly if left in a loopback condition.

Example

xds_bri_loopback(1, 2, 3); this will loopback both channels on BRI interface 2 of board 1.

xds_bri_power_feed

xds_bri_power_feed(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
char *mode; a pointer to a character array indicating the power feed
 mode for each BRI interface.

Applicable boards

XDS H.100 & H.110 S/T and 4 port BRI boards

Purpose

This function is used to control the power feed circuit for each NT port.

Message Sent

“SF(m * 8)” where **m** is the power feed mode information – 4 port.

“SF(m * 16)” where **m** is the power feed mode information, H.100.

“SFL(m * 16)” where **m** is the power feed mode information, lower bank – H.110.

“SFH(m * 16)” where **m** is the power feed mode information, upper bank – H.110.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE invalid power feed mode selected

ILL_ARG the power feed mode parameter was NULL

WRONG_BOARD an non-BRI board was used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

SUCCESS the function returned successfully

Comments

When set to type NT, an interface may supply power to the attached terminal. This function controls the power feed circuit. Ports that are set to type TE should have the power feed mode set to 'F'.

The mode parameter is a string of characters that is used to control the battery feed mode for each port. The valid parameters for the mode string are a 'N', 'F', or '*' for each port. The 'N' mode will turn the power feed on for a given port, 'F' will turn the power feed off for a given port, and '*' will leave the mode unchanged. A value must be entered for each of the ports.

On the 4 port BRI board, the first four characters control the PS1 power for the ports and the last four characters control the PS2 power for the ports. On the H.100 BRI S/T board, the sixteen characters control the PS1 power for the ports. On the H.100 BRI S/T board, the sixteen characters control the PS1 power for the ports.

Example

4 port-

```
xds_bri_power_feed(16, "NNNNFFFF");
```

this will turn the battery feed on for all for of the PS1 ports and turn the battery feed off for all four of the PS2 ports on board 16

H.100-

```
xds_bri_power_feed(17, "NNNNNN**FFFFFF**");
```

this will turn the battery feed on for ports 0 - 5, leave ports 6 and 7 unchanged, turn battery feed off on ports 8 - 13, and leave ports 14 and 15 unchanged on PS1 of board 17

H.110-

```
xds_bri_power_feed(5, "NNNNNNNNNNNNNNNNNN");
```

this will turn the power feed on for the lower bank of 16 ports on board 5

```
xds_bri_power_feed(5, "FFFFFFFFFFFFFFFF");
```

this will turn the power feed off for the upper bank of 16 ports on board 5

xds_bri_reactivate_layer1

xds_bri_reactivate_layer1(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface

Applicable boards

All XDS BRI boards, except the ISA S/T boards

Purpose

This function is used to reactivate Layer 1 on an NT or LT type port.

Message Sent

“RLpp” where **pp** is the port number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function will cause the board to attempt to reactivate Layer 1 on a port. If Layer 1 reactivation is successful, Layers 2 and 3 will be reestablished. This function is present in S/T boards with firmware version 0.7 or later or EuroISDN firmware and U-interface boards with firmware version 0.0h or later.

Example

xds_bri_reactivate_layer1(16, 0x4); this will reactivate layer 1 on port 4 of board 16

xds_bri_reset_bchannel

xds_bri_reset_bchannel(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the Layer 3 B-channel interface

Applicable boards

All XDS BRI boards, except the ISA S/T boards

Purpose

This function is used to reset a B-channel at Layer 3. This involves clearing the call state and call reference for the selected B channel.

Message Sent

“RBxx” where xx is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function resets Layer 3 on the selected B-channel. The call state for that B-channel is set to idle (U0), and the associated call reference is released. This function is present in S/T boards with firmware version 0.7 or later or EuroISDN firmware and U-interface boards with firmware version 0.0h or later. This command does not disable connections to the SCbus.

Example

xds_bri_reset_bchannel(16, 0x4); this will reset B-channel 4 on board 16

xds_bri_test

int xds_bri_test(board_number, port, mode);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int mode; a value between 0 and 2 indicating the test mode

Applicable boards

All XDS BRI boards

Purpose

This function is used to place the BRI interface specified with the port argument into a test mode for diagnostic purposes.

Message Sent

“XTxxm” where **xx** is the BRI interface number and **m** is the test mode.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range
ILL_MODE the test mode is invalid
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to place a BRI interface into one of two test modes. In these modes, the interface will output a square wave at either 2 kHz (mode 1) or 96 kHz (mode 2). Mode 0 clears the test signal. This command should only be issued for test purposes. The interface will not operate when in the test mode.

Example

xds_bri_test(1, 2, 2); this will place BRI interface 2 of board 1 in test mode 2.

xds_bru_eoc

int xds_bru_eoc(board_number, port, mode)

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int mode; a value between 0 and 6 indicating the EOC command

Applicable boards

XDS U-Interface BRI Boards

Purpose

This command is used to send “Embedded Operation Channel” messages from a “Line Termination” (LT) port for maintenance functions.

Message Sent

“XExxm” where **xx** is the BRI Interface number and **m** is the EOC command mode

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range
ILL_MODE the mode is invalid
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The Embedded Operations Channel is used to send commands from an Line Termination to a Network Termination to put the NT in loopback or other test modes. The EOC command modes are:

- 0 - RTN, return to normal
- 1 - LB1, loopback of channel B1
- 2 - LB2, loopback of channel B2
- 3 - LBB, loopback of both B channels and the D channel
- 4 - RRC, request corrupt CRC
- 5 - NCC, notify of corrupt CRC
- 6 - H, Hold

Example

xds_bru_eoc(1, 2, 3) will send the LBB EOC command on port 2 of board 1

xds_bru_mon8

int xds_bru_mon8(board_number, port, mode)

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int mode; a value between 0 and 7 indicating the MON-8 command

Applicable boards

XDS U-Interface BRI Boards

Purpose

This command is used to send local command messages to the U-Interface hardware for maintenance functions.

Message Sent

“XLxxm” where **xx** is the BRI Interface number and **m** is the “MON-8” command mode

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range
ILL_MODE the mode is invalid
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The “MON-8” commands are used to place the U-Interface hardware in maintenance or test modes. The “MON-8” command modes are:

- 0 - PACE, Partial Activation Control External
- 1 - PACA, Partial Activation Control Automatic
- 2 - CCRC, Corrupt CRC (LT only)
- 3 - LBBDD, loopback of both B channels and the D channel (NT only)
- 4 - NORM, return to normal
- 5 - RBEN, read Near-End Block Error Counter
- 6 - RBEF, Read Far-End Block Error Counter
- 7 - SFB, set FEBE bit to 0

Example

xds_bru_mon8(1, 2, 3) will send the “LBBDD MON-8” command to port 2 of board 1

xds_reset_port

xds_reset_port(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface

Applicable boards

All XDS BRI boards

Purpose

This function is used to reset a port. This involves a reset of the Layer 1 and Layer 2 protocols on the port.

Message Sent

“RPPp” where **pp** is the port number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function can be used to force a reset of the Layer 1 and Layer 2 protocols on a port. All data links for that port will be disconnected and the terminal endpoint identifiers (TEIs) associated with the port will be removed. After reestablishing Layer 1, Layer 2 procedures will be followed to assign new TEIs. This command does not disable connections to the CT bus.

The board will respond with a Port Reset Acknowledge message.

Example

xds_reset_port(1, 0x4);

this will reset port 4 on board 1

xds_set_bri_dn

xds_set_bri_dn(board_number, b_channel, dn, spid);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	the number of the b-channel that indicates the BRI interface
char *dn;	a pointer to a character array containing a seven digit ASCII string with the directory number
char *spid;	a pointer to a character array containing an ASCII string with the SPID (TE interfaces only) the SPID may be up to 14 digits

Applicable boards

All XDS BRI boards

Purpose

This function is used to set the default DN or directory number associated with a particular B-channel. If the interface is defined as terminal equipment or TE on an S/T Interface Board or as a Network Termination or NT on a U-Interface Board, this function is also used to set the Service Profile ID or SPID. This function is only relevant when layer 3 support is selected.

Message Sent

“SDddddddd” where **ddddddd** is the directory number for the B-channel, or
“SDddddddd/sssssssssss” for a TE interface where **sssssssssss** is the SPID.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG	invalid length for the directory number
ILL_PORT	the port is out of valid range
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

If layer 3 support is chosen using the **xds_set_layer**, it is necessary to define a default directory number for each B-channel. On a port defined as a network termination or NT, the number is used as the called party number in SETUP messages. If the port is defined as a TE, the number is the calling party in SETUP messages. It is possible for both B-channels on an interface to have the same directory number.

For a port defined as a TE, it is also necessary to define one or two SPID's. If only one SPID is required, it is defined for the even B-channel of that port. No SPID is defined for the odd B-channel. If two SPID's are required, one SPID is defined for each B-channel. The data-link associated with the SPID set for the even B-channel will be used for speech and audio calls while the data-link associated with the SPID set for the odd B-channel will be used for data calls. Note that this does not restrict speech calls to the even B-channel and data calls to the odd B-channel. The directory number and SPID can be save in EEAROM using the **xds_set_config** function. If this is done, this and other information will automatically be restored on a reboot of the board. Therefore, this function will only be needed if the configuration changes.

Example

```
xds_set_bri_dn(1, 0x4, "5551000", '');
```

this will set the default DN for B-channel 4 to 555-1000

```
xds_set_bri_dn(1, 0x6, "5552000", "608555200001")
```

this will set the default DN for B-channel 6 to 555-2000 and the SPID to 608555200001.

xds_set_layer

xds_set_layer(board_number, layer);

unsigned char board_number; a value indicating which board the command is for
char *layer; a pointer to a character array indicating the protocol layer
for each BRI interface, the array consists of a character for
each interface on the board. Valid characters are '2', '3',
'A', 'D', 'E', or 'N'.

Applicable boards

All XDS BRI boards

Purpose

This function is used to set the protocol layer for each of the BRI interfaces. The board can provide support at layer 2, layer 3, AT&T Custom, DMS-100, CACH EKTS, or National ISDN 1.

Message Sent

“SL(p*number of ports)” (ISA / H.100 boards) where p is the protocol layer

“SLL(p*number of ports)” (H.110 boards) where p is the protocol layer, lower bank of 16

“SLH(p*number of ports)” (H.110 boards) where p is the protocol layer, upper bank of 16

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG invalid layer type

WRONG_BOARD a non-BRI board was selected

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The board will provide support at either layer 2 or layer 3. If layer 2 support is chosen, the board will handle all layer 2 functions, but will require the application to compose and interpret the raw Q.931 messages for call control. These messages are sent using the **xds_l3_message()** function. If layer 3 support is chosen, call control communications between the driver and the board is accomplished through 'D' messages. This is a simpler application interface and does not require the program to have as detailed a knowledge of the Q.931 message structure.

Layer types are as follows:

- 2 - basic Layer 2 support
- 3 - basic Layer 3 support
- A - AT&T Custom support
- D - DMS-100 or 5E National ISDN-1
- E- CACH EKTS support
- N - National ISDN-1 (Siemens EWSD switch)

The default support is layer 2. The protocol layer supported can be saved in EEAROM using the **xds_set_config** function. If this is done, the support layer and other information will automatically be restored on a board reboot. Therefore, this function will only be needed if the configuration changes.

Example

`xds_set_layer(1, "333333222222");` this will set board 1 to support layer 3 on the first six interfaces and layer 2 on the last six.

xds_tei_assign

int xds_tei_assign(board_number,port);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface

Applicable boards

All XDS BRI boards

Purpose

This function is used to request the assignment of a Terminal Endpoint Identifier (TEI) to the BRI interface specified with the port argument. The port must be defined as type TE, terminal equipment. At least one TEI is needed for sending Layer 3 call control messages.

Message Sent

“TAXx” where **xx** is the BRI interface number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to request the TEI portion of the DLCI (Data Link Connection Identifier) address. This address is used to specify which data link a Layer 3 message is for in the LAPD protocol. Only TE (terminal equipment) on an S/T Board or NT (network termination) on a U-Interface Board ports can request a TEI. The TEI will be granted by the NT (network termination) or LT (line termination) end of the ISDN circuit. The XDS MVIP BRI Boards can handle the assignment of up to seven TEIs for each BRI interface.

When a TEI has been assigned, the TEI will be reported in a TEI assignment message of type 9 sub_type 4.

Example

xds_tei_assign(1, 2); this will request a TEI on BRI interface 2 of board 1.

xds_tei_check

int xds_tei_check(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface

Applicable boards

All XDS BRI boards

Purpose

This function is used to request that a Terminal Endpoint Identifier (TEI) check procedure be carried out on the BRI interface specified with the port argument. The port must be defined as type NT, network termination for an S/T Board or as type LT for a U-Interface Board. A TEI check procedure is used to verify that no conflict exists in TEI assignment.

Message Sent

“TCxx” where xx is the BRI interface number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to request that a TEI check procedure be carried out. This procedure is used to verify that no TEI is assigned to more than one TE terminal equipment, and that a TE exists for each TEI assigned. If there is a TEI assignment error, this procedure will detect and correct it. Only an NT, network termination, can initiate a TEI check procedure.

Example

xds_tei_check(1, 2); this will request a TEI check on BRI interface 2 of board 1.

xds_tei_cnct

int xds_tei_cnct(board_number, port, tei);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int tei; a value between 0x0 and 0x7E indicating the TEI of the
 data link to be established, a value of 0x7F indicates the
 packet data link

Applicable boards

All XDS BRI boards

Purpose

This function is used to request the establishment of the DLCI specified by a SAPI of 0 and the Terminal Endpoint Identifier (TEI) specified or the packet data link on the BRI interface specified with the port argument.

Message Sent

“TE xx tt” where xx is the BRI interface number and tt is the TEI number.

“TE xx P” where xx is the BRI interface number and P indicates the packet data link.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_ARG the TEI specified is outside the range 0x0 to 0x7F
ILL_PORT the port is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to request that the data link specified by DLCI (Data Link Connection Identifier) address of [0, TEI] be established. If the TEI value is 0x7F, the packet data link with a SAPI of 16 is specified. The data link must be in the TEI assigned state. If successful, the data link will be put into the multi-frame established state.

Example

xds_tei_cnct(1, 2, 0x40); this will request establishment for a TEI of 64 (0x40) on BRI
 interface 2 of board 1.

xds_tei_disc

int xds_tei_disc(board_number, port, tei);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int tei; a value between 0x0 and 0x7E indicating the TEI of the
 data link to be disconnected, a value of 0x7F indicates the
 packet data link

Applicable boards

All XDS BRI boards

Purpose

This function is used to request the disconnection of the DLCI specified by a SAPI of 0 and the Terminal Endpoint Identifier (TEI) specified or the packet data link on the BRI interface specified with the port argument.

Message Sent

“TD xx tt” where xx is the BRI interface number and tt is the TEI number.

“TD xx P” where xx is the BRI interface number and P indicates the packet data link.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG the TEI specified is outside the range 0x0 to 0x7F
ILL_PORT the port is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to request that the data link specified by DLCI (Data Link Connection Identifier) address of [0, TEI] be disconnected. If the TEI argument is 0x7F, the packet data link with a SAPI of 16 is specified. Disconnection does not cause the removal of the TEI, but does place it in the TEI assigned state. The link may be reconnected by an attempt to send a message or a call to **xds_tei_cnct**.

Example

xds_tei_disc(1, 2, 0x40); this will request a disconnection for a TEI of 64 (0x40) on BRI interface 2 of board 1.

xds_tei_fixed

int xds_tei_fixed(board_number, port, tei);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int tei; a value between 0x0 and 0x3F indicating the fixed TEI

Applicable boards

All XDS BRI boards

Purpose

This function is used to specify a fixed Terminal Endpoint Identifier (TEI) to be used on the BRI interface specified with the port argument.

Message Sent

“TFxxtt” where **xx** is the BRI interface number and **tt** is the TEI number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_ARG the TEI specified is outside the range 0x0 to 0x3F
ILL_PORT the port is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to specify a fixed TEI (Terminal Endpoint Identifier) so that it will be available for sending or receiving Layer 3 messages. Fixed TEI are in the range 0x0 to 0x3F. Each port may have up to seven TEI's assigned. This number includes both fixed and dynamic TEI's. Normally, it is only necessary to specify a fixed TEI for a TE (terminal equipment) port. However, this command may also be used to reserve a TEI on an NT (network termination) port if there is a possibility that the assignment of dynamic TEIs may take up all seven possible TEIs. Dynamic TEI assignment using the **xds_tei_assign** command is to be preferred where it is an option.

Example

xds_tei_fixed(1, 2, 0x20); this will specify a TEI of 32 (0x20) on BRI interface 2 of board 1.

xds_tei_packet

int xds_tei_packet(board_number, port, tei);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int tei; a value between 0x0 and 0x7E indicating the packet TEI

Applicable boards

All XDS BRI boards

Purpose

This function is used to specify a packet Terminal Endpoint Identifier (TEI) to be used on the BRI interface specified with the port argument.

Message Sent

“TP xx tt” where xx is the BRI interface number and tt is the TEI number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG the TEI specified is outside the range 0x0 to 0x7F
ILL_PORT the port is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to specify a packet TEI (Terminal Endpoint Identifier) so that it will be available for sending or receiving Layer 3 messages with a SAPI of 16. The Packet TEI are in the range 0x0 to 0x7E. Each port supports a single packet data link that may be used for sending X.31 messages on the D channel. This is in addition to the seven fixed and dynamic TEI's links available for a SAPI of 0. The default packet TEI is 0.

Example

xds_tei_packet(1, 2, 0x20); this will specify a TEI of 32 (0x20) on BRI interface 2 of board 1.

xds_tei_query

int xds_tei_query(board_number, port, teiList);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int teiList[8]; an array for the return of the TEIs assigned to the port

Applicable boards

All XDS BRI boards

Purpose

This function is used to request a list of the TEIs assigned to the on the BRI interface specified with the port argument.

Message Sent

“TQxx” where xx is the BRI interface number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range
WRONG_QUERY the wrong query response was received
WRONG_BOARD a query response was received from the wrong board
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to request a list of TEIs (Terminal Endpoint Identifiers) that have been assigned on a BRI interface. This list will consists of eight TEIs, with the last TEI being that of the packet data link. Unassigned TEIs will have a value of 0xFF. Both dynamic and fixed TEIs will be reported. The broadcast TEI of 0x7F will not be included in the list, though it is available. The response message from the board will take the form “TQxxxxxxxxxxxxxxxx” where each pair of xx’s is a hexadecimal number representing a TEI.

Example

xds_tei_query(1, 2, &list); this will request a TEI list to be placed in the array & list on BRI interface 2 of board 1.

xds_tei_remove

int xds_tei_remove(board_number, port, tei);

unsigned char board_number;	a value indicating which board the command is for
int port;	the number of the port that indicates the BRI interface
int tei;	a value between 0x0 and 0x7F indicating the TEI to be removed, 0x7F will cause all TEIs to be removed

Applicable boards

All XDS BRI boards

Purpose

This function is used to request the removal of Terminal Endpoint Identifier (TEI) specified on the BRI interface specified with the port argument.

Message Sent

“TR xx tt” where xx is the BRI interface number and tt is the TEI number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG	the TEI specified is outside the range 0x0 to 0x7F
ILL_PORT	the port is out of valid range
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to request the removal of the TEI specified. This will cause the DLCI (Data Link Connection Identifier) at [0, TEI] to be eliminated. A TEI value of 0x7F will cause all TEIs in the range 0x0 to 0x7E to be removed. This command is only valid for NT (network termination) ports on S/T BRI Boards and LT (line termination) ports on U-Interface BRI Boards. The removal will be reported with a message of type 9 msg_sub 5.

Example

xds_tei_remove(1, 2, 0x40);	this will request removal of TEI 64 (0x40) on BRI interface 2 of board 1.
-----------------------------	---

xds_tei_verify

int xds_tei_verify(board_number, port, tei);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface
int tei; a value between 0x0 and 0x7F indicating the TEI to be
verified

Applicable boards

All XDS BRI boards

Purpose

This function is used to request the verification of the Terminal Endpoint Identifier (TEI) specified on the BRI interface specified with the port argument.

Message Sent

“TV xx tt” where xx is the BRI interface number and tt is the TEI number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG the TEI specified is outside the range 0x0 to 0x7F
ILL_PORT the port is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to request the verification of a TEI (Terminal Endpoint Identifier). This command is only valid on TE (terminal equipment) ports. The associated NT (network termination) or LT (line termination) will carry out the verification and make corrections if necessary by removing the TEI.

Example

xds_tei_verify(1, 2, 0x40); this will request the verification of TEI 64 (0x40) on BRI
interface 2 of board 1.

xds_set_bri_clock

xds_set_bri_clock(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the BRI interface

Applicable boards

XDS SCSA S/T BRI and XDS SCSA U-Interface BRI Boards

Purpose

This function is used to select a BRI port as the master clock source for the system. This port must be configured as a TE (Terminal Equipment) interface for an S/T interface board and as an N/T for a U-interface board.

Message Sent

“SC2p” where **p** is the port number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-SCSA BRI board was chosen

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The SCSA S/T BRI board has twelve Basic Rate ISDN interfaces while the SCSA U-Interface BRI board has four. This function can be used to select one of these interfaces to act as the source of the master clock for the system. This port must be configured as terminal equipment (TE) for a S/T board or as an N/T for a U-interface board and be connected to a Network Termination or Line Termination respectively, either a central office line or a line from a PBX or other switch intended to interface to terminal equipment. When this command is issued, the board will derive its clock from the BRI interface and supply the clocks to the SCbus. It is important that no other board in the system be supplying clocks to the SCbus when this function is called, as only one board can serve as master clock.

Example

xds_set_bri_clock(1, 0x4);

this will select port 4 on board 1 as the source of the system clock

xds_l3_restart

xds_l3_restart(board_number, b_channel, channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; the number of the b-channel that indicates the BRI interface
int channel; the number of the specific b-channel to be restarted

Applicable boards

All XDS BRI boards

Purpose

This function is used to restart a B-channel.

Message Sent

“DZxx” restart both B-channels, where xx is either of the B-channels

or

“DZxxC” restart a specific B-channel, where xx is the B-channel

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG the channel chosen was not 0 or 1
ILL_PORT the B-channel is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The Restart Indicator is used in RESTART and RESTART ACKnowledge messages to indicate which B-channels are involved. The restart procedure lets either one or both of the B-channels to be returned to the null state.

Examples

xds_l3_restart(1, 0x4, 0); this will restart both B-channels 4 on board 1
xds_l3_restart(1, 0x4, 1); this will restart only B-channel 4 on board 1

Basic Rate ISDN NI-1 Layer 3 'D' Functions

This page was intentionally left blank.

xds_l3_alert

xds_l3_alert(board_number, b_channel, progress, signal);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char progress;	a character indicating the progress indicator if any
char signal;	a character indicating the signal if any

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 ALERTing message for the call currently associated with the specified B-Channel. An ALERTing message is used to notify the caller that the called party is being informed of an incoming call.

Message Sent

“DAXxps” where **xx** is the B-channel number, **p** is the progress indicator, and **s** is the signal argument.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_PROGRESS	invalid progress indicator value
ILL_SIGNAL	invalid signal
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an ALERTing message to be sent for the call associated with the specified B-channel. If the port is configured as an NT (network termination), progress indication and signal information elements can be sent as part of the message.

For NT ports a progress indicator is part of the ALERTing message. Valid values for this are:

- I Inband information or appropriate pattern now available
- N No progress indicator

A value of 'I' should be used for speech or 3.1 kHz audiocalls, while 'N' should be used for data calls. The signal element for NT ports should be set to 'R' for ringback.

For TE ports, the progress and signal arguments should be set to 0x0.

Example

```
xds_13_alert(1, 0x4, 'I', 'R');
```

this will send an ALERTing message for B-channel 4 on board 1. The progress indicator is set for inband signal, the signal is ringback.

xds_l3_connect

xds_l3_connect(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 CONNect message for the call currently associated with the specified B-Channel. A CONNect message is used to notify the caller that the called party has answered an incoming call.

Message Sent

“DCxx” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a CONNect message to be sent for the call associated with the specified B-channel. This should be done when the called party answers.

Example

xds_l3_connect(1, 0x4); this will send a CONNect message for B-channel 4 on board 1.

xds_l3_disconnect

xds_l3_disconnect(board_number, b_channel, cause, signal);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for the disconnect
char signal;	a character indicating the signal if any

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 DISConnect message for the call currently associated with the specified B-Channel. A DISConnect message is used to notify either the network or the user that a disconnect sequence has been initiated.

Message Sent

“DDxxccs” where **xx** is the B-channel number, **cc** is the cause code, and **s** is the signal.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_SIGNAL	invalid signal value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a DISConnect message to be sent for the call associated with the specified B-channel. This should be done when initiating a disconnect sequence.

The cause code is a value used to indicate the reason for a particular message, in this case, why a disconnect is occurring. A cause value of 0x10 indicates normal clearing. Other values may indicate error conditions. For a list of cause codes and their meanings, see the XDS Layer 3 Protocol Software Reference Manual, Q.850 or the relevant Bellcore documentation.

If the port is configured as an NT (network termination), a signal element can be included in the DISConnect message. Valid values for the signal element are:

R	audible ringing tone on
N	network congestion/reorder tone on
B	busy tone on
C	confirmation tone
F	tones off

Example

<code>xds_l3_disconnect(1, 0x4, 0x10, 'F');</code>	this will send a DISConnect message for B-channel 4 on board 1, with a cause of 0x10 (normal clearing), and tones off.
--	--

xds_l3_feature_ind

xds_l3_feature_ind(board_number, b_channel, feature, status);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int feature;	a value between 0x0 and 0x3F indicating the feature indicator
char status;	a character indicating the status of the feature indicator

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 INFORMATION message to activate a feature indicator on an ISDN terminal or station set. Feature indicators are used to inform the user of the status of a particular feature.

Message Sent

“DFxxffa” where **xx** is the B-channel number, **ff** is the feature indicator code, and **a** is the feature status.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_FEATURE	invalid feature value
ILL_ARG	invalid status value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an INFOrmation message to be sent to control the specified feature indication. If the feature indicator is not associated with a call the specified B-channel should correspond to that associated with the data-link of the terminal. This will normally be the even B-channel. If the indicator is associated with a call, the B-channel of that call should be specified. A feature indication message is normally sent by a port configured as an NT (network termination).

Feature indicators can have a range of 0x0 to 0x3F. There is no universal standard as to the meaning of a particular feature indicator number, and different ISDN station sets will have varying numbers of indicators, some of which have fixed numbers and some of which can be programmed. The specific feature indicator number used will depend on the equipment and application.

Four status values are allowed which control the appearance of the indicator. The valid status characters and corresponding appearance are:

I	idle, off
A	active, solid illumination
P	pending, a slow blink
Q	prompt, a fast blink

The blinking rate will depend on the specific equipment used.

Example

`xds_l3_feature_ind(1, 0x4, 0x3F, 'P');` this will send an INFOrmation message for B-channel 4 on board 1, for feature indicator 0x3F with a status of pending.

xds_l3_feature_key

xds_l3_feature_key(board_number, b_channel, feature);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int feature; a value between 0x0 and 0x3F indicating the feature key

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 INFOrmation message to indicate a feature key activation. Feature keys are used to notify the network for the purpose of controlling a feature.

Message Sent

“DFxxff” where **xx** is the B-channel number and **ff** is the feature key

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_FEATURE invalid feature value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an INFOrmation message to be sent to control the specified feature. If a call is not active on the specified B-channel, the feature activation will be associated with the data-link that corresponds to that B-channel, otherwise, if a call is active, the feature key will be associated with the call. A feature key message is normally sent by a port configured as a TE (terminal equipment).

Feature keys can have a range of 0x0 to 0x3F. There is no universal standard as to the meaning of a particular feature key or feature key number. This will depend on the specifics of the switch and on any pre-subscribed features associated with the interface.

Example

xds_l3_feature_key(1, 0x4, 0x2); this will send a INFOrmation message for B-channel 4 on board 1, for feature key 2.

xds_l3_hold

xds_l3_hold(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 HOLD message for the call currently associated with the specified B-Channel. A HOLD message is used to release the B-channel allocated to a call, but retain the call and its reference. A hold can be initiated by either the network or a terminal.

Message Sent

“DHxx” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a HOLD message to be sent for the call associated with the specified B-channel. This can be done by either the network or a terminal. If the hold is accepted, a HOLD ACKnowledge message will be sent in response. This response message will include the call reference. It is the responsibility of the application to retain this call reference as it is needed to retrieve the call. A HOLD REJect message will be sent in response if the hold is not accepted. This will include the cause for rejection.

It should be noted that not all network switches support the HOLD feature. On others, it may be available by pre-subscription at the time the interface is ordered. Similarly, terminal equipment such as ISDN station sets may or may not support the feature.

Example

xds_l3_hold(1, 0x4); this will send a HOLD message for B-channel 4 on board 1.

xds_l3_hold_ack

xds_l3_hold_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 HOLD_ACKnowledge message for the call currently associated with the specified B-Channel. A HOLD_ACK message is used to respond to a HOLD message and accept the release of the B-channel while retaining the call.

Message Sent

“DAXxA” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a HOLD ACKnowledge message to be sent for the call associated with the specified B-channel. This should be done in response to a HOLD message. The B-channel will be released for use with another call. The application should retain the call reference received in the hold message as it will be used when the call is retrieved.

Example

xds_l3_hold_ack(1, 0x4); this will send a HOLD ACKnowledge message for B-channel 4 on board 1.

xds_l3_hold_rej

xds_l3_hold_rej(board_number, b_channel, cause);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for the hold reject

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 HOLD REJECT message for the call currently associated with the specified B-Channel. A HOLD REJECT message is used to respond to a HOLD message and reject the release of the B-channel.

Message Sent

“DAXxRcc” where **xx** is the B-channel number and **cc** is the cause code

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a HOLD REJECT message to be sent for the call associated with the specified B-channel. This should be done in response to a Hold message when the B-channel is not to be released. The cause value should indicate the reason for rejecting the hold. Typical reasons would be 0x45, “requested facility not implemented” if the application does not support hold, or 0x3, “requested facility not subscribed” when the hold feature is not enabled for a specific interface. For a list of cause codes and their meaning, see the XDS Layer 3 Protocol Software Manual, Q.850 or the relevant Bellcore documentation.

Example

xds_l3_hold_rej(1, 4, 0x45);	this will send a HOLD REJECT message for B-channel 4 on board 1 with a cause of 0x45, “requested facility not implemented”.
------------------------------	---

xds_l3_info_request

xds_l3_info_request(board_number, b_channel, query, complete);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char query;	a character indicating the query type
char complete;	a 0x1 if the query is complete, otherwise 0

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 INFOrmation message with an information request. This is done to query a terminal for information such as the SPID or address digits.

Message Sent

“DIxxQq” where **xx** is the B-channel number, and **q** is the information request type

“DIxxQqC” if the information is complete

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_ARG	invalid query or complete value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an INFOrmation message containing an information request. This should only be done on a port set to type NT if an S/T interface or to type LT if a U-Interface. Depending on the request type this may be for the terminal associated with the B- channel of for the call currently associated with the specified B-channel. The information will be returned in an INFOrmation message. It may be desirable to send a request complete message once the information and been received or upon a time out.

The valid information requests are:

A - Authorization Code

D - Address Digits

S - Terminal Identification (SPID)

U - Undefined

Example

```
xds_l3_info_request(1, 0x4, 'S', 0);
```

this will send a request for the SPID for B-channel 4 on board 1.

xds_l3_key_hold

xds_l3_key_hold(board_number, b_channel, reference);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int reference; a value between 0x00 and 0xFF that indicates the call
reference of the call for which the notification is intended

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 KEY_HOLD message to inform an EKTS terminal that a call has been answered by another terminal.

Message Sent

“DExxHrr” where **xx** is the B-channel number and **rr** is the call reference.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_REFERENCE invalid call reference value

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a KEY_HOLD message to be sent from an NT port to a Electronic Key Telephone Service (EKTS) terminal. The notification is used to inform the terminal that the call has been answered by another terminal. This is done so that the terminal may reflect the change of state by changing the state of an indicator (i.e. controlling whether the indicator blinks or not.)

Example

xds_l3_key_hold(1, 0x4, 0x1); this will send a KEY_HOLD message for B-channel
4 on board 1 with a notification of Call Retrieved.

xds_l3_key_setup

xds_l3_key_setup(board_number, b_channel, bearer_cap, called_num, call_app);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char bearer_cap; a character indicating the type of call, i.e. speech or data
char *called_num; a pointer to an ASCII string with the called number
int call_app; a value between 0x1 and 0xFE indicating the call
 appearance

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 KEY_SETUP message to an EKTS terminal to inform that terminal that another terminal is originating a call a call appearance or called number.

Message(s) Sent

For an EKTS port "DExxSb/#" where **xx** is the B-channel, **b** is the bearer capability and **/#** is the called party number.

For a CACH EKTS port "DExxSbA=ca" where **xx** is the B-channel, **b** is the bearer capability and **ca** is the call appearance number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_FEATURE invalid feature value
ILL_ARG an invalid bearer capability, call appearance value, or an invalid
 number format was used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a KEY_SETUP message to be sent to an Electronic Key Telephone Service (EKTS) terminal to indicate that another terminal in the EKTS group is originating a call on a Directory Number (DN) or call appearance. The DN format is used for Basic EKTS while the call appearance form is used for Call Appearance Call Handling (CACH) EKTS. If the CACH EKTS format is used, the called number may be set to a NULL string.

Example

```
xds_l3_key_setup(1, 0x4, 'S', '', 0x3)
```

This will send a KEY_SETUP message for B-channel 4 on board 1 with a bearer capability of “speech” and a call appearance number of 3.

xds_l3_notify

xds_l3_notify(board_number, b_channel, notification, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char notification;	a character indicating the notification type
int reference;	a value between 0x00 and 0xFF that indicates the call reference of the call for which the notification is intended

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 NOTIFY message to inform an EKTS terminal of the change of state of a call.

Message Sent

“DExxNrr” where **xx** is the B-channel number, **n** is the notification indicator, and **rr** is the call reference.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_REFERENCE	invalid call reference value
ILL_ARG	an invalid notification value was used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a NOTIFY message to be sent from an NT port to a Electronic Key Telephone Service (EKTS) terminal. The notification is used to inform the terminal of a change of state of a call so that the terminal may reflect the change of state by changing the state of an indicator (i.e. controlling whether the indicator blinks or not.)

The valid notification indicators are:

B - Call Bridged

H - Call Held

M - Monitored User Idle

R - Call Retrieved

Example

```
xds_l3_notify(1, 0x4, 'R', 0x1);
```

this will send a NOTIFY message for B-channel 4 on board 1 with a notification of Call Retrieved.

xds_l3_proceed

xds_l3_proceed(board_number, b_channel, progress, signal);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char progress;	a character indicating the progress indicator
char signal;	a character indicating the signal if any

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 CALL PROCEEDing message for the call currently associated with the specified B-Channel. A CALL PROCEEDing message is used to notify the caller that all information necessary to process a call has been received and that call establishment has been initiated.

Message Sent

“DPxxps” where **xx** is the B-channel number, **p** is the progress indicator, and **s** is the signal

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_PROGRESS	invalid call progress indicator value
ILL_SIGNAL	an invalid signal value was used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a CALL PROCEEDing message to be sent for the call associated with the specified B-channel. This should be done in response to a SETUP message for Enbloc sending, or when all necessary digits have been received for Overlap sending when the call can be initiated.

For ports that are configured as a TE (terminal equipment), the progress and signal values are not used. For an NT (network termination) port, the progress indicator may take values of:

I Inband information or appropriate pattern now available
N No progress indicator

The signal element is optional and may take the value 'C' for confirmation tone, or 0. If the signal element is indicates confirmation tone, the progress indicator should be set to 'I'.

Example

```
xds_13_proceed(1, 0x4, 'N', 0);
```

this will send a CALL PROCEEDing message for B-channel 4 on board 1, with no progress indicator or signal.

xds_l3_progress

xds_l3_progress(board_number, b_channel, cause, progress, signal);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x0 and 0xff that indicates the reason for
 the PROGRESS message
char progress; a character indicating the progress indicator
char signal; a character indicating the signal if any

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 PROGRESS message for the call currently associated with the specified B-Channel. A PROGRESS message is used to notify the caller of interworking with non-ISDNs, a call is routed to an inband tone or announcement, or when a progress delay at the destination is reported.

Message Sent

“DPxxPccps” where **xx** is the B-channel number, **cc** is the cause value, **p** is the progress indicator, and **s** is the signal.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_PROGRESS	an invalid call progress value was used
ILL_SIGNAL	an invalid signal element was used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a PROGRESS message to be sent for the call associated with the specified B-channel. This command is only valid for NT (network termination) ports.

A PROGRESS message is sent to report an interworking situation such as the destination is non-ISDN, or a situation where the call, though initiated can not be completed, such as the called party is busy. In those cases, an inband signal such as busy tone is usually present and indicated by the progress indicator and signal elements in the message. An application would send a PROGRESS message when it realizes that a call can not be completed, or when interworking with the destination switch results in the receipt of a PROGRESS message from the destination.

The cause code is a value used to indicate the reason for the PROGRESS message. Examples would be a cause code of 0x11, "User busy", or 0x12, "No user responding". For a list of cause codes and their meaning, see the XDS Layer 3 Protocol Software Reference Manual, Q.850, or the relevant Bellcore documentation.

Progress values that are valid may be:

- C Call is not end to end ISDN, call progress information may be available inband
- D Destination address is non-ISDN
- I Inband information or appropriate pattern now available
- W Delay in response at destination interface

Valid Signal values are:

- D Dial tone
- R Audible ringback
- N Network Congestion/Reorder tone
- B Busy tone
- C Confirmation tone
- F Tones off

Example

```
xds_13_progress(1, 0x4, 0x11, 'I', 'B');
```

this will send a PROGRESS message for B-channel 4 on board 1. The cause is 0x11, "User busy", the progress indicator is inband information, and the signal is busy.

xds_l3_query_spid

xds_l3_query_spid(board_number, b_channel, spid_info);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
struct spid_dn *spid_info; a structure to contain the SPID and DN information
{
char spid[15]; an array to contain the SPID
char dn[8]; an array to contain the Directory Number
}

Applicable boards

All XDS BRI boards

Purpose

This function is used to query the board to obtain the SPID and default Directory Number for a specific data-link.

Message Sent

“DQxx” where xx is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call
WRONG_QUERY the wrong query response was received
WRONG_BOARD a query response was received from the wrong board

Comments

This command is used to obtain the SPID (Service Profile Identifier) and default DN (directory number) associated with a specific data-link. Data-links are identified using numbers that take the same form as the B-channel numbers for a port. Each port may have one or two data-link and SPIDs associated with it. The data-link used for voice calls is associated with the even numbered B-channel and the data-link for data calls is associated with the odd numbered B-channel. If only one SPID and data-link is present, it will use the even number. Each B-channel has a default directory number that is used as the called party number unless otherwise specified in a SETUP message (see **xds_l3_setup**).

For ports configured as a TE (terminal equipment), the SPID and DN are programmed in by the application, and therefore, this command is only of use for diagnostic purposes, i.e. to see that the correct information has been programmed in. For NT (network termination) ports, the DN is programmed, but the SPID is received from the TEs connected to the port. This command may therefore be of use to determine which terminals are connected to which ports.

For the purposes of the XDS BRI Board, TE SPIDs are not used for call handling, but are used to associate TE equipment with data-links. The SPID programmed into the TE equipment must be at least 12 digits long. The 12th digit for the voice SPID must be a '1', while the 12th digit for the data SPID must be a '2'. If two voice terminals are present, the second voice terminal must have a SPID where the 12th digit is '3'. If only one SPID is needed, it should have a 12th digit of '1'. The other digits are ignored by the board.

The default directory number, or DN, is a seven digit number in local ISDN format. It is used as the default called party number for calls placed to a TE by a port configured as an NT, unless otherwise specified. For a TE port, the DN is used as the calling number for calls placed using the specified B-channel.

Example

```
xds_l3_query_spid(0x1, 0x4, spid_buf)
```

This would request the SPID and DN for the B1 channel of the third port on board 1. The SPID and DN would be returned in the structure spid_buf.

xds_l3_query_status

xds_l3_query_status(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 STATUS Enquiry message to the board. This can be done to prompt a terminal to respond with a STATUS message. This command is valid only for ports that are configured as an NT (network termination).

Message Sent

“DXxx” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to query an ISDN terminal about the call state of a call in a case where a received message indicates there may be a call processing discrepancy. The terminal, if it responds will send a STATUS message. As this response is dependent on the terminal and not on the board, this message will be returned on the receive message queue and will be retrieved with the **xds_msg_receive** function rather than the **xds_query_receive** function. This function is only valid for ports that are configured as an NT (network termination) port, and only for B-channels that have a call in process.

Example

xds_l3_query_status(0x1, 0x4) This would send a STATUS ENquiry message for the B1 channel of the third port on board 1.

xds_l3_rel_com

xds_l3_rel_com(board_number, b_channel, cause, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for the release
int reference;	a value between 0x00 and 0xFF that indicates the call reference for the call being released if it does not have an B-channel allocated

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 REL_COMPLETE message for the call currently associated with the specified B-Channel or that is specified by the call reference. A REL_COMPLETE message is used to release a call reference and terminate a call in cases where a DISCONNECT message is inappropriate.

Message Sent

“DRxxccrr” where **xx** is the B-channel number **cc** is the cause value and **rr** is the optional call reference.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_REFERENCE	an invalid call reference value was used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a REL_COMplete message to be sent for the call associated with the specified B-channel. This should be done to release a call and associated call reference under some circumstances. An example, is when a call has been placed on hold and is to be cleared, or when no TE equipment has responded to a SETUP message. If the call has been allocated a B-channel, a DISConnect should be sent to clear a call (see **xds_l3_disconnect**).

The cause code is a value used to indicate the reason for the REL_COMplete message. Examples would be a cause code of 0x10, “Normal clearing”, or 0x41, “Bearer capability not supported”. For a list of cause codes and their meaning, see the XDS Layer 3 Protocol Software Manual, Q.850, or the relevant Bellcore documentation.

In cases where the call is not currently allocated a B-channel, for example when on hold, it is necessary to use the call reference. This is a value in the range of 0x01 to 0x7F or 0x81 to 0xFF. This value should have been retained by the application from the received hold message. If the call reference is not implicitly specified, this value should be 0. The B channel in this case should correspond to the data-link of the call.

Example

```
xds_l3_rel_com(1, 0x4, 0x10, 0x02);
```

this will send a RELease COMplete message for B-channel 4 on board 1 with a cause value of 0x10, “Normal Clearing” and a call reference of 0x02.

xds_l3_retrieve

xds_l3_retrieve(board_number, b_channel, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int reference;	a value between 0x00 and 0xFF that indicates the call reference of the call being retrieved

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 RETrieve message for the call currently specified with the call reference. The call is to be allocated the specified B-channel, which must be available. A RETrieve message is used to reestablish a connection to a call placed on hold.

Message Sent

“DGxxrr” where **xx** is the B-channel number and **rr** is the call reference

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_REFERENCE	an invalid call reference value was used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a RETrieve message to be sent for the call with the specified reference. The call will be allocated the specified B-channel. This is done when retrieving a call previously placed on hold using a HOLD message. Note that while both the NT (network termination) and TE (terminal equipment) side of a connection can place a call on hold, only the TE user can retrieve a call. Also, the selected B-channel must be available.

The NT side should respond with either a RETrieve ACKnowledge message, in which case, the connection is restored, or a RETrieve REJect message, in which case, the call remains in the held state.

The call reference used in this message is the one given in the original HOLD or HOLD ACKnowledge message. It is the responsibility of the application to retain this reference as long as the call is in the hold state. Failure to do so may result in irretrievable or un-releasable calls.

Example

```
xds_13_retrieve(1, 0x4, 0x83);
```

this will send a RETreive message for B-channel 4 on board 1 with a call reference of 0x83.

xds_l3_retrieve_ack

xds_l3_retrieve_ack(board_number, b_channel, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int b_channel;	the number of the B-channel that indicates the BRI interface
int reference;	a value between 0x00 and 0xFF indicating the call being retrieved

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 RETrieve ACKnowledge message for the call specified by the call reference. The call will be allocated the specified B-channel. A RETrieve ACKnowledge message is used to grant a RETrieve request.

Message Sent

“DGxxArr” where **xx** is the B-channel number and **rr** is the call reference number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_REFERENCE an invalid call reference value was used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a RETrieve ACKnowledge message to be sent for the specified call reference. The call will be allocated the specified B-channel. Note, that this channel must match that received in the RETrieve message. If a RETrieve ACKnowledge message is sent, the connection to the call is reestablished on the specified B channel. Note, that as only a TE (terminal equipment) port can send a retrieve message, only an NT (network termination) port can grant the retrieval by sending a RETrieve ACKnowledge message.

Example

xds_l3_retrieve_ack(1, 0x4, 0x3);	this will send a RETrieve ACKnowledge message for B-channel 4 on board 1, with call reference of 0x3.
-----------------------------------	---

xds_l3_retrieve_rej

xds_l3_retrieve_rej(board_number, b_channel, cause, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for rejecting the retrieval
int reference;	a value between 0x00 and 0xFF that indicates the call for which retrieval is being rejected

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 RETrieve REJect message for the call specified by the call reference. A RETrieve REJect message is used to deny a RETrieve request message.

Message Sent

“DGxxRccrr” where **xx** is the B-channel number, **cc** is the cause, and **rr** is the call reference.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_REFERENCE	an invalid call reference value was used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a RETrieve REJect message to be sent for the call specified by the call reference value. The B-channel should match that in the received RETrieve message. Note, that as only a TE (terminal equipment) port can send a retrieve message, only an NT (network termination) port can grant the retrieval by sending a RETrieve REJect message. The cause code is a value used to indicate the reason for the PROGress message. Examples would be a cause code of 0x06, “Channel unacceptable”, or 0x2A, “Switching equipment congestion”. For a list of cause codes and there meaning, see the XDS Layer 3 Protocol Software Reference Manual, Q.850, or the relevant Bellcore documentation.

Example

```
xds_l3_retrieve_rej(1,0x4,2A,0x03);
```

this will send a RETrieve REJect message for B-channel 4 on board 1, with a cause of 0x2A, “Switching equipment congestion”, and a call reference of 0x3.

xds_l3_setup

xds_l3_setup(board_number, b_channel, port_typ, bearer_cap, progress, signal, called_num, calling_num, call_app);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char port_type;	a character indicating whether the port is of type NT or TE
char bearer_cap;	a character indicating the type of call, i.e. speech or data
char progress;	a character indicating the progress indicator
char signal;	a character indicating the alerting signal if any
char *called_num;	a pointer to an ASCII string with the called number
char *calling_num;	a pointer to an ASCII string with the calling number
int call_app;	a value between 0x1 and 0xFE indicating the call appearance

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 SETUP message. A SETUP message is used to initiate a call. As the SETUP message takes a different form for NT and TE interface, the presence of some arguments will depend on the port type.

Message Sent

For an NT interface: “DSxxbps(calling#/called#)”, where **xx** is the B-channel number, **b** is the bearer capability, **p** is the progress indicator, **s** is the signal, **calling#** is the directory number of the calling party and optionally **called#** is the number being called.

For an NT interface to CACH EKTS terminals: “DSxxbps(calling#)A=ca” where **ca** is the two digit call appearance number in hex.

For a TE interface: “DSxxbcalled#”, where **xx** is the B-channel number, **b** is the bearer capability, **called #** is the number being called.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_TYPE	invalid port type
ILL_ARG	an invalid bearer capability, call appearance number, or an invalid calling or called number format was used
ILL_SIGNAL	invalid signal value
ILL_PROGRESS	an invalid call progress indicator value was used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a SETUP message to be sent to initiate a call. The board level command takes a different form depending on whether the port is configured as an NT (network termination) or a TE (terminal equipment). The progress and signal elements are not present in the SETUP message for a TE, and the treatment of the directory numbers is different. The port type must be specified by the **port_type** argument. Valid values are 'N' for an NT interface and 'T' for a TE interface. Note, that for a U-Interface board, the 'N' will refer to a Line Termination or LT port type and the 'T' will refer to a Network Termination or NT port type.

The XDS and Infinity BRI Boards uses exclusive B channel allocation only. If the B-channel specified for the SETUP message is in use at the time the message is sent, the channel ID will be for no B-channel. This should only be used for ports of type 'N'.

The bearer capability indicates the type of call being made, i.e. voice or data. There are four valid choices:

- A 3.1 kHz. audio (used for voice and modems)
- D unrestricted digital information, 64 kbps
- R rate adapted data, rate adapted for 56 kbps
- S speech (suitable for voice, but not modems)

For NT ports, a progress indicator must be specified. This argument is used in interworking situations where the call is not completely ISDN. The valid values are:

- C Call is not end-to-end ISDN
- O Originating address is non-ISDN
- N No progress indicator

Note that if there are no interworking problems, the progress indicator must be set to 'N'. The signal argument is used to indicate which type of alerting is to be used by the terminal. This argument is used only for NT ports. Valid values are:

- N Normal alerting
- D Distinctive alerting
- S Special alerting
- I Intercom alerting
- R Reminder ring
- F Alerting off

It should be noted that there is no uniform format for the various forms of alerting. Normal ringing typically sounds like the standard 2 on 4 off ring of an analog phone, but the other alerting codes will produce different patterns of tones or beeps depending on the specific piece of terminal equipment.

For TE ports, the called number is the directory number of the destination. The calling number

element will use the default DN of the B-channel on which the call is being placed. The called number can be up to 15 digits long.

For NT ports, only the calling number is required. If the called number is a NULL string, the default DN of the B-channel will be used as the called number element, otherwise, the called number specified will be used. The calling number can be up to 15 digits long. If a called number is specified, the total number of digits in the called and calling number is restricted to 24. This should not be a problem as a 7 digit called number is normally sufficient.

The call_app value is used for CACH EKTS calls only for an NT port. For non-CACH EKTS calls, this value should be 0.

Example

For an NT port

`xds_l3_setup(1, 0x4, 'N', 'S', 'N', 'N', '', "5551212", 0);` this will send a SETUP message for B-channel 4 on board 1 with a bearer capability of speech, no progress indicator, normal alerting, the default called number and calling number of 555-1212.

For an NT CACH EKTS port

`xds_l3_setup(1, 0x4, 'N', 'S', 'N', 'N', '', "5551212", 3);` this will send a SETUP message for B-channel 4 on board 1 with a bearer capability of speech, no progress indicator, normal alerting, a calling number of 555-1212, and call appearance of 3.

For a TE port

`xds_l3_setup(1, 0x4, 'T', 'D', "18005551212", "", 0);` this will send a SETUP message for B-channel 4 on board 1 with a bearer capability of data and a called number of 1-800-555-1212.

xds_l3_spid

xds_l3_spid(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 INFOrmation message containing the SPID for the data-link associated with the specified B-Channel. The SPID (Service Profile Identifier) is used by the network switch to identify a particular piece of terminal equipment.

Message Sent

“DIxxS” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an INFOrmation message to be sent containing the SPID for the data-link associated with the specified B-channel. The SPID must first be programmed into the board using the function **xds_set_bri_dn**. This information can be stored in the EEAROM on the board using the **xds_set_config** function so that it does not have to be reloaded every time the computer is rebooted.

This command is valid only for ports configured as TE (terminal equipment). If only one SPID is required on the port, it should be sent for the even B-channel. If two SPIDs are required, it should be sent for both B-channels.

A SPID must be sent to the network before calls can be placed or received. The data-link must first be established and a TEI assigned. This normally takes place automatically within a few seconds of power up. A TEI assigned message will be received for each data-link on the port.

Example

xds_l3_spid(1, 0x4); this will send an INFOrmation message for B-channel 4 on board 1 containing the SPID.

xds_l3_text

xds_l3_text(board_number, b_channel, mode, l3_text);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char mode;	a value between 0x0 and 0x4 indicating the text operation
char *l3_text	a pointer to a NULL terminated string of up to 20 ASCII characters

Applicable boards

All XDS BRI boards

Purpose

This function is used to prepare text to be included in a Layer 3 message. Several modes are available that allow inclusion of text into a SETUP message, or provide for sending an INFORMATION message with text. Only NT (network termination) ports can send text.

Message Sent

“DTxxC0” where **xx** is the B-channel, clears the buffer

“DIxxTtext” where **xx** is the B-channel number to send a single line of **text**

“DTxxC1” where **xx** is the B-channel and line 1 is the line to clear

“DTxxC2” where **xx** is the B-channel and line 2 is the line to clear

“DIxxL1text” where **xx** is the B-channel and line 1 is the line for the **text** to be sent

“DIxxL2text” where **xx** is the B-channel and line 2 is the line for the **text** to be sent

“DIxxB” where **xx** is the B-channel to send the buffer

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_ARG text message is too long

ILL_MODE invalid call forwarding mode

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to prepare text to send in a Layer 3 message. Display text is arranged in lines of up to 20 ASCII characters. Text can be sent immediately, or it can be put in a buffer with 2 lines. Text can either be sent as an INFOrmation message, or as part of a SETUP message. The “mode” controls which text operation is carried out. The valid mode values are:

0x0	clear the buffer
0x1	send the text in the buffer as an INFOrmation message
0x2	send a line of text immediately as an INFOrmation message
0x3	set line 1 of the buffer
0x4	set line 2 of the buffer
0x5	clear line 1 of the buffer
0x6	clear line 2 of the buffer

Each B-channel has its own two-line buffer. If a SETUP command is sent, it will include any text in the buffer as an information element. The text buffer is cleared whenever a RELEase COMplete message is sent or received by the board.

If a call is allocated a B-channel, then any INFOrmation message containing text will be sent with the call reference for that call, and the terminal will associate the text with the call. If the B-channel is not allocated to a call, the INFOrmation message will not have a call reference. The even B-channel will send text using the data-link associated with the 1st SPID, and the odd B-channel will use the data-link associated with the 2nd SPID. Where a terminal has two SPIDs, one for voice and one for data, the first SPID will normally be used. Most terminals will display the text. However, at least some terminals will clear the display after a few seconds.

Example

<code>xds_l3_text(1, 0x4, 0, ‘');</code>	this will clear the text buffer for B-channel 4 on board 1.
<code>xds_l3_text(1, 0x4, 1, “This is text”);</code>	this will send the text “This is text” for B channel 4 on board 1.
<code>xds_l3_text(1, 0x4, 3, “This is line 1”);</code>	will set line 1 of the buffer for B channel 4 on board 1 to “This is line 1”
<code>xds_l3_text(1, 0x4, 2, ‘');</code>	this will send the contents of the buffer as an INFOrmation message for B-channel 4 on board 1.

xds_l3_message

int xds_l3_message(board_number, port, sapi, tei, type, l3MsgLen, l3message);
unsigned char board_number; a value indicating which board the command is for
int port; a value indicating the port to control
int sapi; a value between 0x0 and 03F indicating the SAPI to be used

int tei; a value between 0x0 and 0x7F indicating the TEI to be used
char type; either 'C' for command or 'R' for response
unsigned short l3MsgLen; the number of octets in the Layer 3 message
unsigned char *l3message; a pointer to an array containing the Layer 3 message

Applicable boards

All XDS BRI boards

Purpose

This function is used to send a Layer 3 message using the DLCI specified by the SAPI and TEI specified on the BRI interface specified by the port argument. The message is contained in the array pointed to by *message and has a length in octets of l3MsgLen.

Message Sent

"LC xx yyzz" where **C** is the message type, **xx** is the BRI interface number, **yy** is the SAPI and **zz** is the TEI number. The Layer 3 message and length is placed in the auxiliary mailbox.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_TYPE invalid message type
ILL_ARG the TEI specified is outside the range 0x0 to 0x7E, or the SAPI is outside the range 0x0 to 0x3F.
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a Layer 3 message using the data link specified by the DLCI (Data Link Connection Identifier) composed of the SAPI (Service Access Point Identifier) and TEI (Terminal Endpoint Identifier). The SAPI has a range of 0-63 while the TEI has a range of 0-127. The normal rules for allowed DLCI addresses apply, and the board currently only supports DLCIs of [63, 127] and in the range [0, 0-127] and [16, 0-126]. The Layer 3 message itself consists of 1 to 260 octets. The contents of the message are not restricted to a Q.931 format. The TEI must already be assigned to the interface if it is in the range 0-126. The broadcast TEI of 127 is always available, and will result in an unnumbered information frame being sent. Two types of messages are allowed, commands and responses, specified by a type of 'C' or a 'R'. The normal type should be 'C' or command, and Q.931 specifies only commands.

Example

```
xds_l3_message(0, 2, 0, 0x40, 'C', 4, cncmsg);
```

this will send a Layer 3 message 4 octets in length with a SAPI of 0 and a TEI of 64 on port 2 of board 0. In this case, cncmsg is an array of four octets that is a Q.931 connect message.

```
cncmsg[4] = {0x8, 0x1, 0x81, 0x7};
```

Basic Rate EuroISDN Layer 3 'D' Functions

This page was intentionally left blank.

xds_euro_l3_alert

xds_euro_l3_alert(board_number, b_channel, progress);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char progress; a character indicating the progress indicator if any

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 ALERTing message for the call currently associated with the specified B-Channel. An ALERTing message is used to notify the caller that the called party is being informed of an incoming call.

Message Sent

“DAXxp” where **xx** is the B-channel number, and **p** is the progress indicator.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_PROGRESS invalid progress indicator value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an ALERTing message to be sent for the call associated with the specified B-channel. If the port is configured as an NT (network termination), progress indication and signal information elements can be sent as part of the message.

Valid values for the progress indicator are:

C Call not end-to-end ISDN
D Destination not ISDN
I Inband information or appropriate pattern now available
N No progress indicator

Example

xds_euro_l3_alert(1, 0x4, 'I'); this will send an ALERTing message for B-channel 4 on board 1. The progress indicator is set for inband signal.

xds_euro_l3_call_forwarding

xds_euro_l3_call_forwarding(board_number, b_channel, mode, id, operation, procedure, service, forwarding_num, subscriber_num, error_code)

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char mode;	a value indicating the command mode to be used
int id;	the invoke id value
char operation;	a value indicating which operation to be used
char procedure;	a value indicating which procedure to be used
char service;	a value indicating which basic service to be used
char *forwarding_num;	a pointer to an ASCII string with the number to be forwarded
char *subscriber_num;	a pointer to an ASCII string with the subscriber number
int error;	the value of an error code

Applicable boards

All EuroISDN BRI Boards

Purpose

This function is used to send a Layer 3 FACILITY message to control the call forwarding. It takes on one of three different forms, described below.

Message Sent

“DFxxFlidopb##/##”	Invoke mode, where xx is the b-channel, id is the invoke id value, o is the operation mode, p is the procedure mode, b is the basic service mode, the # is the number forwarded to, and /## is the subscriber number
“DFxxFRid”	Return Result mode, where xx is the b-channel, id is the invoke id value
“DFxxFEidee”	Return Error mode, where xx is the b-channel, id is the invoke id value, ee is the error value

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_ARG	invalid ID tag code, operation value, procedure value, called number, or calling number
ILL_MODE	invalid call forwarding mode
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause various messages to be sent according to how the function is called.

The legal values for **mode** are 'I' = invoke, 'R' = return results, and 'E' = return error. The legal values for **id** are from 0 to 0xFF. The legal values for **operation** can be 'F' = call forwarding off, 'N' = call forwarding on, and 'Q' = call forwarding interrogate. The legal **procedure** values are 'A' = call forward always CFU, 'B' = call forward busy CFB, and 'N' = call forward on answer CFNR. The **basic service** values are as follows: 'A' = 1, 'B' = 3, 'C' = 20h, and 'D' = others, 0.

Examples

```
xds_euro_l3_call_forwarding(1, 0x4, 'I', 0x3F, 'F', 'B', 'A', '9', '3');
```

this will send an invoke message for B-channel 4, on board 1, an invoke id of 3F, call forwarding off 'F', call forward busy CFB 'B', basic service value A 'A', '9' is the forwarded number, and '3' is the subscriber number.

```
xds_euro_l3_call_forwarding(1, 0x4, 'R', 0x3F);
```

this will send a return result message for B-channel 4, on board 1, and an invoke id of 3F

```
xds_euro_l3_call_forwarding(1, 0x4, 'E', 0x3F, 0x93);
```

this will send a return error message for B-channel 4, on board 1, an invoke id of 3F, and a value of 93 for the error

xds_euro_l3_called_num

xds_euro_l3_called_num(board_number, b_channel, complete, called_num);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char complete; a value indicating whether sending complete element
 should be included
char *called_num; a pointer to an ASCII string with the called number

Applicable boards

All EuroISDN BRI Boards

Purpose

This function is used to send a Layer 3 INFORMATION message containing called party number information for the call currently associated with the specified B-Channel. This is done as part of the call setup procedure when using overlap sending.

Message Sent

“DKxxk” where **xx** is the B-channel number and **k** is the called party number digit
“DKxxC” where the sending complete element is to be sent (complete = 0x1)

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an INFORMATION message containing a called party number element to be sent for the call associated with the specified B-channel. This should be done as part of the overlap sending procedure. The function **xds_euro_l3_setup** should first be called with a NULL called number string. Typically, only the user or terminal side of a BRI interface will use overlap sending. One or more digits may be sent in each message. If the value of complete is 0x1, an INFORMATION message with the Sending Complete element will be sent. This may be done after the last digit of the called party number has been sent, however, it is not mandatory.

Example

`xds_euro_l3_called_num(1, 0x4, 0, '3');` this will send an INFORMATION message for B-channel 4 on board 1 with a called party number digit of '3'.

xds_euro_l3_connect

xds_euro_l3_connect(board_number, b_channel, progress);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char progress; a character indicating the progress indicator if any

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 CONNect message for the call currently associated with the specified B-Channel. A CONNect message is used to notify the caller that the called party has answered an incoming call.

Message Sent

“DCxx” where xx is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_PROGRESS invalid progress indicator value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a CONNect message to be sent for the call associated with the specified B-channel. This should be done when the called party answers.

Valid values for the progress indicator are:

- C Call not end to end ISDN
- D Destination not ISDN
- I Inband information or appropriate pattern now available
- O Origination not ISDN
- R Return to ISDN
- N No progress indicator

Example

`xds_euro_l3_connect(1, 0x4, 'N');` this will send a CONNect message for B-channel 4 on board 1 with no progress indicator.

xds_euro_l3_disconnect

xds_euro_l3_disconnect(board_number,b_channel,cause,progress,reference);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
 for the disconnect
char progress; a character indicating the progress indicator if any
int reference; a value between 0x00 and 0xFF that is the call reference
 number

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 DISConnect message for the call currently associated with the specified B-Channel. A DISConnect message is used to notify either the network or the user that a disconnect sequence has been initiated.

Message Sent

“DDxxccpr” where **xx** is the B-channel number and **cc** is the cause code, **p** is the progress indicator and **rr** is the optional call reference

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_CAUSE the cause value is invalid
ILL_PROGRESS invalid progress indicator value
ILL_FEATURE invalid feature value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a DISConnect message to be sent for the call associated with the specified B-channel. This should be done when initiating a disconnect sequence.

The cause code is a value used to indicate the reason for a particular message, in this case, why a disconnect is occurring. A cause value of 0x10 indicates normal clearing. Other values may indicate error conditions. For a list of cause codes and their meanings, see the appropriate Basic Rate ISDN Board Technical Manual (EuroISDN Version), Q.850 or ETS 300 102.

If a tone or announcement is to be played, the progress should be 'T', otherwise, this should be 'N'. If the call to be released is associated with a B-channel, the call reference specified should be 0x00.

Example

```
xds_euro_13_disconnect(1, 0x4, 0x10, 'N', 0);
```

this will send a DISConnect message for B-channel 4 on board 1, with a cause of 0x10 (normal clearing).

xds_euro_l3_facility

xds_euro_l3_facility(board_number, b_channel, component, reference, id_tag, tag_value)

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char component;	'I', 'E', 'R', or 'J'
int reference;	a value between 0x00 and 0xFF that is the call reference number
int id_tag	a value between 0x00 and 0x7F that is the id tag number
int tag_value	a value between 0x00 and 0xFF that is the operation or error tag value

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 EURO ISDN Facility message.

Message Sent

“DFxxcrrid(tt)” where **xx** is the B-channel number, **c** is the component selected, **rr** is the call reference number, **id** is the id tag number, and **tt** is the operation or error tag number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_ARG	invalid component or ID tag
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a Layer 3 FACILITY message to the board. The legal values for component are: 'I' = invoke, 'E' = return error, 'R' = return results, and 'J' = reject.

Example

```
xds_euro_l3_facility(1, 0x4, 'I', 0x7F, 0x19, 0x11);
```

this will send a FACILITY message for B-channel 4, on board 1, using the invoke 'I' component, a call reference of 0x7F, an id tag value of 0x19, and operation/error tag value of 0x11.

xds_euro_l3_hold

xds_euro_l3_hold(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 HOLD message to a B-channel.

Message Sent

“DHxx” where xx is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a HOLD message to a selected b-channel on a selected board.

Example

xds_euro_l3_hold(1,0x4); this sends a message (B-channel 4 to board 1)

xds_euro_l3_hold_ack

xds_euro_l3_hold_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 HOLD ACKnowledge message to a B-channel.

Message Sent

“DHxxA” where **xx** is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a HOLD ACKnowledge message to a selected b-channel on a selected board.

Example

xds_euro_l3_hold_ack(1, 0x4); this sends a message (B-channel 4 to board 1)

xds_euro_l3_hold_rej

xds_euro_l3_hold_rej(board_number, b_channel, cause);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
for the hold rejection

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 HOLD REJECT message to a B-channel.

Message Sent

“DHxxRcc” where **xx** is the B-channel number, and **cc** is the cause for the rejection

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_CAUSE invalid cause value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a HOLD REJECT message to a selected b-channel on a selected board.

Example

xds_euro_l3_hold_rej(1, 0x4, 0x7F); this sends a message (B-channel 4 to board 1, cause
0x7FF)

xds_euro_l3_info

xds_euro_l3_info(board_number, b_channel, cause);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value for the cause element

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 INFOrmation message for the call currently associated with the specified B-Channel containing a cause element for the call.

Message Sent

“DIxxcc” where **xx** is the B-channel number and **cc** is the cause value

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_CAUSE invalid cause element value

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send an INFOrmation message to be sent for the call associated with the specified B-channel.

Example

xds_euro_l3_info(1, 0x4, 0x1F); this will send an INFOrmation message for B-channel 4 on board 1 with a cause element value of 0x1F.

xds_euro_l3_key_fac

xds_euro_l3_key_fac(board_number, b_channel, cause, key_fac);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
 for the disconnect
char *key_fac a pointer to an ASCII string with any keypad facility digits

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 INFORMATION message with a Keypad Facility element to control supplemental services.

Message Sent

“DIxxKkk” where **xx** is the B-channel number and **kk** are the Keypad digits, or
“DIxxccKkk” where **cc** is a optional cause value.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_CAUSE invalid cause value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a INFORMATION message with a Keypad Facilities element to be sent. An optional cause element may be included if the cause value is non-zero. The Keypad Facilities element is used to control supplemental services which are specific to the particular switch and network. The Keypad Facilities element consists of one or more keypad digits as found on a telephone set keypad, typically in the range 0-9, *, and #.

Example

xds_euro_l3_key_fac(1, 0x4, 0, “2*”); this sends a message (B-channel 4 to board 1 with Keypad Facility string of “2*”, no cause element)

xds_euro_l3_notify

xds_euro_l3_notify(board_number, b_channel, notification);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char notification; a character indicating the notification type

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 NOTIFY message to inform a user or terminal of the change of state of a call due to call rearrangement.

Message Sent

“DNxxn” where **xx** is the B-channel number and **n** is the notification indicator.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_ARG invalid notification value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a NOTIFY message to be sent from an NT port to a user or terminal. The notification is used to inform the terminal of a change of state of a call due to call rearrangement so that the terminal may reflect the change of state by changing the state of an indicator (i.e. controlling whether an indicator blinks or not) or other means of informing the user.

The valid notification indicators are:

R - Call Resumed

S - Call Suspended

Example

xds_euro_l3_notify(1, 0x4, 'R'); this will send a NOTIFY message for B-channel 4 on board
1 with a notification of Call Resumed.

xds_euro_l3_proceed

xds_euro_l3_proceed(board_number, b_channel, progress);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char progress; a character indicating the progress indicator

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 CALL PROCEEDing message for the call currently associated with the specified B-Channel. A CALL PROCEEDing message is used to notify the caller that all information necessary to process a call has been received and that call establishment has been initiated.

Message Sent

“DPxxp” where xx is the B-channel number and p is the progress indicator.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_PROGRESS invalid progress indicator value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a CALL PROCEEDing message to be sent for the call associated with the specified B-channel. This should be done in response to a SETUP message for Enbloc sending, or when all necessary digits have been received for Overlap sending when the call can be initiated.

The progress indicator may take values of:

- C Call not end to end ISDN
- D Destination not ISDN
- I Inband information or appropriate pattern now available
- O Origination not ISDN
- R Return to ISDN
- N No progress indicator

Example

`xds_euro_l3_proceed(1, 0x4, 'C');` this will send a CALL PROCEEDing message for B-channel 4 on board 1, with a progress indicator of call not end to end ISDN.

xds_euro_l3_progress

xds_euro_l3_progress(board_number, b_channel, cause, progress);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x0 and 0xff that indicates the reason for the PROGRESS message
char progress;	a character indicating the progress indicator

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 PROGRESS message for the call currently associated with the specified B-Channel. A PROGRESS message is used to notify the caller of interworking with non-ISDNs, a call is routed to an inband tone or announcement, or when a progress delay at the destination is reported.

Message Sent

“DPxxPccp” where **xx** is the B-channel number, **cc** is the cause value, and **p** is the progress indicator.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_PROGRESS	invalid progress indicator value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a PROGRESS message to be sent for the call associated with the specified B-channel. A PROGRESS message is sent to report an interworking situation such as the destination is non ISDN, or a situation where the call, though initiated can not be completed, such as the called party is busy. In those cases, an inband signal such as busy tone is usually present and indicated by the progress indicator and signal elements in the message. An application would send a PROGRESS message when it realizes that a call can not be completed, or when interworking with the destination switch results in the receipt of a PROGRESS message from the destination.

The cause code is a value used to indicate the reason for the PROGRESS message. Examples would be a cause code of 0x11, 'User busy', or 0x12, 'No user responding'. For a list of cause codes and their meaning, see the appropriate XDS Basic Rate ISDN Board Technical Manual (EuroISDN Version), Q.850, or ETS 300 102.

Progress values that are valid may be:

- C Call is not end-to-end ISDN
- D Destination is non-ISDN
- I Inband information or appropriate pattern now available
- O Origination is non-ISDN
- R Return to ISDN
- N no progress indicator

Example

`xds_euro_l3_progress(1, 0x4, 0x11, 'I');` this will send a PROGRESS message for B-channel 4 on board 1. The cause is 0x11, 'User busy', the progress indicator is inband information.

xds_euro_l3_query_dn

xds_euro_l3_query_dn(board_number, b_channel, dn_info);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
struct dn *dn_info; a structure to contain the DN information
{
char dn[16]; an array to contain the Directory Number
}

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to query the board to obtain the default Directory Number for a specific B-channel.

Message Sent

“DQxx” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
WRONG_BOARD	a query response was received from the wrong board
WRONG_QUERY	the wrong query response was received
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to obtain the default DN (directory number) or subscriber number associated with a specific B-channel. Each B-channel has a default directory or subscriber number that is used as the calling party number for TE ports and the called party number for NT ports in SETUP messages (see **xds_euro_l3_setup**) unless otherwise specified. This number may contain up to 15 digits. The DN is programmed in by the application, and therefore, this command is only of use for diagnostic purposes, i.e. to see that the correct information has been programmed in.

Example

`xds_euro_l3_query_dn(0x1, 0x4, dn_buf)` This would request the DN for the B1 channel of the third port on board 1. The DN would be returned in the structure `dn_buf`.

xds_euro_l3_query_status

xds_euro_l3_query_status(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 STATUS Enquiry message to the board. This can be done to prompt a terminal to respond with a STATUS message.

Message Sent

“DXxx” where xx is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to query an ISDN terminal about the call state of a call in a case where a received message indicates there may be a call processing discrepancy. The terminal, if it responds will send a STATUS message. As this response is dependent on the terminal and not on the board, this message will be returned on the receive message queue and will be retrieved with the **xds_msg_receive** function rather than the **xds_query_receive** function. This function is only valid for B-channels that have a call in process.

Example

`xds_euro_l3_query_status(0x1, 0x4)`

This would send a STATUS ENquiry message for the B1 channel of the third port on board 1.

xds_euro_l3_rel_com

xds_euro_l3_rel_com(board_number, b_channel, cause, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for the release
int reference;	a value between 0x00 and 0xFF that indicates the call reference for the call being released if it is not allocated a B-channel

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 REL_COMplete message for the call currently associated with the specified B-Channel or that is specified by the call reference. A REL_COMplete message is used to release a call reference and terminate a call in cases where a DISConnect message is inappropriate.

Message Sent

“DRxxccrr” where **xx** is the B-channel number **cc** is the cause value and **rr** is the optional call reference.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_REFERENCE	invalid reference value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a REL_COMplete message to be sent for the call associated with the specified B-channel. This should be done to release a call and associated call reference under some circumstances. An example is when no TE equipment has responded to a SETUP message.

If the call has been allocated a B-channel, a DISConnect should be sent to clear a call (see **xds_l3_disconnect**).

The cause code is a value used to indicate the reason for the REL_COMplete message.

Examples would be a cause code of 0x10, 'Normal clearing', or 0x41, 'Bearer capability not supported'. For a list of cause codes and their meaning, see the appropriate XDS Basic Rate ISDN Board Technical Manual (EuroISDN Version), Q.850, or ETS 300 102.

If the call to be released is associated with a B-channel, the call reference specified should be 0x00.

Example

`xds_euro_l3_rel_com(1, 0x4, 0x10, 0x00);` this will send a RELease COMplete message for B-channel 4 on board 1 with a cause value of 0x10, "Normal Clearing" and no call reference.

xds_euro_l3_resume_ack

xds_euro_l3_resume_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 RESUME ACKnowledge message in response to a RESUME message. The call will be allocated the specified B-channel.

Message Sent

“DGxxA” where xx is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a RESUME ACKnowledge message to be sent in response to a RESUME message. The call will be allocated the specified B-channel. Note, that this channel must match that received in the RESUME message. If a RESUME ACKnowledge message is sent, the connection to the call is reestablished on the specified B channel. The Call Identity of the resumed call is then free. Note, that as only a TE (terminal equipment) port can send a RESUME message, only an NT (network termination) port can grant the resumption by sending a RESUME ACKnowledge message.

Example

xds_euro_l3_resume_ack(1, 0x4); this will send a RESUME ACKnowledge message for B-channel 4 on board 1.

xds_euro_l3_resume_rej

xds_euro_l3_resume_rej(board_number, b_channel, cause);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for rejecting the resumptions

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 RESUME REJECT message for the call on the specified B-channel. A RESUME REJECT message is used to deny a RESUME request message.

Message Sent

“DGxxRcc” where **xx** is the B-channel number and **cc** is the cause.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a RESUME REJECT message to be sent for the call on the specified B-channel. The B-channel should match that in the received RESUME message. Note, that as only a TE (terminal equipment) port can send a RESUME message, and only an NT (network termination) port can grant the retrieval by sending a RESUME REJECT message. The cause code is a value used to indicate the reason for the PROGRESS message. Examples would be a cause code of 0x06, ‘Channel unacceptable’, or 0x2A, ‘Switching equipment congestion’. For a list of cause codes and their meaning, see the appropriate XDS Basic Rate ISDN Board Technical Manual (EuroISDN Version), Q.850, or ETS 300 102.

Example

xds_euro_l3_resume_rej(1, 0x4, 0x2A);	this will send a RESUME REJECT message for B-channel 4 on board 1, with a cause of 0x2A, ‘Switching equipment congestion’.
---------------------------------------	--

xds_euro_l3_retrieve

xds_euro_l3_retrieve(board_number, b_channel, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int reference;	a value between 0x0 and 0xFF that indicates the reference number of the call

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 RETrieve message to a B-channel.

Message Sent

“DGxxrr” where **xx** is the B-channel number, and **rr** is the call reference number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_REFERENCE invalid reference value

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a RETrieve message to a selected b-channel on a selected board.

Example

xds_euro_l3_retrieve(1, 0x4, 0x93);	this sends a message (B-channel 4 to board 1 with a call reference # of 0x93)
-------------------------------------	---

xds_euro_l3_retrieve_ack

xds_euro_l3_retrieve_ack(board_number, b_channel, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int reference;	a value between 0x0 and 0xFF that indicates the reference number of the call

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 RETrieve ACKnowledge message to a B-channel.

Message Sent

“DGxxArr” where **xx** is the B-channel number, and **rr** is the call reference number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_REFERENCE invalid reference value

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a RETrieve ACKnowledge message to a selected b-channel on a selected board.

Example

xds_euro_l3_retrieve_ack(1, 0x4, 0x93);	this sends a message (B-channel 4 to board 1 with a call reference # of 0x93)
---	---

xds_euro_l3_retrieve_rej

xds_euro_l3_retrieve_rej(board_number, b_channel, cause, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x0 and 0xFF that indicates the cause value of the rejection
int reference;	a value between 0x0 and 0xFF that indicates the reference number of the call

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 RETrieve REJect message to a B-channel.

Message Sent

“DGxxArr” where **xx** is the B-channel number, and **rr** is the call reference number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_REFERENCE	invalid reference value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a RETrieve REJect message to a selected b-channel on a selected board.

Example

xds_euro_l3_retrieve_ack(1, 0x4, 0x19, 0x93);	this sends a message (B-channel 4 to board 1 with a cause value of 0x19 and a call reference # of 0x93)
---	---

xds_euro_l3_setup

xds_euro_l3_setup(board_number, b_channel, port_type, bearer_cap, progress, complete, key_fac, called_num, calling_num);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char port_type;	a character indicating whether the port is of type NT or TE
char bearer_cap;	a character indicating the type of call, i.e. speech or data
char progress;	a character indicating the progress indicator
char complete;	a value indicating whether sending complete element should be included
char *key_fac	a pointer to an ASCII string with any keypad facility digits
char *called_num;	a pointer to an ASCII string with the called number
char *calling_num;	a pointer to an ASCII string with the calling number

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 SETUP message. A SETUP message is used to initiate a call. As the SETUP message takes a different form for NT and TE interface, the presence of some arguments will depend on the port type.

Message Sent

“DSxxbp(Kk)(calling#)/(called#)”, where **xx** is the B-channel number, **b** is the bearer capability, **p** is the progress indicator, **(Kk)** is the keypad facility element if present, **(calling#)** is the directory number of the calling party and optionally the **(called#)** is the number being called. If the sending complete element is included, the ‘/’ is replaced with a ‘C’.

Note, that for an NT interface, the default directory number is used for the called party number if the called# is omitted and for a TE interface, the default directory number is used for the calling party number if the calling# is omitted.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_TYPE	invalid port type (not NT)
ILL_ARG	an invalid bearer capability value, complete value, or an invalid calling or called number was used
ILL_PROGRESS	invalid progress indicator value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a SETUP message to be sent to initiate a call. The board level command takes a different form depending on whether the port is configured as an NT (network termination) or a TE (terminal equipment). The treatment of the directory numbers is different for the two port types. The port type must be specified by the **port_type** argument. Valid values are 'N' for an NT interface and 'T' for a TE interface.

The XDS BRI Boards uses exclusive B channel allocation only. The B-channel specified for the SETUP message must not be in use at the time the message is sent.

The bearer capability indicates the type of call being made, i.e. voice or data. There are three valid choices:

- A 3.1 kHz. audio (used for voice and modems)
- D unrestricted digital information, 64 kbps
- S speech (suitable for voice, but not modems)

For NT ports, or TE ports where Enbloc sending is used a progress indicator must be specified. This argument is used in interworking situations where the call is not completely ISDN. The valid values are:

- C Call is not end to end ISDN
- O Originating address is non-ISDN
- N No progress indicator

Note that if there are no interworking problems, the progress indicator should be set to 'N'. For TE ports, the called number is the subscriber number of the destination. The calling number element will use the default DN of the B-channel on which the call is being placed. The called number can be up to 15 digits long. If Overlap sending is used, the called number is set to a NULL string.

For NT ports, if the called number is a NULL string, the default DN of the B-channel will be used as the called number element, otherwise, the called number specified will be used. The calling number can be up to 15 digits long. If a called number is specified, the total number of digits in the called and calling number is restricted to 24. Overlap sending is rarely used, so the called number should either be specified or a default number used.

When using Enbloc sending, the Sending Complete element can be included to indicate that the entire called party number is present in the message. This is done by setting the value of complete to 0x1. If it is set to 0x0, the Sending Complete element will not be included.

Some systems may use the Keypad Facility element for supplemental services. Keypad Facility digits are included in the SETUP message by placing them in a string pointed to by key_fac. The digits must be from the set 0-9, #, and *.

Example

For an NT port:

```
xds_euro_l3_setup(1, 0x4, 'N', 'S', 'N', 0x1, '', '', "5551212");
```

this will send a SETUP message for B-channel 4 on board 1 with a bearer capability of speech, no progress indicator, sending complete, no Keypad Facility digits, the default called number and calling number of 555-1212.

For a TE port using Enbloc sending:

```
xds_euro_l3_setup(1, 0x4, 'T', 'D', 'N', 0, '', "18005551212", '');
```

this will send a SETUP message for B-channel 4 on board 1 with a bearer capability of data, no progress indicator, no sending complete, no Keypad Facility digits, the default calling number, and a called number of 1-800-555-1212.

For a TE port using Overlap sending:

```
xds_euro_l3_setup(1, 0x4, 'T', 'A', 'N', 0, '', '', '');
```

this will send a SETUP message for B-channel 4 on board 1 with a bearer capability of 3.1 kHz audio, no progress indicator, no sending complete, no Keypad Facility digits, the default calling number, and no calling number. Subsequent digits would be sent using the function **xds_euro_l3_called_num**.

xds_euro_l3_suspend

xds_euro_l3_suspend(board_number, b_channel, call_id);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char *call_id; a pointer to an ASCII string used to identify the suspended
 call

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 SUSPEND message for the call currently associated with the specified B-Channel. A SUSPEND message is used to temporarily release the B-channel allocated to a call for call rearrangement. A hold can only be initiated by the user or terminal side of an interface.

Message Sent

“DHxxS(id)” where **xx** is the B-channel number, and **id** is the call_id value

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a SUSPEND message to be sent for the call associated with the specified B-channel. This can only be done by the user or terminal side of the interface. If the suspend is accepted, a SUSPEND ACKnowledge message will be sent in response.

A SUSPEND REJect message will be sent in response if the hold is not accepted. This will include the cause for rejection.

A call identity string of up to 10 IA5 characters can be used to identify the call. This string is specified by the application and should be chosen so as to be unique over the user-network on which the user resides. If it is not unique, the call suspension will be rejected. This call identity string is then used when attempting to resume the call.

It should be noted that not all network switches support the SUSPEND feature. On others, it may be available by pre-subscription at the time the interface is ordered. Similarly, terminal equipment such as ISDN station sets may or may not support the feature.

Example

`xds_euro_13_suspend(1, 0x4, 'Call #1');` this will send a SUSPEND message for B-channel 4 on board 1 with a call identity of 'Call #1'

xds_euro_l3_suspend_ack

xds_euro_l3_suspend_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 SUSPEND_ACKnowledge message for the call currently associated with the specified B-Channel. A SUSPEND_ACK message may be used to respond to a SUSPEND message and accept the release of the B-channel while retaining the call.

Message Sent

“DHxxA” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a SUSPEND ACKnowledge message to be sent for the call associated with the specified B-channel. This should be done in response to a SUSPEND message. The B-channel will be released for use with another call. The application should retain the call identity received in the hold message as it will be used when the call is resumed.

Example

xds_euro_l3_suspend_ack(1, 0x4); this will send a SUSPEND ACKnowledge message for B-channel 4 on board 1.

xds_euro_l3_suspend_rej

xds_euro_l3_suspend_rej(board_number, b_channel, cause);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
for the suspend reject

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to send a Layer 3 SUSPEND REJECT message for the call currently associated with the specified B-Channel. A SUSPEND REJECT message may be used to respond to a SUSPEND message and reject the release of the B-channel.

Message Sent

“DHxxRcc” where **xx** is the B-channel number and **cc** is the cause code

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_CAUSE invalid cause value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a SUSPEND REJECT message to be sent for the call associated with the specified B-channel. This should be done in response to a SUSPEND message when the B-channel is not to be released. The cause value should indicate the reason for rejecting the hold. Typical reasons would be 0x45, “requested facility not implemented” if the application does not support call rearrangement or 0x3, “requested facility not subscribed” when the call rearrangement feature is not enabled for a specific interface. For a list of cause codes and their meaning, see the appropriate XDS Basic Rate ISDN Board Technical Manual (EuroISDN Version), Q.850 or the ETS 300 102.

Example

xds_euro_l3_suspend_rej(1, 4, 0x45); this will send a SUSPEND REJECT message for B-channel 4 on board 1 with a cause of 0x45, ‘requested facility not implemented’.

xds_euro_l3_text

xds_euro_l3_text(board_number, b_channel, mode, l3_text);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int mode;	a value between 0x0 and 0x3 indicating the text operation
char *l3_text	a pointer to a NULL terminated string of up to 27 ASCII characters

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to prepare text to be included in a Layer 3 message. Several modes are available that allow inclusion of text into a SETUP message, or provide for sending an INFORMATION message with text. Only NT (network termination) ports can send text.

Message Sent

“DI xx T” where xx is the B-channel number to send a single line of text

“DT xx C” where xx is the B-channel and the text buffer is to be cleared

“DT xx L(text)” where xx is the B-channel and **text** is to be put in the buffer

“DT xx A(text)” where xx is the B-channel and **text** is to be added to the buffer

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_MODE invalid text mode

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to prepare text to send in a Layer 3 message. Display text is held in a buffer of up to 80 ASCII characters. Text to be sent is put in a buffer associated with a B-channel. Text can either be sent as an INFORMATION message, or as part of call control messages such as an ALERTING, a CONNECT, or a SETUP message.

The “mode” controls which text operation is carried out. The valid mode values are:

0x0	clear the buffer
0x1	send the text in the buffer as an INFORMATION message
0x2	place text in the buffer
0x3	add text to the buffer

Each B-channel has its own 80 character buffer. If a SETUP or other message is sent it will include any text in the buffer as an information element. The text buffer is then cleared. Some equipment will only accept a maximum of 32 characters. The text should be ASCII characters. Up to 27 characters may be added to the buffer per function call using modes 2 or 3. It is the responsibility of the application to format the text for readability.

Example

<code>xds_euro_l3_text(1, 0x4, 0, 0);</code>	this will clear the text buffer for B-channel 4 on board 1.
<code>xds_euro_l3_text(1, 0x4, 1, 0);</code>	this will send the text in the buffer as an INFORMATION message
<code>xds_euro_l3_text(1, 0x4, 2, “This is line 1”);</code>	this will place ‘This is line 1’ in the text buffer for B channel 4 on board 1.
<code>xds_euro_l3_text(1, 0x4, 3, “This is more text”);</code>	this will add “This is more text” to the text buffer for B-channel 4 on board 1.

xds_euro_set_bri_dn

xds_euro_set_bri_dn(board_number, b_channel, dn);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char* dn;	a pointer to a character array containing an ASCII string of up to 15 digits with the directory number

Applicable boards

All EuroISDN BRI boards

Purpose

This function is used to set the default DN or directory or subscriber number associated with a particular B-channel.

Message Sent

“SDdddddd” where **dddddd** is the directory or subscriber number for the B-channel.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_ARG	directory number is NULL or too long
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

On a port defined as a network termination or NT, the number is used as the called party number in SETUP messages. If the port is defined as a TE, the number is the calling party in SETUP messages. Typically both B-channels on an interface will have the same directory number. If the directory number is not set with this function, it must be included in the arguments for **xds_euro_l3_setup**.

The directory number can be save in EEAROM using the **xds_set_config** function. If this is done, this and other information will automatically be restored on a board reset. Once set, this function will only be needed if the configuration changes.

Example

`xds_euro_set_bri_dn(1, 0x4, “5551000”);` this will set the DN for B-channel 4 to 555-1000

**Basic Rate ISDN
INS-Net 64
Layer 3 'D' Functions**

This page was intentionally left blank.

xds_net64_l3_alert

xds_net64_l3_alert(board_number, b_channel, progress, signal, feature, indicator);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char progress; a character indicating the progress indicator if any
char signal; a character indicating the signal element if any
int feature; a value between 0 and 63 specifying the feature indicator, if
 any
char ind; a character indicating the state of the feature indicator

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 ALERTing message for the call currently associated with the specified B-Channel. An ALERTing message is used to notify the caller that the called party is being informed of an incoming call.

Message Sent

“DAXx(p(s(ff*i*))” where **xx** is the B-channel number, **p** is the optional progress indicator, **s** is the optional signal, **ff** is the optional feature indicator, and **i** is the indicator state.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_ARG	invalid indicator value
ILL_PROGRESS	invalid progress indicator value
ILL_SIGNAL	invalid signal value
ILL_FEATURE	invalid feature value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an ALERTing message to be sent for the call associated with the specified B-channel. If the port is configured as an NT (network termination), progress indication, signal and feature indication information elements can be sent as part of the message.

Valid values for the optional progress indicator are:

- C Call not end-to-end ISDN
- D Destination not ISDN
- I Inband information or appropriate pattern now available
- N No progress indicator

For supplemental services, the signal and feature indication elements may be added, valid signal values are:

- D Dial tone
- W Call waiting tone
- F Tones off
- A Alerting tone
- N No signal

If the feature indicator value is non-zero, the feature indicator state must be set to a valid value.

Valid values for the feature indication state are:

- D deactivation, indicator off
- A activation, indicator on
- Q prompt, indicator blink slow
- P pending, indicator blink fast

Example

`xds_net64_l3_alert(16, 0x4, 'I', 'F', 5, 'A');` this will send an ALERTing message for B-channel 4 on board 16. The progress indicator is set for inband signal, signal to tones off, feature indicator 5 active.

xds_net64_l3_connect

xds_net64_l3_connect(board_number, b_channel, progress, signal, feature, indicator);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char progress; a character indicating the progress indicator if any
char signal; a character indicating the signal element if any
int feature; a value between 0 and 63 specifying the feature indicator if
any
char ind; a character indicating the state of the feature indicator

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 CONNect message for the call currently associated with the specified B-Channel. A CONNect message is used to notify the caller that the called party has answered an incoming call.

Message Sent

“DCxx(p(s(ffs)))” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_ARG	invalid indicator value
ILL_PROGRESS	invalid progress indicator value
ILL_SIGNAL	invalid signal value
ILL_FEATURE	invalid feature value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a CONNect message to be sent for the call associated with the specified B-channel. This should be done when the called party answers.

Valid values for the progress indicator are:

- C Call not end to end ISDN
- D Destination not ISDN
- I Inband information or appropriate pattern now available
- O Origination not ISDN
- R Return to ISDN
- N No progress indicator

For supplemental services, the signal and feature indication elements may be added, valid signal values are:

- D Dial tone
- W Call waiting tone
- F Tones off
- A Alerting tone
- N No signal

If the feature indicator value is non-zero, the feature indicator state must be set to a valid value. Valid values for the feature indication state are:

- D deactivation, indicator off
- A activation, indicator on
- Q prompt, indicator blink slow
- P pending, indicator blink fast

Example

```
xds_net64_13_connect(1, 0x4, 'N', 'N', 0, 0);
```

this will send a CONNect message for B-channel 4 on board 1 with no progress indicator, signal, or feature indication.

xds_net64_l3_disconnect

xds_net64_l3_disconnect(board_number, b_channel, cause, progress, signal, feature, indicator, charge);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for the disconnect
char progress;	a character indicating the progress indicator if any
char signal;	a character indicating the signal element if any
int feature;	a value between 0 and 63 specifying the feature indicator, if any
char ind;	a character indicating the state of the feature indicator
char *charge;	a pointer to a string of IA5 characters indicating the charge, in Yen

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 DISConnect message for the call currently associated with the specified B-Channel. A DISConnect message is used to notify either the network or the user that a disconnect sequence has been initiated.

Message Sent

“DDxxcc(p(s(ffl)(\$xx))” where **xx** is the B-channel number, **cc** is the cause code, **p** is the optional progress indicator, **s** is the optional signal, **ffl** is the optional feature indicator element, and **\$xx** is the optional advise of charge.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_ARG	invalid indicator value
ILL_PROGRESS	invalid progress indicator value
ILL_SIGNAL	invalid signal value
ILL_FEATURE	invalid feature value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a DISConnect message to be sent for the call associated with the specified B-channel. This should be done when initiating a disconnect sequence.

The cause code is a value used to indicate the reason for a particular message, in this case, why a disconnect is occurring. A cause value of 0x10 indicates normal clearing. Other values may indicate error conditions. For a list of cause codes and their meanings, see the XDS Layer 3 Protocol Software Reference Manual or Q.850.

Valid values for the progress indicator are:

- C Call not end to end ISDN
- D Destination not ISDN
- I Inband information or appropriate pattern now available
- O Origination not ISDN
- R Return to ISDN
- N No progress indicator

For supplemental services, the signal and feature indication elements may be added, valid signal values are:

- D Dial tone
- W Call waiting tone
- F Tones off
- A Alerting tone
- N No signal

If the feature indicator value is non-zero, the feature indicator state must be set to a valid value.

Valid values for the feature indication state are:

- D deactivation, indicator off
- A activation, indicator on
- Q prompt, indicator blink slow
- P pending, indicator blink fast

The optional Advise of Charge is an IA5 character string indicating the charge for the call in Yen.

Example

`xds_net64_13_disconnect(16, 0x4, 0x10, 'N', 'F', 0, 0, '')`; this will send a DISConnect message for B-channel 4 on board 16, with a cause of 0x10 (normal clearing), no progress indicator, a signal of tones off, and no feature indicator or advise of charge.

xds_net64_l3_disc_ref

xds_net64_l3_disc_ref(board_number, b_channel, cause, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for the disconnect
int reference;	a value between 0x00 and 0xFF that indicates the call reference for the call being released if it is not allocated a B-channel

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 DISConnect message for the call currently associated with the specified B-Channel. A DISConnect message is used to notify either the network or the user that a disconnect sequence has been initiated.

Message Sent

“DDxxccRrr” where **xx** is the B-channel number and **cc** is the cause code and **rr** is the call reference.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_REFERENCE	invalid reference value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a DISConnect message to be sent for the call associated with the specified B-channel. This should be done when initiating a disconnect sequence.

The cause code is a value used to indicate the reason for a particular message, in this case, why a disconnect is occurring. A cause value of 0x10 indicates normal clearing. Other values may indicate error conditions. For a list of cause codes and their meanings, see the XDS Layer 3 Protocol Software Reference Manual or Q.850.

Example

```
xds_net64_13_disconnect(1, 0x4, 0x10, 0x12);
```

this will send a DISConnect message for B-channel 4 on board 1, with a cause of 0x10 (normal clearing) and call reference of 0x12.

xds_net64_l3_feature_act

xds_net64_l3_feature_ind(board_number, b_channel, feature, block_id);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int feature; a value between 0 and 63 specifying the feature activation,
 if any
char block_id; a character between 0 and 3 identifying the blocked channel

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 INFORMATION message with a Feature Activation element to control supplemental services.

Message Sent

“DFxxff(Bb)” where **xx** is the B-channel number, **ff** is the feature activation number, and **Bb** is the optional Blocking channel ID.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_ARG invalid blocking channel ID value
ILL_FEATURE invalid feature value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a INFORMATION message with a Feature Activation element to be sent from a User port (TE). The Feature Activation element is used to control supplemental services, which are specific to the particular switch and network.

The optional Blocking Channel ID element is used to indicate that a B-channel is not available to receive incoming calls. Valid values are ‘0’ if no channel blocks, ‘1’ if the B1 channel is blocked, ‘2’ if the B2 channel is blocked, and ‘3’ if both channels block. If a 0x0, no element is included.

Example

xds_net64_l3_feature_act(16, 0x4, 5, 0); this will send an INFORMATION message for B-channel 4 on board 16 with a feature activation code of 5 and no blocking id element.

xds_net64_l3_feature_ind

xds_net64_l3_feature_ind(board_number, b_channel, feature, indicator, block_id, charge);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int feature; a value between 0 and 63 specifying the feature indicator, if
any
char ind; a character indicating the state of the feature indicator
char block_id; a character between 0 and 3 identifying the blocked channel
char *charge; a pointer to a string of IA5 characters indicating the charge,
in Yen

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 INFORMATION message with a Feature Indication element to control supplemental services. An optional blocking channel ID or advise of charge element may be included.

Message Sent

“DFxxffi(Bb)” or “DFxxffi(\$xx)” where **xx** is the B-channel number, **ffi** is the feature indication and indicator status, **(Bb)** is the optional blocking channel id and **(\$xx)** is the advise of charge string.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_ARG	invalid indicator, or blocking channel ID value
ILL_FEATURE	invalid feature value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a INFORMATION message with a Feature Indication element to be sent. The Feature Indication element is used to control supplemental services which are specific to the particular switch and network. If the feature indicator value is non-zero, the feature indicator state must be set to a valid value. Valid values for the feature indication state are:

- D deactivation, indicator off
- A activation, indicator on
- Q prompt, indicator blink slow
- P pending, indicator blink fast

Either the Blocking Channel ID or Advise of Charge element may optionally be included in the message. The optional Blocking Channel ID element is used to indicate that a B-channel is not available to receive incoming calls. Valid values are '0' if no channel blocks, '1' if the B1 channel is blocked, '2' if the B2 channel is blocked, and '3' if both channels block. If a 0x0, no element is included. The Advise of Charge is used to indicate an amount in Yen that is the charge for the previous call.

Example

```
xds_net64_13_feature_ind(16, 0x4, 5, 'P', 0, "$237");
```

this will send an INFORMATION message for B-channel 4 on board 16 with a feature indication element 5 set to prompt, and an advise of charge of 237 Yen.

xds_net64_l3_hold

xds_net64_l3_hold(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 HOLD message to a B-channel.

Message Sent

“DHxx” where xx is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a HOLD message to a selected b-channel on a selected board.

Example

xds_net64_l3_hold(1, 0x4); this sends a message (B-channel 4 to board 1)

xds_net64_l3_hold_ack

xds_net64_l3_hold_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 HOLD ACKnowledge message to a B-channel.

Message Sent

“DHxxA” where **xx** is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a HOLD ACKnowledge message to a selected b-channel on a selected board.

Example

xds_net64_l3_hold_ack(1, 0x4); this sends a message (B-channel 4 to board 1)

xds_net64_l3_hold_rej

xds_net64_l3_hold_rej(board_number, b_channel, cause);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
for the hold rejection

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 HOLD REJECT message to a B-channel.

Message Sent

“DHxxRcc” where **xx** is the B-channel number, and **cc** is the cause for the rejection

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_CAUSE invalid cause value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a HOLD REJECT message to a selected b-channel on a selected board.

Example

xds_net64_l3_hold_rej(1, 0x4, 0x7F); this sends a message (B-channel 4 to board 1, cause
0x7F)

xds_net64_l3_info

xds_net64_l3_info(board_number, b_channel, cause, key_fac, feature, indicator, block_id, charge);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for the disconnect
char *key_fac	a pointer to an ASCII string with any keypad facility digits
int feature;	a value between 0 and 63 specifying the feature indicator, if any
char ind;	a character indicating the state of the feature indicator
char block_id;	a character between 0 and 3 identifying the blocked channel
char *charge;	a pointer to a string of IA5 characters indicating the charge, in Yen

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 INFORMATION message from the network with optional cause, Keypad Facility, feature indication, blocking channel id or advise of charge elements.

Message Sent

“DIxx(cc)(Kkk)(ffi)(Bb)(\$xx)” where **xx** is the B-channel number, **cc** is the optional cause, **Kkk** are the optional Keypad digit(s), **ffi** is the optional feature indication, and **Bb** or **\$xx** are the optional blocking channel id or advise of charge elements.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_ARG	invalid indicator value
ILL_FEATURE	invalid feature value
ILL_CAUSE	invalid cause value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an INFORMATION message to be sent from a port set to the network type. An optional cause element may be included if the cause value is non-zero. The Keypad Facilities or feature indication elements may be included to control supplemental services, which are specific to the particular switch and network. The Keypad Facilities element consists of one or more keypad digits as found on a telephone set keypad, typically in the range 0-9, *, and #.

If the feature indicator value is non-zero, the feature indicator state must be set to a valid value. Valid values for the feature indication state are:

D	deactivation, indicator off
A	activation, indicator on
Q	prompt, indicator blink slow
P	pending, indicator blink fast

Either the Blocking Channel ID or Advise of Charge element may optionally be included in the message. The optional Blocking Channel ID element is used to indicate that a B-channel is not available to receive incoming calls. Valid values are '0' if no channel blocks, '1' if the B1 channel is blocked, '2' if the B2 channel is blocked, and '3' if both channels block. If a 0x0, no element is included. The Advise of Charge is used to indicate an amount in Yen that is the charge for the previous call.

Example

```
xds_net64_13_info(1, 0x4, 0, '2*', 0, 0, 0, '');
```

this will send an INFORMATION message for B-channel 4 on board 1 with Keypad Facility string of '2*' and no cause, feature indication, blocking channel id or advise of charge elements

xds_net64_l3_info_text

xds_net64_l3_info_text(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 INFORMATION message with the display text continued in the buffer for that channel.

Message Sent

“DIxxT” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a INFORMATION message with a Display element consisting of the text in the buffer for the specified channel to be sent. Text may be placed in the buffer using the **xds_net64_l3_text** function.

Example

xds_net64_l3_info_text(1, 0x4); this will send an INFORMATION message for B-channel 4
on board 16 with a Display element consisting of the text in
the buffer.

xds_net64_l3_notify

xds_net64_l3_notify(board_number, b_channel, notification);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char notification; a character indicating the notification type

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 NOTIFY message to inform a user or terminal of the change of state of a call due to call rearrangement.

Message Sent

“DNxxn” where **xx** is the B-channel number and **n** is the notification indicator.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_ARG invalid notification indicator value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a NOTIFY message to be sent from an NT port to a user or terminal. The notification is used to inform the terminal of a change of state of a call due to call rearrangement so that the terminal may reflect the change of state by changing the state of an indicator (i.e. controlling whether an indicator blinks or not) or other means of informing the user.

The valid notification indicators are:

R - Call Resumed
S - Call Suspended

Example

xds_net64_l3_notify(1, 0x4, 'R'); this will send a NOTIFY message for B-channel 4 on board 1 with a notification of Call Resumed.

xds_net64_l3_proceed

xds_net64_l3_proceed(board_number, b_channel, progress, signal, feature, indicator);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char progress; a character indicating the progress indicator if any
char signal; a character indicating the signal element if any
int feature; a value between 0 and 63 specifying the feature indicator, if
any
char ind; a character indicating the state of the feature indicator

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 CALL PROCEEDing message for the call currently associated with the specified B-Channel. A CALL PROCEEDing message is used to notify the caller that all information necessary to process a call has been received and that call establishment has been initiated.

Message Sent

“DPxx(p(s(ff))” where **xx** is the B-channel number, **p** is the progress indicator, **s** is the optional signal, and **ff** is the optional feature indication.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_ARG	invalid indicator value
ILL_FEATURE	invalid signal indicator value
ILL_SIGNAL	invalid progress indicator value
ILL_PROGRESS	invalid progress indicator value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a CALL PROCEEDing message to be sent for the call associated with the specified B-channel. This should be done in response to a SETUP message for Enbloc sending, or when all necessary digits have been received for Overlap sending when the call can be initiated.

The progress indicator may take values of:

C	Call not end to end ISDN
D	Destination not ISDN
I	Inband information or appropriate pattern now available
O	Origination not ISDN
R	Return to ISDN
N	No progress indicator

For supplemental services, the signal and feature indication elements may be added, valid signal values are:

D	Dial tone
W	Call waiting tone
F	Tones off
A	Alerting tone
N	No signal

If the feature indicator value is non-zero, the feature indicator state must be set to a valid value. Valid values for the feature indication state are:

D	deactivation, indicator off
A	activation, indicator on
Q	prompt, indicator blink slow
P	pending, indicator blink fast

Example

```
xds_net64_13_proceed(16, 0x4, 'D', 'F', 0, 0);
```

this will send a CALL PROCEEDing message for B-channel 4 on board 1, with a progress indicator of destination not end ISDN, signal of tones off, and no feature indication.

xds_net64_l3_progress

xds_net64_l3_progress(board_number, b_channel, cause, progress, signal);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x0 and 0xff that indicates the reason for
 the PROGRESS message
char progress; a character indicating the progress indicator, if any
char signal; a character indicating the signal element, if any

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 PROGRESS message for the call currently associated with the specified B-Channel. A PROGRESS message is used to notify the caller of interworking with non-ISDNs, a call is routed to an inband tone or announcement, or when a progress delay at the destination is reported.

Message Sent

“DPxxPccp(s)” where **xx** is the B-channel number, **cc** is the cause value, **p** is the progress indicator, and **s** is the optional signal

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_SIGNAL	invalid progress indicator value
ILL_PROGRESS	invalid progress indicator value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a PROGRESS message to be sent for the call associated with the specified B-channel. A PROGRESS message is sent to report an interworking situation such as the destination is non-ISDN, or a situation where the call, though initiated can not be completed, such as the called party is busy. In those cases, an inband signal such as busy tone is usually present and indicated by the progress indicator and signal elements in the message. An application would send a PROGRESS message when it realizes that a call can not be completed, or when interworking with the destination switch results in the receipt of a PROGRESS message from the destination.

The cause code is a value used to indicate the reason for the PROGRESS message. Examples would be a cause code of 0x11, "User busy", or 0x12, "No user responding". For a list of cause codes and their meaning, see the XDS Layer 3 Protocol Software Reference Manual or Q.850. Progress values that are valid may be:

C	Call is not end to end ISDN
D	Destination is non-ISDN
I	Inband information or appropriate pattern now available
O	Origination is non-ISDN
R	Return to ISDN
N	no progress indicator

For supplemental services, the signal and feature indication elements may be added, valid signal values are:

D	Dial tone
W	Call waiting tone
F	Tones off
A	Alerting tone
N	No signal

Example

```
xds_net64_l3_progress(1, 0x4, 0x11, 'I', 'N');
```

this will send a PROGRESS message for B-channel 4 on board 1. The cause is 0x11, "User busy", the progress indicator is inband information, and no signal.

xds_net64_l3_query_status

xds_net64_l3_query_status(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 STATUS Enquiry message to the board. This can be done to prompt a terminal to respond with a STATUS message.

Message Sent

“DXxx” where xx is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to query an ISDN terminal about the call state of a call in a case where a received message indicates there may be a call processing discrepancy. The terminal, if it responds will send a STATUS message. As this response is dependent on the terminal and not on the board, this message will be returned on the receive message queue and will be retrieved with the **xds_msg_receive** function rather than the **xds_query_receive** function. This function is only valid for B-channels that have a call in process.

Example

`xds_net64_l3_query_status(0x1, 0x4)`

This would send a STATUS ENquiry message for the B1 channel of the third port on board 1.

xds_net64_l3_rel_com

xds_net64_l3_rel_com(board_number, b_channel, cause, signal, feature, indicator, charge);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
 for the release
char signal; a character indicating the signal element, if any
int feature; a value between 0 and 63 specifying the feature indicator, if
 any
char ind; a character indicating the state of the feature indicator
char charge; a pointer to a string of IA5 characters indicating the charge,
 in Yen

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 RELease or REL_COMplete message for the call currently associated with the specified B-Channel. A RELease or REL_COMplete message is used to release a call reference and terminate a call in cases where a DISConnect message is inappropriate.

Message Sent

“DRxxcc(s(ffs)(\$xx)” where **xx** is the B-channel number, **cc** is the cause value, **ff** is the feature number, and **rr** is the optional call reference.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_SIGNAL	invalid progress indicator value
ILL_FEATURE	invalid feature value
ILL_ARG	invalid indicator value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a RELease REL_COMplete message to be sent for the call associated with the specified B-channel. The message type will depend on the call state of the call. This should be done to release a call and associated call reference under some circumstances. An example is when no TE equipment has responded to a SETUP message. If the call has been allocated a B-channel, a DISConnect should be sent to clear a call (see **xds_net64_l3_disconnect**).

The cause code is a value used to indicate the reason for the REL_COMplete message. Examples would be a cause code of 0x10, “Normal clearing”, or 0x41, “Bearer capability not supported”. For a list of cause codes and their meaning, see the XDS Layer 3 Protocol Software Reference Manual or Q.850. The signal element is used to indicate the presence or absence of inband signals and should be ‘F’ for tones of or ‘N’. The Feature Indication element is used for supplemental services. The advise of charge element consists of a string of IA5 characters indicating the call charge in Yen.

Example

`xds_net64_l3_rel_com(16, 0x4, 0x10, 'N', 0, '');` this will send a RELease COMplete message for B-channel 4 on board 16 with a cause value of 0x10, “Normal Clearing” and no signal, feature indication or advise of charge element.

xds_net64_l3_rel_com_ref

xds_net64_l3_rel_com_ref(board_number, b_channel, cause, reference);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
 for the release
int reference; a value between 0x00 and 0xFF that indicates the call
 reference for the call being released

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 REL_COMplete message for the call specified by the call reference. A REL_COMplete message is used to release a call reference and terminate a call in cases where a DISConnect message is inappropriate.

Message Sent

“DRxxccRrr” where **xx** is the B-channel number **cc** is the cause value and **rr** is the call reference.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_CAUSE invalid cause value
ILL_REFERENCE invalid call reference value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a REL_COMplete message to be sent for the call associated with the specified B-channel. This should be done to release a call and associated call reference under some circumstances. An example is when no TE equipment has responded to a SETUP message.

If the call has been allocated a B-channel, a DISConnect should be sent to clear a call (see **xds_net64_l3_disconnect**).

The cause code is a value used to indicate the reason for the message. Examples would be a cause code of 0x10, "Normal clearing", or 0x41, "Bearer capability not supported". For a list of cause codes and their meaning, see the XDS Layer 3 Protocol Software Reference Manual or Q.850.

Example

`xds_net64_l3_rel_com_ref(16, 0x4, 0x10, 0x12);` this will send a RELEase COMplete message for B-channel 4 on board 16 with a cause value of 0x10, "Normal Clearing" and a call reference of 0x12.

xds_net64_l3_resume

xds_net64_l3_resume(board_number, b_channel, call_id);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char *call_id; a pointer to an IA5 string used to identify the suspended
call

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 RESUME message for the call specified by the call identity. A RESUME message is used to reestablish a connection to a suspended call.

Message Sent

“DGxxS(id)” where **xx** is the B-channel number and **id** is the call identity string.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_ARG invalid calling number
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a RESUME message to be sent for the call with the specified call identity. The call will be allocated the specified B-channel. This is done when resuming a call previously suspended using a SUSPEND message. The selected B-channel must be available. The NT side should respond with either a RESUME ACKnowledge message, in which case, the connection is restored, or a RESUME REJect message, in which case, the call remains in the suspended state.

The call identity used in this message is the one used in the original SUSPEND message. It is the responsibility of the application to retain this call identity as long as the call is in the suspended state. Failure to do so may result in unretrievable or unreleasable calls.

Example

xds_net64_l3_resume(16, 0x4, “Call #1”); this will send a RESUME message for B-channel 4
on board 16 with a call identity of “Call #1”

xds_net64_l3_resume_ack

xds_net64_l3_resume_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 RESUME ACKnowledge message in response to a RESUME message. The call will be allocated the specified B-channel.

Message Sent

“DGxxA” where xx is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a RESUME ACKnowledge message to be sent in response to a RESUME message. The call will be allocated the specified B-channel. Note, that this channel must match that received in the RESUME message. If a RESUME ACKnowledge message is sent, the connection to the call is reestablished on the specified B channel. The Call Identity of the resumed call is then free. Note, that as only a TE (terminal equipment) port can send a RESUME message, only an NT (network termination) port can grant the resumption by sending a RESUME ACKnowledge message.

Example

xds_net64_l3_resume_ack(16, 0x4); this will send a RESUME ACKnowledge message
for B-channel 4 on board 16.

xds_net64_l3_resume_rej

xds_net64_l3_resume_rej(board_number, b_channel, cause, charge);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
 for rejecting the resumption
char *charge; a pointer to a string of IA5 characters indicating the charge,
 in Yen

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 RESUME REJECT message for the call on the specified B-channel. A RESUME REJECT message is used to deny a RESUME request message.

Message Sent

“DGxxRcc(\$xx)” where **xx** is the B-channel number and **cc** is the cause, and **\$xx** is the optional advise for charge.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_CAUSE invalid cause value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a RESUME REJect message to be sent for the call on the specified B-channel. The B-channel should match that in the received RESUME message. Note, that as only a TE (terminal equipment) port can send a RESUME message, and only an NT (network termination) port can grant the retrieval by sending a RESUME REJect message. The cause code is a value used to indicate the reason for the PROGress message. Examples would be a cause code of 0x06, “Channel unacceptable”, or 0x2A, “Switching equipment congestion”. For a list of cause codes and their meaning, see the XDS Layer 3 Protocol Software Reference Manual or Q.850.

An optional advise of charge element can be included which gives the charge for the call in yen.

Example

```
xds_net64_13_resume_rej(16, 0x4, 0x2A);
```

this will send a RESUME REJect message for B-channel 4 on board 16, with a cause of 0x2A, “Switching equipment congestion”.

xds_net64_l3_retrieve

xds_net64_l3_retrieve(board_number, b_channel, reference);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int reference; a value between 0x0 and 0xFF that indicates the reference
 number of the call

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 RETrieve message to a B-channel.

Message Sent

“DGxxrr” where **xx** is the B-channel number, and **rr** is the call reference number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_REFERENCE invalid call reference value

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a RETrieve message to a selected b-channel on a selected board.

Example

xds_net64_l3_retrieve(1, 0x4, 0x93); this sends a message (B-channel 4 to board 1 with a
 call reference # of 0x93)

xds_net64_l3_retrieve_ack

xds_net64_l3_retrieve_ack(board_number, b_channel, reference);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int reference; a value between 0x0 and 0xFF that indicates the reference
 number of the call

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 RETrieve ACKnowledge message to a B-channel.

Message Sent

“DGxxArr” where **xx** is the B-channel number, and **rr** is the call reference number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_REFERENCE invalid call reference value

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a RETrieve ACKnowledge message to a selected b-channel on a selected board.

Example

xds_net64_l3_retrieve_ack(1, 0x4, 0x93); this sends a message (B-channel 4 to board 1
 with a call reference # of 0x93)

xds_net64_l3_retrieve_rej

xds_net64_l3_retrieve_rej(board_number, b_channel, cause, reference);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x0 and 0xFF that indicates the cause value of the rejection
int reference;	a value between 0x0 and 0xFF that indicates the reference number of the call

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 RETrieve REJect message to a B-channel.

Message Sent

“DGxxArr” where **xx** is the B-channel number, and **rr** is the call reference number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause value
ILL_REFERENCE	invalid call reference value
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a RETrieve REJect message to a selected b-channel on a selected board.

Example

xds_net64_l3_retrieve_ack(1, 0x4, 0x19, 0x93);	this sends a message (B-channel 4 to board 1 with a cause value of 0x19 and a call reference # of 0x93)
--	---

xds_net64_l3_setup

xds_net64_l3_setup(board_number, b_channel, port_type, bearer_cap, progress, key_fac, called_num, calling_num);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char port_type;	a character indicating whether the port is of type NT or TE
char bearer_cap;	a character indicating the type of call, i.e. speech or data
char progress;	a character indicating the progress indicator
char *key_fac	a pointer to an ASCII string with any keypad facility digits
char *called_num;	a pointer to an ASCII string with the called number
char *calling_num;	a pointer to an ASCII string with the calling number

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 SETUP message. A SETUP message is used to initiate a call. As the SETUP message takes a different form for NT and TE interface, the presence of some arguments will depend on the port type.

Message Sent

“DSxxbp(Kk)(calling#)/(called#)”, where **xx** is the B-channel number, **b** is the bearer capability, **p** is the progress indicator, (**Kk**) is the keypad facility element if present, (**calling#**) is the directory number of the calling party and optionally the (**called#**) is the number being called.

Note, that for an NT interface, the default directory number is used for the called party number if the **called#** is omitted and for a TE interface, the default directory number is used for the calling party number if the **calling#** is omitted.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_PROGRESS	invalid progress indicator value
ILL_TYPE	invalid port type
ILL_ARG	invalid bearer capability value, calling number, or called number
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a SETUP message to be sent to initiate a call. The board level command takes a different form depending on whether the port is configured as an NT (network termination) or a TE (terminal equipment). The treatment of the directory numbers is different for the two port types. The port type must be specified by the **port_type** argument. Valid values are 'N' for an NT interface and 'T' for a TE interface.

The XDS BRI Boards uses exclusive B channel allocation only. The B-channel specified for the SETUP message must not be in use at the time the message is sent.

The bearer capability indicates the type of call being made, i.e. voice or data. There are three valid choices:

- A 3.1 kHz. audio (used for voice and modems)
- D unrestricted digital information, 64 kbps
- S speech (suitable for voice, but not modems)

For NT ports, or TE ports where Enbloc sending is used a progress indicator must be specified. This argument is used in interworking situations where the call is not completely ISDN. The valid values are:

- C Call is not end to end ISDN
- O Originating address is non-ISDN
- N No progress indicator

Note that if there are no interworking problems, the progress indicator should be set to 'N'.

For TE ports, the called number is the subscriber number of the destination. The calling number element will use the default DN of the B-channel on which the call is being placed. The called number can be up to 15 digits long.

For NT ports, if the called number is a NULL string, the default DN of the B-channel will be used as the called number element, otherwise, the called number specified will be used. The calling number can be up to 15 digits long. If a called number is specified, the total number of digits in the called and calling number is restricted to 24.

Note that in INS-Net 64, overlap sending is not used, so the called number should either be specified or a default number used.

Some systems may use the Keypad Facility element for supplemental services. Keypad Facility digits are included in the SETUP message by placing the in a string pointed to by key_fac. The digits must be from the set 0-9, #, and *.

Example

For an NT port:

```
xds_net64_l3_setup(1, 0x4, 'N', 'S', 'N', ',', '5551212');
```

this will send a SETUP message for B-channel 4 on board 1 with a bearer capability of speech, no progress indicator, sending complete, no Keypad Facility digits, the default called number and calling number of 555-1212.

For a TE port using Enbloc sending:

```
xds_net64_l3_setup(1, 0x4, 'T', 'D', 'N', ',', "18005551212", ',');
```

this will send a SETUP message for B-channel 4 on board 1 with a bearer capability of data, no progress indicator, no Keypad Facility digits, the default calling number, and a called number of 1-800-555-1212.

xds_net64_l3_suspend

xds_net64_l3_suspend(board_number, b_channel, call_id);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char *call_id; a pointer to an ASCII string used to identify the suspended
 call

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 SUSPEND message for the call currently associated with the specified B-Channel. A SUSPEND message is used to temporarily release the B-channel allocated to a call for call rearrangement. A hold can only be initiated by the user or terminal side of an interface.

Message Sent

“DHxx” where xx is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a SUSPEND message to be sent for the call associated with the specified B-channel. This can only be done by the user or terminal side of the interface. If the suspend is accepted, a SUSPEND ACKnowledge message will be sent in response.

A SUSPEND REJect message will be sent in response if the hold is not accepted. This will include the cause for rejection.

A call identity string of up to 10 IA5 characters can be used to identify the call. This string is specified by the application and should be chosen so as to be unique over the user-network on which the user resides. If it is not unique, the call suspension will be rejected. This call identity string is then used when attempting to resume the call.

It should be noted that not all network switches support the SUSPEND feature. On others, it may be available by presubscription at the time the interface is ordered. Similarly, terminal equipment such as ISDN station sets may or may not support the feature.

Example

`xds_ins64_13_suspend(16, 0x4, "Call #1");` this will send a SUSPEND message for B-channel 4 on board 16 with a call identity of "Call #1".

xds_net64_l3_suspend_ack

xds_net64_l3_suspend_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 SUSPEND_ACKnowledge message for the call currently associated with the specified B-Channel. A SUSPEND_ACK message may be used to respond to a SUSPEND message and accept the release of the B-channel while retaining the call.

Message Sent

“DHxxA” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a SUSPEND ACKnowledge message to be sent for the call associated with the specified B-channel. This should be done in response to a SUSPEND message. The B-channel will be released for use with another call. The application should retain the call identity received in the hold message as it will be used when the call is resumed.

Example

xds_net64_l3_suspend_ack(16, 0x4); this will send a SUSPEND ACKnowledge message for B-channel 4 on board 16.

xds_net64_l3_suspend_rej

xds_net64_l3_suspend_rej(board_number, b_channel, cause);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
for the suspend reject

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 SUSPEND REJECT message for the call currently associated with the specified B-Channel. A SUSPEND REJECT message may be used to respond to a SUSPEND message and reject the release of the B-channel.

Message Sent

“DHxxRcc” where **xx** is the B-channel number and **cc** is the cause code

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range
ILL_CAUSE invalid cause value
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a SUSPEND REJECT message to be sent for the call associated with the specified B-channel. This should be done in response to a SUSPEND message when the B-channel is not to be released. The cause value should indicate the reason for rejecting the hold. Typical reasons would be 0x45, “requested facility not implemented” if the application does not support call rearrangement or 0x3, “requested facility not subscribed” when the call rearrangement feature is not enabled for a specific interface. For a list of cause codes and their meaning, see the XDS Layer 3 Protocol Software Reference Manual or Q.850.

Example

xds_net64_l3_suspend_rej(16, 4, 0x45); this will send a SUSPEND REJECT message for B-channel 4 on board 16 with a cause of 0x45.

xds_net64_l3_text

xds_net64_l3_text(board_number, b_channel, mode, l3_text);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int mode;	a value between 0x0 and 0x3 indicating the text operation
char *l3_text;	a pointer to a NULL terminated string of up to 27 IA5 characters

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to prepare text to be included in a Layer 3 message. Several modes are available that allow inclusion of text into a SETUP message, or provide for sending an INFORMATION message with text. Only NT (network termination) ports can send text.

Message Sent

“DI xx T” where xx is the B-channel number to send a single line of text

“DT xx C” where xx is the B-channel and the text buffer is to be cleared

“DT xx L(text)” where xx is the B-channel and **text** is to be put in the buffer

“DT xx A(text)” where xx is the B-channel and **text** is to be added to the buffer

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the B-channel is out of valid range

ILL_MODE invalid text mode

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to prepare text to send in a Layer 3 message. Display text is held in a buffer of up to 32 ASCII characters. Text to be sent is put in a buffer associated with a B-channel. Text can either be sent as an INFORMATION message, or as part of call control messages such as an ALERTING, a CONNECT, or a SETUP message.

The “mode” controls which text operation is carried out. The valid mode values are:

0x0	clear the buffer
0x1	send the text in the buffer as an INFORMATION message
0x2	place text in the buffer
0x3	add text to the buffer

Each B-channel has its own 32 character buffer. If a SETUP or other message is sent it will include any text in the buffer as an information element. The text buffer is then cleared. The text should be ASCII characters. Up to 27 characters may be added to the buffer per function call using modes 2 or 3. It is the responsibility of the application to format the text for readability.

Example

<code>xds_net64_13_text(16, 0x4, 0, 0);</code>	this will clear the text buffer for B-channel 4 on board 16.
<code>xds_net64_13_text(16, 0x4, 1, 0);</code>	this will send the text in the buffer as an INFORMATION message
<code>xds_net64_13_text(16, 0x4, 2, “This is line 1”);</code>	this will place “This is line 1” in the text buffer for B channel 4 on board 16.
<code>xds_net64_13_text(16, 0x4, 3, “This is more text”);</code>	this will add “This is more text” to the text buffer for B-channel 4 on board 16.

xds_net64_l3_user_alert

xds_net64_l3_user_alert(board_number, b_channel, progress, feature);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char progress;	a character indicating the progress indicator if any
int feature;	a value between 0 and 63 specifying the feature activation if any

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 ALERTing message for the call currently associated with the specified B-Channel. An ALERTing message is used to notify the caller that the called party is being informed of an incoming call.

Message Sent

“DAXxpf” where **xx** is the B-channel number, **p** is the optional progress indicator and **ff** is the optional feature activation.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_PROGRESS	invalid progress indicator mode
ILL_FEATURE	invalid feature indicator mode
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an ALERTing message to be sent from a user port (TE) for the call associated with the specified B-channel. Progress indication and feature activation information elements can be sent as part of the message.

Valid values for the optional progress indicator are:

- C Call not end-to-end ISDN
- D Destination not ISDN
- N No progress indicator

For supplemental services the feature activation element may be added.

Example

`xds_net64_13_user_alert(16, 0x4, 'D', 5,);` this will send an ALERTing message for B-channel 4 on board 16. The progress indicator is set for destination not ISDN, feature activation 5.

xds_net64_l3_user_info

xds_net64_l3_user_info(board_number, b_channel, cause, key_fac, feature, block_id);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int cause;	a value between 0x00 and 0xFF that indicates the reason for the disconnect
char *key_fac	a pointer to an ASCII string with any keypad facility digits
int feature;	a value between 0 and 63 specifying the feature activation, if any
char block_id;	a character between 0 and 3 identifying the blocked channel

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 INFORMATION message from a user (terminal). Optional cause, Keypad Facility, Feature, of Blocking Channel ID elements may be included.

Message Sent

“DIxx(cc)(Kkk)(ff)(Bb)” where **xx** is the B-channel number, **cc** is the optional cause indicator, **Kkk** is the optional Keypad Facility digits, **ff** is the optional feature activation, and **Bb** is the optional blocking channel id.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_PORT	the B-channel is out of valid range
ILL_CAUSE	invalid cause mode
ILL_ARG	invalid blocking ID value
ILL_FEATURE	invalid feature indicator mode
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause an INFOrmation message to be sent from a user port (TE) for the call associated with the specified B-channel. Optional Cause and Blocking channel ID elements may be included. Keypad Facility and Feature activation information elements may also be sent as part of the message to control supplemental services.

The cause code is a value used to indicate the reason for the message. Examples would be a cause code of 0x10, “Normal clearing”, or 0x41, “Bearer capability not supported”. For a list of cause codes and their meaning, see the XDS Layer 3 Protocol Software Reference Manual or Q.850.

The optional Blocking Channel ID element is used to indicate that a B-channel is not available to receive incoming calls. Valid values are ‘0’ if no channel blocks, ‘1’ if the B1 channel is blocked, ‘2’ if the B2 channel is blocked, and ‘3’ if both channels block. If a 0x0, no element is included

Example

```
xds_net64_l3_user_info(16, 0x4, 0, '2', 0, 0);
```

this will send an INFOrmation message for B-channel 4 on board 16. A Keypad Facility digit of ‘2’ is included. No cause, feature, or blocking channel id is included.

xds_net64_l3_user_proceed

xds_net64_l3_user_proceed(board_number, b_channel, progress, feature);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char progress;	a character indicating the progress indicator, if any
int feature;	a value between 0 and 63 specifying the feature activation, if any

Applicable boards

XDS INS-Net 64 BRI boards

Purpose

This function is used to send a Layer 3 CALL PROCEEDing message for the call currently associated with the specified B-Channel. A CALL PROCEEDing message is used to notify the caller that all information necessary to process a call has been received and that call establishment has been initiated.

Message Sent

“DPxxp(ff)” where **xx** is the B-channel number, **p** is the progress indicator, and **ff** is the optional feature activation.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	the B-channel is out of valid range
ILL_PROGRESS	invalid progress mode
ILL_FEATURE	invalid feature indicator mode
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a CALL PROCEEDing message to be sent from a user port (TE) for the call associated with the specified B-channel. This should be done in response to a SETUP message for Enbloc sending, or when an interworking situation exists.

The progress indicator may take values of:

- C Call not end to end ISDN
- D Destination not ISDN
- O Origination not ISDN
- N No progress indicator

Example

`xds_net64_13_user_proceed(1, 0x4, 'D', 0);` this will send a CALL PROCEEDing message for B-channel 4 on board 1, with a progress indicator of destination not ISDN, and no feature activation.

This page was intentionally left blank.

**Basic Rate ISDN
AT&T Custom
Layer 3 'D' Functions**

This page was intentionally left blank.

xds_att_l3_assoc

int xds_att_l3_assoc(board_number, b_channel, reference, type, call_app)

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
int reference;	a value between 0x0 and 0x1F indicating the B-channel
char type;	the call type, valid characters are 'S', 'C', 'H', 'R', 'E', 'D', 'X', or 'S'
int call_app;	a value between 0 and 254 the Call Appearance number

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

The purpose of this function is to send an ASSOCIated type message. This message is used by the network to indicate to a user that activity is taking place on an associated user endpoint. It is only valid when the port is acting as the network.

Message Sent

“DBxxGrrt”, where **xx** is the b-channel, **rr** is the reference number, **t** is the type. If the call appearance number is not 0, that number will be appended to the end of the message.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_TYPE	an invalid association type was selected
ILL_ARG	an invalid call appearance was called
ILL_PORT	an invalid B-channel value was called for the board being used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

In the AT&T Custom Protocol, several users can be associated in a group that share common call appearances. When a call is presented to a group it appears at that call appearance for each member of the group. When an activity takes place such as one of the members of the group answering the call, the other users of the group are informed of this by an ASSOCIATED message. The associated type element in the message indicates which type of activity has taken place. The state of the call will be indicated by the LEDs for that particular call appearance.

The first message to establish an associated call will be an ASSOCIATED message with an associated type of Setup. A terminal will respond with an ASSOCIATED ACKNOWLEDGE message (see **xds_att_13_assoc_ack**). When sending an ASSOCIATED message with an associated type of SETUP, the application should always use a call reference of 0. The ASSOCIATED ACKNOWLEDGE message returned by the terminal will contain that call reference that should be used in all subsequent messages. For more information on the use of ASSOCIATED messages see *The XDS Layer 3 Protocol Manual 251M031*.

Example

`xds_att_13_assoc(16, 0x11, 0x19, 'S', 3);` this will send an ASSOCIATED type message to board 16, b-channel 11, with a call reference number of 19, a call type 'S', and a call appearance of 3

xds_att_l3_assoc_ack

int xds_att_l3_assoc_ack(board_number, b_channel, reference)

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int reference; a value between 0x0 and 0x1F indicating the B-channel

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

The purpose of this function is to send an ASSOCIated ACKnowledge type message. This message is used by the network to ACKnowledge to a user that activity is taking place on an associated user endpoint. It is only valid when the port is acting as the network. The ASSOCIated ACKnowledge message returned by the terminal will contain that call reference that should be used in all subsequent messages.

Message Sent

“DBxxGrrA”, where **xx** is the b-channel, **rr** is the reference number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The first message to establish an associated call will be an ASSOCIated message with an associated type of Setup. The ASSOCIated ACKnowledge message returned by the terminal will contain that call reference that should be used in all subsequent messages. For more information on the use of ASSOCIated messages see *The XDS Layer 3 Protocol Manual 251M031*.

Example

xds_att_l3_assoc_ack(16, 0x12, 0x18); this will send and ASSOCIated type message to
board 16, b-channel 12, with a call reference
number of 18

xds_att_l3_call_appearance

int xds_att_l3_call_appearance(board_number, b_channel, call_app, adj_ctnl)

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int call_app; a value indicating the call appearance
int adj_ctnl; a value of 'N', 'F', or 0

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to select the active call appearance on a terminal using the AT&T Custom Protocol. It is only valid when the port is acting as the network.

Message Sent

“DBxxA=caa”, where **xx** is the b-channel, **ca** is the call_app, and the **a** is the adj_ctnl.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG an invalid adjunct control value was called
ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

ISDN Station sets using the AT&T Custom Protocol have a number of call appearances which correspond to button/indicator pairs on the set. The active call appearance is indicated by an LED associated with that call appearance. Some sets have one LED which indicates both the selected call appearance and the state of the call associated with that appearance while others have two LEDs, one for selected call appearance and the other for call state. Typically, the LED will be red when the call appearance is selected.

If the user wishes to change the call appearance, an AT&T Feature Key message will be sent with the feature key corresponding to the call appearance number. The network, if it accepts the change will send an INFO message with the new selected call appearance. This will change the selected call appearance LED. Any activity on the station set such as going off-hook, hanging up, or placing a call on hold will now be for the call associated with the selected call appearance.

The adjunct control element may be used to direct the station set to go on or off hook. As an example, when responding to an AT&T TRANSFER message from a user, the network may send an AT&T HOLD message followed by an INFO message with a new call appearance number (using a vacant call appearance) and the adjunct control set to off-hook. The station set will automatically respond by sending a SETUP message using the new call appearance number. The adjunct control element is optional, and can be "F" for off-hook, "N" for on-hook, or NUL if it is not to be included.

Example

```
xds_bri_13_call_appearance(16, 0x12, 2, 'F');
```

this will display the call appearance for board 16, b-channel 12, with a call appearance number of 2, as being "OFF-HOOK"

xds_att_l3_conference

int xds_att_l3_conference(board_number, b_channel)

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

The purpose of this function is to send a conference message to the specified b-channel.

Message Sent

“DBxxC”, where xx is the b-channel

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

For more information on the use of the conference messages see *The XDS Layer 3 Protocol Manual 251M031*.

Example

xds_att_l3_conference(16, 0x11); this will send a layer 3 conference message to
board 16, b-channel 11

xds_att_l3_conference_ack

int xds_att_l3_conference_ack(board_number, b_channel)

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

The purpose of this function is to send a conference acknowledge message to the specified b-channel.

Message Sent

“DBxxCA”, where xx is the b-channel

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

For more information on the use of conference messages see *The XDS Layer 3 Protocol Manual 251M031*.

Example

xds_att_l3_conference_ack(16, 0x11); this will send a layer 3 conference acknowledge message to board 16, b-channel 11

xds_att_l3_conference_rjc

int xds_att_l3_conference_rjc(board_number, b_channel, cause)

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0 and 0xFF of the cause

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

The purpose of this function is to send a conference reject message to the specified b-channel.

Message Sent

“DBxxCRcc”, where **xx** is the b-channel, and **cc** is the cause value

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_CAUSE an invalid rejection cause value was used
ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

For more information on the use of conference messages see *The XDS Layer 3 Protocol Manual 251M031*.

Example

xds_att_l3_conference_rjc(16, 0x11, 0xFF); this will send a layer 3 conference reject message to board 16, b-channel 11, with a cause of 0xFF

xds_att_l3_drop

int xds_att_l3_drop(board_number, b_channel)

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send an AT&T DROP message. This message is used by a user to request that the last call added to a conference be removed. For a two party call, the DROP message requests that the call be disconnected. This function is only valid when the port is acting as a terminal or user.

Message Sent

“DBxxD”, where xx is the b-channel

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The AT&T DROP message is used by a station set to request that the last party added to a conference be removed from the conference. If the call only involves two parties, the request is for a disconnect. The call or conference is that associated with the currently selected call appearance for the terminal. It is not necessary to specify this call appearance in the message.

The network will respond with a DROP ACKnowledge or DROP REJect message. If only two parties are involved, it will respond with a DISConnect.

Example

```
xds_att_l3_drop(16, 0x11);                        this will drop b-channel 11 on board 16 from a  
                                                  connection or conference
```

xds_att_l3_drop_ack

int xds_att_l3_drop_ack(board_number, b_channel)

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 DROP ACKnowledge message to a B-channel.

Message Sent

“DBxxDA”, where **xx** is the b-channel

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a DROP ACKnowledge message to a selected b-channel on a selected board.

Example

xds_att_l3_drop_ack(16, 0x11); this will send the drop ACK message to b-channel
11 on board 16

xds_att_l3_drop_rej

int xds_att_l3_drop_rej(board_number, b_channel)

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 DROP REJECT message to a B-channel.

Message Sent

“DBxxDR”, where xx is the b-channel

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a DROP REJECT message to a selected b-channel on a selected board.

Example

xds_att_l3_drop_rej(16, 0x11);

this will send the drop REJECT message to b-channel
11 on board 16

xds_att_l3_feature_ind

xds_att_l3_feature_ind(board_number, b_channel, button, module, type, feature, *status)
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char button; the character 'U', 'C', 'F', or 'N', indicating the button type
char module; a value between 0 and 7, indicating the module mode to use
char type; the call type, valid characters are 'N', 'B', 'S', or 'M'
int feature; a value between 0 and 255 specifying the feature indicator
char *status; a pointer to a character, between 0 and 7, indicating the
indicator status (if any)

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 INFORMATION message with a Feature Indication element to control supplemental services. An optional indicator status may be included.

Message Sent

“DBxxFbmsffi”, where **xx** is the B-channel number, **b** is the button type on the phone, **m** is the module mode used, **s** indicates the status type, **ff** is the feature number used, and **i** is the optional indicator status

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_TYPE	an invalid call type was selected
ILL_FEATURE	an invalid feature indicator value was passed
ILL_ARG	an invalid button type or module mode was used
ILL_PORT	an invalid B-channel value was called for the board being used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a INFORMATION message with a Feature Indication element to be sent. The Feature Indication element is used to control supplemental services, which are specific to a particular switch and network.

Valid values for the button type are:

U	unknown
C	call appearance button
F	feature button
N	non applicable

Valid values for the status type are:

N	feature number status
B	button number status
S	multiple button status
M	maintenance status

Valid values for the feature indicator status are:

0	active
1	deactivated
2	slow blink
3	medium blink
4	fast blink
5	solid, then off
6	fast blink, then off
7	pulse, then off

Example

```
xds_att_13_feature_ind(16, 0x4, 'N', 1, 'B', 43);
```

this will send an INFORMATION message for B-channel 4, on board 16, with a button type of 'N', module mode of 1, a status type of 'B', feature number 43, with no status message.

xds_att_l3_feature_key

xds_att_l3_feature_key(board_number, b_channel, feature)

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int feature; a value between 0 and 63 specifying the feature number

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 INFORMATION message with a Feature number to control supplemental services.

Message Sent

“DBxxFff”, where **xx** is the B-channel number and **ff** is the feature number used

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_FEATURE an invalid feature key was used
ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to cause a INFORMATION message with a Feature number to be sent. The Feature number is used to control supplemental services which are specific to the particular switch and network.

Example

xds_att_l3_feature_key(16, 0x4, 63); this will send an INFORMATION
message for B-channel 4, on board
16, with a feature number of 63.

xds_att_l3_hold

xds_att_l3_hold(board_sw1, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 HOLD message to a B-channel.

Message Sent

“DBxxH” where **xx** is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a HOLD message to a selected b-channel on a selected board.

Example

xds_att_l3_hold_ack(1, 0x4); this sends a HOLD message to B-channel 4 on board 1

xds_att_l3_hold_ack

xds_att_l3_hold_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 HOLD ACKnowledge message to a B-channel.

Message Sent

“DBxxHA” where **xx** is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a HOLD ACKnowledge message to a selected b-channel on a selected board.

Example

xds_att_l3_hold_ack(1, 0x4); this sends a HOLD ACKnowledge message to B-channel 4
on board 1

xds_att_l3_hold_rej

xds_att_l3_hold_rej(board_number, b_channel, cause);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0x00 and 0xFF that indicates the reason
for the hold rejection

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 HOLD REJECT message to a B-channel.

Message Sent

“DBxxHRcc” where **xx** is the B-channel number, and **cc** is the cause for the rejection

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_CAUSE an invalid hold rejection cause value was passed
ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to send a HOLD REJECT message to a selected b-channel on a selected board.

Example

xds_att_l3_hold_rej(1, 0x4, 0x7F); this sends a HOLD REJECT message to B-channel 4
on board 1, with a cause of 0x7F

xds_att_l3_mim_spid

xds_att_l3_mim_spid(board_number, b_channel, *spid);

unsigned char board_number;	a value indicating which board the command is for
int b_channel;	a value indicating the B-channel to control
char *spid;	a pointer to a character array containing an ASCII string with the SPID (TE interfaces only) the SPID may be up to 14 digits

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This message is used by a terminal to send the Service Profile Identity (SPID) to identify the terminal and its capabilities to the network.

Message Sent

“DBxxMS” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	an invalid B-channel value was called for the board being used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

In the AT&T Custom Protocol a Management Information Message (MIM) is used to transfer the Service Profile Identity (SPID) rather than as an information element in an INFOrmation message. This may be done by a terminal either as part of the initialization procedure or in response to a MIM request from the network. Such a request would be indicated by a message of the form DBxxMS where **xx** is the b-channel when received by a port acting as a terminal. If the **spid** argument in the function call is a NULL string, the default SPID programmed on the board will be used. Otherwise, the argument string will be used. SPIDs are between 10 and 14 digits long depending on the switch. Typically, switches using the AT&T Custom protocol use a 10 digit SPID.

Example

xds_att_l3_mim_spid(1, 0x4, “608555200001”); this sends the message to B-channel 4 on board 1, with a SPID of “608555200001”

xds_att_l3_reconnect

xds_att_l3_reconnect(board_number, b_channel, reference);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int reference; the call reference number

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 reconnect message to a B-channel that is on hold.

Message Sent

“DBxxRrr” where **xx** is the B-channel number, and **rr** is the call reference number used

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function is used to reconnect a channel that is on hold.

Example

xds_att_l3_reconnect(1, 0x4, 0x7F); this sends a reconnect message to B-channel 4 on
board 1, with a call reference number of 0x7F

xds_att_l3_reconnect_ack

xds_att_l3_reconnect_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 reconnect ACKnowledge message to a B-channel.

Message Sent

“DBxxRA” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function is used to send a reconnect ACKnowledge message.

Example

xds_att_l3_reconnect_ack(1, 0x4); this sends a reconnect ACKnowledge message to B-channel 4 on board 1

xds_att_l3_reconnect_rej

xds_att_l3_reconnect_rej(board_number, b_channel, cause);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; the cause number of the rejection

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 reconnect message to a B-channel that is on hold.

Message Sent

“DBxxRRrr” where **xx** is the B-channel number, and **rr** is the call reference number used

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_CAUSE an invalid hold rejection cause value was passed
ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function is used to send a reconnect REJect message, with the cause.

Example

xds_att_l3_reconnect_rej(1, 0x4, 0x7F); this sends a reconnect REJ message to B-channel 4
on board 1, with a call reference number of 0x7F

xds_att_l3_redirect

xds_att_l3_redirect(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used to send a Layer 3 redirect message to a B-channel that is on hold.

Message Sent

“DBxxK” where **xx** is the B-channel number

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function is used to reconnect a channel that is on hold.

Example

xds_att_l3_redirect(1, 0x4); this sends a redirect message to B-channel 4 on
board 1

xds_att_l3_switchhook

xds_att_l3_switchhook(board_number, b_channel, hook_state);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
char hook_state; the hook state value to be used, either 'N' or 'F'

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is used by a terminal to inform the network of the switch-hook status of the terminal. It is valid only for ports acting as a terminal or user.

Message Sent

“DBxxSs” where **xx** is the B-channel number, and **s** is the hook state chosen

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG an invalid hook state value was called
ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

In the AT&T Protocol, an INFOrmation message is used by a terminal to indicate whether the terminal is on-hook or off-hook. This is done as part of the initialization procedure and after a call has been completed and the terminal returns to the idle state.

The hook-state argument can be either 'F' for off-hook, or 'N' for on-hook.

Example

xds_att_l3_switchhook(1, 0x4, 'N'); this reports B-channel 4 on board 1 being ON-HOOK

xds_att_l3_text

xds_att_l3_text(board_number, b_channel, call_app, text_b, text_c);
unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int call_app; a value between 0x0 and 0xFF indicating the text operation
char *text_b a pointer to a NULL terminated string of up to 12 ASCII
 characters
char *text_c a pointer to a NULL terminated string of up to 12 ASCII
 characters

Applicable boards

XDS BRI Boards

Purpose

This function is used to prepare text to be included in a Layer 3 message. Several modes are available that allow inclusion of text into a SETUP message, or provide for sending an INFOrmation message with text. Only NT (network termination) ports can send text.

Message Sent

“DBxxAaa” where **xx** is the B-channel, and **aa** is the call appearance

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG

an invalid call appearance value or text string was used

IOCTL

an IOCTL was called and returns a return value from the IOCTL call

Comments

The display on the 8510T station set is broken up into three areas, a two character call appearance, a twelve character area for displaying a number, and a third area for displaying text. non-standard Codeset 6 information element that uses this format. The element includes coding to specify which of the three areas the text is to be displayed in. To accommodate this format, the XDS Layer 3 message to send text to a terminal from an NT port uses the form DBxxXaabbbbbbbbbbbcccccccccc where xx is the B-channel, aa is the call appearance, bbbbbbbbbbbb is a 12 character field for the number, and ccccccccccc is any additional text.

The number field must be padded with spaces to be exactly twelve characters long. As an example, the message:

Example

```
xds_l3_text(1, 0x4, 11, "6085551000", "Caller One");
```

this will display for B-channel 4 on board 1: the call appearance of 11, the number "6085551000" and "Caller One".

xds_att_l3_transfer

xds_att_l3_transfer(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for

int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is to send a layer 3 transfer to a b-channel.

Message Sent

“DBxxT” where xx is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used during the call transfer stage of a call.

Example

xds_att_l3_tranfer(1, 0x4); this send a transfer message to B-channel 4 on
board 1

xds_att_l3_transfer_ack

xds_att_l3_transfer_ack(board_number, b_channel);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is to send a layer 3 transfer ACKnowledge message to a b-channel.

Message Sent

“DBxxTA” where **xx** is the B-channel number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to ACKnowledge a transfer being made.

Example

xds_att_l3_tranfer_ack(1, 0x4); this sends a transfer ACKnowledge message to B-channel 4 on board 1

xds_att_l3_transfer_rej

xds_att_l3_transfer_rej(board_number, b_channel, cause);

unsigned char board_number; a value indicating which board the command is for
int b_channel; a value indicating the B-channel to control
int cause; a value between 0 and 0xFF for the REJection cause

Applicable boards

AT&T Custom XDS BRI ISDN Boards

Purpose

This function is to send a layer 3 transfer REJect message to a b-channel.

Message Sent

“DBxxTRcc” where **xx** is the B-channel number and **cc** is the cause for the REJection.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_CAUSE an invalid hold rejection cause value was passed
ILL_PORT an invalid B-channel value was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command is used to REJect a transfer from being made.

Example

xds_att_l3_tranfer_rej(1, 0x4, 0xFF); this sends a transfer REJect message to B-channel 4
on board 1, with a cause of 255

XDS PCI-based MC3 Board Functions

This page was intentionally left blank.

xds_mc3_ring_mode

xds_mc3_ring_mode(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
char mode; a value between 0 and 6 indicating the which mode the
MC3 fiber should operate in

Applicable boards

Infinity Series H.100/H.110 MC3 boards

Purpose

This function is used to set the operating/ring mode for the MC3 fiber rings.

Message Sent

“SM0” and “SR0” for mode 0.
“SM1” and “SR0” for mode 1.
“SM2” and “SR0” for mode 2.
“SM2” and “SR1” for mode 3.
“SR2” for mode 4.
“SR1” for mode 5.
“SM1” and “SR2” for mode 6.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_ARG an invalid mode was chosen

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The **xds_mc3_ring_mode** is used to set the operating mode of the MC3 fiber rings. Three modes are supported. Mode 0 is the extended mode. Both rings are available providing a total of 4846 timeslots. In modes 1-6 the rings operate redundantly and 2423 timeslots are available.

The modes are:

0 - extended mode

1 - redundant, ring 0 primary

2 - redundant, ring 1 primary

3 - redundant, ring 0 primary, ring 0 failed, ring 1 becomes primary

4 - redundant, ring 0 primary, ring 1 failed

5 - redundant, ring 1 primary, ring 0 failed

6 - redundant, ring 1 primary, ring 1 failed, ring 0 becomes primary

Example

```
xds_mc3_ring_mode(16, 1);
```

this will set board 16 to operate in mode 1.

xds_ct_listen

xds_ct_listen(board_number, ct_xmt_timeslot, mc3_rcv_timeslot);

unsigned char board_number; a value indicating which board the command is for
int ct_xmt_timeslot; a value between 0 and 4095 for the output H.100/H.110 timeslot
int mc3_rcv_timeslot; a value between 0 and 4845 indicating which timeslot on the MC3 bus is serving as an input

Applicable boards

Infinity Series H.100/H.110 MC3 boards

Purpose

This function creates a one-way audio connection from an MC3 bus timeslot to a H.100/H.110 timeslot.

Message Sent

“XLl0000zzzz” where **0000** is the H.100/H.110 timeslot and **zzzz** is the MC3 bus timeslot.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_SLOT an invalid timeslot was chosen
WRONG_BOARD an invalid board type was chosen
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function performs essentially half of an **xds_mc3_connect**. Any of the 4846 MC3 bus timeslots may be connected to a port on the H.100/H.110 using this function. If the MC3 bus is configured as redundant rings, only 2423 timeslots are available. See **xds_mc3_connect** for details of the MC3 bus.

Example

xds_ct_listen(16, 2, 1535); this creates an audio path from timeslot 1535 on the MC3 bus to timeslot 2 on the H.100/H.110 bus using board 16.

xds_ct_pattern

xds_ct_pattern(board_number, ct_xmt_timeslot, pattern);

unsigned char board_number; a value indicating which board the command is for
int ct_xmt_timeslot; a value between 0 and 4095 for the output H.100/H.110
 timeslot
unsigned char pattern; a value between 0x0 and 0xFF for the pattern

Applicable boards

Infinity Series H.100/H.110 MC3 boards

Purpose

This function may be used to output a fixed pattern on the H.100/H.110 bus. This may be used for diagnostic purposes or to place “silence” or some other pattern on a particular timeslot.

Message Sent

“XPIxxxxpp” where **xxxx** is the port number and **pp** is the pattern value.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_ARG an invalid pattern was chosen
ILL_SLOT an invalid timeslot was chosen
WRONG_BOARD an invalid board type was chosen
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function will cause the pattern value to be output to the H.100/H.110 timeslot, which is defined by the base timeslot set for the board at initialization and the port number defined by the ct_xmt_timeslot parameter. Placing a pattern on a timeslot may be used as a diagnostic tool to check bus integrity. It may also be used to place a “silence” pattern of 0xFF on the H.100/H.110 bus. A pattern may be disabled by using the **xds_ct_rls** function.

Example

xds_ct_pattern(16, 2, 0xFF); this outputs the pattern 0xFF on the timeslot reserved for timeslot 2 on board 16.

xds_ct_rls

xds_ct_rls(board_number, ct_xmt_timeslot);

unsigned char board_number;	a value indicating which board the command is for
int ct_xmt_timeslot;	a value between 0 and 4095 for the output H.100/H.110 timeslot

Applicable boards

Infinity Series H.100/H.110 MC3 boards

Purpose

This function will release a connection between the MC3 bus and the H.100/H.110 bus and disable the output on the H.100/H.110 bus timeslot reserved for the port to transmit on. This function will break down only half of a full duplex connection.

Message Sent

“XDIoooo” where **oooo** is the H.100/H.110 timeslot number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_SLOT	an invalid timeslot was chosen
WRONG_BOARD	an invalid board type was chosen
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

The **xds_ct_rls** function releases a connection from the MC3 bus to the H.100/H.110 bus that has been established using the **xds_mc3_connect**, **xds_mc3_listen**, or **xds_ct_pattern** functions. To release a connection from the H.100/H.110 bus to the MC3 bus, the **xds_mc3_rls** function must be used. Therefore, to release a full duplex connection both the **xds_ct_rls** and **xds_mc3_rls** functions must be used.

Example

<code>xds_ct_rls(16, 2);</code>	this will disable the H.100 output from timeslot 2 on board 16.
---------------------------------	---

Comments

The MC3 bus consists of two fiber optic rings, each of which has a capacity of 2423 timeslots. When operated in the extended mode, a total of 4846 timeslots are available. When operated in one of the redundant modes, only 2423 timeslots are available.

The **xds_mc3_connect** function is used to create a full-duplex connection between the local H.100/H.110 bus and the MC3 bus. The H.100/H.110 MC3 board at the other end of the connection must have a complementary connection established. The choice of MC3 bus timeslots is up to the application to select appropriate streams for transmitting on and to choose the specific timeslots to be used for each connection.

Example

```
xds_mc3_connect(16, 2, 103, 512, 1535);
```

this creates a connection between timeslot 1535 on the MC3 bus and timeslot 2 on the H.100/H.110 bus, and between timeslot 103 on the H.100/H.110 bus and timeslot 512 on the MC3 bus.

xds_mc3_loopback_mode

xds_mc3_loopback_mode(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
char mode; a value between 0 and 0xF indicating the loopback mode
for the MC3 fiber

Applicable boards

Infinity Series H.100/H.110 MC3 boards

Purpose

This function is used to set the loopback mode for the MC3 fiber rings. This can be used for diagnostic or testing purposes.

Message Sent

“SLx” where x is the mode in hexadecimal.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG an invalid mode was chosen

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The **xds_mc3_loopback_mode** is used to set the loopback mode of the MC3 fiber rings. Available modes are from 0 to 15. This can be used for testing purposes. Each of four bits in the mode value control one of the loopback functions.

These bits are:

0 TLBB - Terminal Loopback B

1 FLBB - Facilities Loopback B

2 TLBA - Terminal Loopback A

3 FLBA - Facilities Loopback A

In normal operation, all bits should be 0. For more details on the loopback operation see board's hardware reference manual.

Example

xds_mc3_loopback_mode(16, 1); this will enable TLBB on board 16.

xds_mc3_2k_tone

xds_mc3_2k_tone_det(board_number, ct_rcv_timeslot, mode);

unsigned char board_number; a value indicating which board the command is for
int ct_rcv_timeslot; a value between 0 and 4095 for the H.100/H.110 timeslot
char mode; a value indicating whether to enable or disable generation

Applicable boards

Infinity Series H.100/H.110 MC3 boards

Purpose

This function will either enable or disable SS7 2 kHz tone generation on a H.100/H.110 timeslot.

Message Sent

“CTooooE” where **oooo** is the H.100/H.110 timeslot that the tone will be outputted on.

“CTooooD” where **oooo** is the H.100/H.110 timeslot to disable the tone on.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_SLOT invalid timeslot

WRONG_BOARD invalid board type

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The generator should not be enabled if any of the 2 kHz. detectors are enabled as this may result in a possible CCA assignment conflict. If a conflict occurs, an error message of the form **EG01** will be reported. The generator may be disabled with a command of the form **CTooooD** where **oooo** is the H.100/H.110 bus stream and timeslot. In practice, it is best to enable the generator at start up time and leave it connected to the H.100/H.110 bus.

Example

xds_mc3_2k_tone(16, 512, 'E');

this will enable 2 kHz tone generation on timeslot 512 of board 16

xds_mc3_2k_tone(16, 512, 'D');

this will disable 2 kHz tone generation on timeslot 512 of board 16

xds_mc3_2k_tone_det

xds_mc3_2k_tone_det(board_number, ct_rcv_timeslot, mode);

unsigned char board_number; a value indicating which board the command is for
int ct_rcv_timeslot; a value between 0 and 4095 for the H.100/H.110 timeslot
char mode; a value indicating whether to enable or disable detection

Applicable boards

Infinity Series H.100/H.110 MC3 boards

Purpose

This function will either enable or disable SS7 2 kHz tone detection on a given timeslot.

Message Sent

“CLiiii” where **iiii** is the H.100/H.110 timeslot to enable detection on.

“CLiiiiD” where **iiii** is the H.100/H.110 timeslot to disable detection on.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition.

ILL_SLOT invalid timeslot
WRONG_BOARD invalid board type
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

Signaling System 7 (SS7) uses a 2 kHz. tone for performing continuity checks to verify the operation of speech circuits. The H.100/H.110 MC3/Conferencing Board is capable of generating and detecting this tone. A single generator is provided to play a 2 kHz. tone to an H.100/H.110 bus timeslot. From there, it may be routed to multiple MC3 bus timeslots. Up to 128 detectors are available for the detection of the 2 kHz tones. The DSP resources for 2 kHz. detection are shared with those used for DTMF detection and clamping of conferenced inputs. Conferencing **must** be enabled for the 2 kHz. detection and generation to be available. Note, that at most, 128 detectors, generators and conference parties can be assigned at a time.

Example

xds_mc3_2k_tone_det(16, 512, 'E'); this will enable detection on timeslot 512, on
board 16
xds_mc3_2k_tone_det(16, 512, 'D'); this will disable detection on timeslot 512,
on board 16

This page was intentionally left blank.

H.100/110 Line Board Common Functions

This page was intentionally left blank.

xds_line_ct_listen

int xds_line_ct_listen(board_number, port, stream, timeslot);

unsigned char board_number;	a value indicating which board the command is for
int port;	a value indicating the port to use for the listen
int stream;	a value indicating the H.100 stream to use for the listen
int timeslot;	a value indicating the H.100 timeslot to use for the listen

Applicable Boards

XDS Infinity H.100 & H.110 line-interface boards

Purpose

This function will create a one-way audio interface between the CT bus timeslot and the port on the indicated board.

Message Sent

“CAxxsstt” where **xx** is the port, **ss** is the CT bus stream, and **tt** is the CT bus timeslot.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	an interface port was called for the board being used
ILL_SLOT	an invalid CT bus stream and/or timeslot was called
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This function will create a one-way audio interface between the CT bus timeslot and the port on the indicated board.

Example

xds_line_ct_listen(16, 0, 2, 1); directs timeslot 1 of stream 2 to port 0 on board 16

xds_line_ct_connect

```
int xds_line_ct_connect(board_number, port, output_stream, output_timeslot
                        input_stream, input_timeslot);
```

unsigned char board_number;	a value indicating which board the command is for
int port;	a value indicating the port to connect
int output_stream;	the output stream to connect
int output_timeslot;	the output timeslot to connect
int input_stream;	the input stream to connect
int input_timeslot;	the input timeslot to connect

Applicable Boards

XDS Infinity H.100 & H.110 line-interface boards

Purpose

This function is used to connect an input stream and timeslot with an output stream and timeslot to a port interface.

Message Sent

“CCxxssttaabb” where **xx** is the port, **ss** is the output stream, **tt** is the out put timeslot, **aa** is the input stream, and **bb** is the input timeslot

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	an invalid B-channel value was called for the board being used
ILL_SLOT	an invalid stream or timeslot was called for the board being used
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

The valid range of streams is from 0 to 31, while the valid range of timeslots is 0 to 127.

Example

xds_line_ct_connect(16, 11, 0, 3, 10, 50);	connects input stream 0 timeslot 3 and output stream 10 timeslot 50 to port 11, on board 16
--	---

xds_line_ct_play_call_progress

int xds_line_ct_play_call_progress(board_number, tone, stream, timeslot);

unsigned char board_number; a value indicating which board the command is for
int tone; a value between 0 and 5 indicating which call progress tone
 to play
int stream; a value indicating which CT bus stream to use
int timeslot; a value indicating which CT bus timeslot to use

Applicable Boards

XDS Infinity H.100 & H.110 line-interface boards

Purpose

This function can be used to play any of the six precise call progress tones to a port. These tones are dial tone, reorder, busy, and audible ringback. In addition, silence and a calibration tone of 1004 Hz. can be played at a level corresponding to 1 milliwatt.

Message Sent

“CS2xsstt” where **x** is the tone number, **ss** is the CT bus stream, and **tt** is the CT bus timeslot.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_SLOT an illegal stream and/or timeslot value was passed
ILL_MODE an invalid call progress tone was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comment

There is no limit on the number of ports listening to any of the call progress tones.

The six tones produced by this function are selected with the tone argument. The tones produced are:

- 0 - dial tone
- 1 - reorder tone
- 2 - busy tone
- 3 - audible ringback
- 4 - digital milliwatt 1004 Hz steady tone.
- 5 - silence

Example

```
xds_line_ct_play_call_progress(16, 5, 3, 10);
```

this will play “silence” to stream 3, timeslot 5 of board 16

xds_line_ct_generate_customtone

xds_line_ct_generate_customtone(board_number, port, frequency1, frequency2, level1 level2, on_dur, off_dur, reps);

unsigned char board_number;	a value indicating which board the command is for
int port;	a value between 0 and the maximum number of ports on the board indicating which port should receive the tone
int frequency1;	a value between 0 and 0x0CFF (0 to 3327 Hz) for the frequency of the first tone (in Hz), a value of 0 disables the tone
int frequency2;	a value between 0 and 0x0CFF (0 to 3327 Hz) for the frequency of the second tone (in Hz), a value of 0 disables the tone
int level1;	a value between 0 and 0x3F giving the level in -dBm of the first tone
int level2;	a value between 0 and 0x3F giving the level in -dBm of the second tone
int on_dur;	a value between 0 and 0xFF giving the duration in 50 msec. increments of the on portion of the tone
int off_dur;	a value between 0 and 0xFF giving the duration in 50 msec. increments of the off portion of the tone
int reps;	a value between 1 and 255 giving the number of times the tone repeats

Applicable Boards

XDS Infinity H.100 & H.110 line-interface boards

Purpose

This function can be used to generate a custom tone for user alerts or other purposes. The tone can be made up of either a single frequency or dual frequencies. The duration of both the on and off portion of the tone can be controlled, as well as the number of repetitions.

Message Sent

“CVxxfffflfffflInnffrr” where **xx** is the port, **ffff** is the frequency of the first and second tones, **ll** is the level of the first and second tones, **nn** is the on duration, **ff** is the off duration, and **rr** is the number of repetitions.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	an interface port was called for the board being used
ILL_ARG	an out of range value was used for one of the arguments
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This function uses the DTMF generators to generate the custom tone.

The custom tones produced with this function can consist of one or two frequencies. The frequencies should be within the telephone frequency range of 0 to 3327 Hz. If only a single frequency is desired, the other frequency should have a value of 0. The level of the frequencies can be specified in a range of 0 dBm to -62 dBm, with a larger level value producing a quieter tone. If the frequency is not to be used, the level value should be set to 63.

The duration of the ON portion of the tone is given by on_dur in 50 msec. increments in a range of 50 msec to 12700 milliseconds. A value of 0xFF for the duration variable will play a continuous tone.

Example

```
xds_line_ct_generate_customtone(16, 2, 1000, 1350, 10, 15, 8, 5, 3);
```

This will cause a tone consisting of 1000 Hz. at -10 dBm and 1350 Hz at -15 dBm. to be repeated 3 times on port 2 with an ON duration of 400 msec and an OFF duration of 250 msec.

xds_line_ct_set_transmit

xds_line_ct_set_transmit(board_number, port, stream, timeslot);

unsigned char board_number;	a value indicating which board the command is for
int port;	a value between 0 and the maximum number of ports on the board indicating which port should receive the tone
int stream;	a value indicating the stream number to use
int timeslot;	a value indicating the timeslot number to use

Applicable Boards

XDS Infinity H.100 & H.110 line-interface boards

Purpose

This function will create a one-way audio interface between the port and a CT bus timeslot. The port will be seized by this function.

Message Sent

“CXxxsstt” where **xx** is the port number, **ss** is the stream, and **tt** is the timeslot.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	an interface port was called for the board being used
ILL_SLOT	an invalid CT bus timeslot and or stream was selected for use
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

The timeslot the port will transmit on is determined by the port number and the base timeslot of the board set at initialization in accordance with the procedure spelled out in the 4.1 SC specification. Any audio path created by a previous command will be disabled.

The board will respond with a state change message of sub-type 14.

Example

xds_line_ct_set_transmit(16, 2, 0, 1);	this will cause port 2, stream 0 and timeslot 1, of board 16 to transmit to the CT bus
--	--

xds_line_ct_reset_lineport

xds_line_ct_reset_lineport(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; the number of the port that indicates the port interface

Applicable boards

XDS Infinity H.100 & H.110 line-interface boards

Purpose

This function is used to reset a port.

Message Sent

“RPxx” where **xx** is the port number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT the port is out of valid range

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function can be used to force a reset of a port.

Example

xds_line_ct_reset_lineport(16, 4);

this will reset port 4 on board 1

xds_line_ct_set_custom_dialtone

xds_line_ct_set_custom_dialtone(board_number, port, frequency1, frequency2, level1 level2, on_duration, off_duration);

unsigned char board_number;	a value indicating which board the command is for
int tone;	a value between 0 and 0x1A indicating which tone to use
int frequency1;	a value between 0 and 0x0CFF (0 to 3327 Hz) for the frequency of the first tone (in Hz), a value of 0 disables the tone
int frequency2;	a value between 0 and 0x0CFF (0 to 3327 Hz) for the frequency of the second tone (in Hz), a value of 0 disables the tone
int level2;	a value between 0 and 0x3F giving the level in -dBm of the second tone
int level1;	a value between 0 and 0x3F giving the level in -dBm of the first tone
int on_duration;	a value between 0 and 0xFF giving the duration in 50 msec. increments of the on portion of the tone
int off_duration;	a value between 0 and 0xFF giving the duration in 50 msec. increments of the off portion of the tone

Applicable Boards

XDS Infinity H.100 & H.110 line-interface boards

Purpose

This function can be used to generate a custom tone for user alerts or other purposes. The tone can be made up of either a single frequency or dual frequencies. The duration of both the on and off portion of the tone can be controlled, as well as the number of repetitions.

Message Sent

“SDttffffllffffllnnff” where **tt** is the tone, **ffff** is the frequency of the first and second tones, **ll** is the level of the first and second tones, **nn** is the on duration, and **ff** is the off duration.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	an interface port was called for the board being used
ILL_ARG	an out of range value was used for one of the arguments
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This function uses the DTMF generators to generate the custom tone.

The custom tones produced with this function can consist of one or two frequencies. The frequencies should be within the telephone frequency range of 0 to 3327 Hz. If only a single frequency is desired, the other frequency should have a value of 0. The level of the frequencies can be specified in a range of 0 dBm to -62 dBm, with a larger level value producing a quieter tone. If the frequency is not to be used, the level value should be set to 63.

The tone can be repeated, with the number of repetitions given by the reps value. This must be within the range 1 to 255. A value of 0 is not valid. The duration of the ON portion of the tone is given by on_dur in 50 msec. increments in a range of 50 msec to 12700 milliseconds. A value of 0xFF for the duration variable will play a continuous tone.

Example

```
xds_line_ct_set_custom_dialtone(16, 2, 1000, 1350, 10, 15, 8, 5, 3);
```

This will cause a tone consisting of 1000 Hz. at -10 dBm and 1350 Hz at -15 dBm. to be repeated 3 times on port 2 with an ON duration of 400 msec and an OFF duration of 250 msec.

xds_line_ct_set_custom_inftone

**xds_line_ct_set_custom_inftone(board_number, port, frequency1, frequency2, level1
level2, on_duration1, on_duration2, off_duration1,
off_duration2);**

unsigned char board_number;	a value indicating which board the command is for
int tone;	a value between 0 and 0x1A indicating which tone to use
int frequency1;	a value between 0 and 0x0CFF (0 to 3327 Hz) for the frequency of the first tone (in Hz), a value of 0 disables the tone
int level1;	a value between 0 and 0x3F giving the level in -dBm of the first tone
int frequency2;	a value between 0 and 0x0CFF (0 to 3327 Hz) for the frequency of the second tone (in Hz), a value of 0 disables the tone
int level2;	a value between 0 and 0x3F giving the level in -dBm of the second tone
int on_duration1;	a value between 0 and 0xFF giving the duration in 50 msec. increments of the on portion of the 1 st tone
int on_duration2;	a value between 0 and 0xFF giving the duration in 50 msec. increments of the on portion of the 2 nd tone
int off_duration1;	a value between 0 and 0xFF giving the duration in 50 msec. increments of the off portion of the 1 st tone
int off_duration2;	a value between 0 and 0xFF giving the duration in 50 msec. increments of the off portion of the 2 nd tone

Applicable Boards

XDS Infinity H.100 & H.110 line-interface boards

Purpose

This function can be used to generate a custom tone for user alerts or other purposes. The tone can be made up of either a single frequency or dual frequencies. The duration of both the on and off portion of the tone can be controlled, as well as the number of repetitions.

Message Sent

“SIttffffllffffllnnffnnff” where **tt** is the tone, **ffff** is the frequency of the first and second tones, **ll** is the level of the first and second tones, **nn** is the on duration for the first and second tones, and **ff** is the off duration for the first and second tones.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	an interface port was called for the board being used
ILL_ARG	an out of range value was used for one of the arguments
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This function uses the DTMF generators to generate the custom tone.

The custom tones produced with this function can consist of one or two frequencies. The frequencies should be within the telephone frequency range from 0 to 3327 Hz. If only a single frequency is desired, the other frequency should have a value of 0. The level of the frequencies can be specified in a range of 0 dBm to -62 dBm, with a larger level value producing a quieter tone. If the frequency is not to be used, the level value should be set to 63.

The tone can be repeated, with the number of repetitions given by the reps value. This must be within the range 1 to 255. A value of 0 is not valid. The duration of the ON portion of the tone is given by on_dur in 50 msec. increments in a range of 50 msec to 12700 milliseconds. A value of 0xFF for the duration variable will play a continuous tone.

Example

```
xds_line_ct_set_custom_inftone(16, 2, 1000, 10, 1350, 15, 8, 5, 3);
```

This will cause a tone consisting of 1000 Hz. at -10 dBm and 1350 Hz at -15 dBm. to be repeated 3 times on port 2 with an ON duration of 400 msec and an OFF duration of 250 msec.

xds_line_ct_set_gain

```
int xds_line_ct_set_gain(board_number, port, xmt_polarity, xmt_gain, rcv_polarity,  
                        rcv_gain);
```

unsigned char board_number;	a value indicating which board the command is for
int port;	a value between 0 and the maximum number of ports on the board indicating which port should receive the tone
char xmt_polarity;	'+' positive gain, '-' negative gain
short xmt_gain;	a value between 0 and 10 indicating the amount of gain in 1 dBm steps
char rcv_polarity;	'+' positive gain, '-' negative gain
short rcv_gain;	a value between 0 and 10 indicating the amount of gain in 1 dBm steps

Applicable Boards

XDS Infinity H.100 & H.110 line-interface boards

Purpose

This function is used to control the gain associated with each port on the board.

Message Sent

“SGxxpttpr” where **xx** is the port is the port, the first **p** is the polarity of the transmit gain, **tt** is the transmit gain level, the second **p** is the polarity of the receive gain, and **rr** is the receive gain level.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	an interface port was called for the board being used
ILL_MODE	an invalid polarity was selected for use
ILL_ARG	an out of range value was used for one of the gain arguments
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

There may be situations because of loop length where it is desirable to add gain or attenuation to a port. This can be done with this command. The polarity can be either a “+” or a “-”. The gain is specified in 1 dBm steps with a range from -10 to + 10 dBm.

Example

xds_line_ct_set_gain(16, 2, '+', 6, '-', 8)	this sets port 2 on board 16 for 6 dB of transmit gain and -8 dB receive gain
---	---

xds_line_ct_set_hookflash

int xds_line_ct_set_hookflash(board_number, port, min_time, max_time);
unsigned char board_number; a value indicating which board the command is for
int port; a value indicating which port is being controlled
unsigned char min_time; the minimum amount of time to detect a hookflash
unsigned char max_time; the maximum amount of time to detect a hookflash

Applicable Boards

XDS Infinity Loop, E&M, and Station boards

Purpose

This function is used to set the hookflash detect times for a specified port.

Message Sent

“SHxxaabb” where xx is the port, aa = minimum time, bb = maximum time.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used

ILL_PORT an invalid port was selected

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The timing of the flash or pause signal may vary with switch type and country. In North America the “flash” signal is normally between 350 and 1000 msec. In Europe the “pause” signal is shorter, between 50 and 100 msec. To allow for these variations, the hook-flash timing can be changed on a port by port basis. On-hook signals of less than the minimum time will be ignored, those that are greater than the maximum will be treated as a disconnect. Note that this only affects ports set to type “**phone**”.

The hook flash timing parameters may be saved in the EEROM.

Example

`xds_line_ct_set_hookflash(16, 2, 28, 4B);` this would set the minimum time for port 2
to 400 msec. and the maximum time to 750
msec.2 of board 16

xds_em_ct_lead_control

int xds_em_ct_lead_control(board_number, port, mode);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
 board indicating which port should receive the tone
char mode; a value indicating the relay mode, 'O' - open, 'C' - close

Applicable Boards

XDS Infinity H.100 & H.110 E&M boards

Purpose

This function is used to control a pair of relays associated with each port on this special interface board.

Message Sent

“XCxx” closes relay for port xx

“XOxx” opens relay for port xx

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an interface port was called for the board being used
ILL_MODE an invalid relay mode was chosen
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function controls the relays on a special interface board developed for connecting to radio systems. There is also an audio interface that is identical to that on an E&M board. No 'E' or 'M' lead control signals are supported.

Example

xds_em_ct_lead_control(16, 2, 'C') this closes relay for port 2 on board 16

xds_em_ct_lead_control(16, 2, 'O') this opens relay for port 2 on board 16

xds_em_ct_guard_tone

int xds_em_ct_guard_tone(board_number, port, channel_number);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
 board indicating which port should receive the tone
char channel_number; a value indicating the relay mode, 'O' - open, 'C' - close

Applicable Boards

XDS Infinity H.110 E&M boards

Purpose

This function is used to play a guard tone on a port for a given channel number or remove the guard tone.

Message Sent

“CGxxt” plays low-level guard tone on port **xx** for channel number **t**

“CGxxX” removes low-level guard tone for port **xx**

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an interface port was called for the board being used
ILL_MODE an invalid channel number was chosen
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This command will only work if the port is in the Connect or MVIP connect state. It will cause a 2175 HZ tone followed by a tone that selects the channel number. The valid tones are: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'. They correspond to channel numbers 1 – 16. To remove the tone is 'X'.

Example

xds_em_ct_guard_tone(1, 2, '0') this plays a guard tone followed by a tone for channel number 1 on port 2 for board 1.

xds_em_ct_guard_tone(1, 2, '1') this plays a guard tone followed by a tone for channel number 2 on port 2 for board 1.

xds_em_ct_guard_tone (1, 2, 'X') this removed the guard tone on port 2 for board 1

xds_loop_ct_port_options

int xds_loop_ct_port_options(board_number, port, option_a, option_b, option_c);
unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
 board indicating which port should receive the tone
char option_a; a value indicating the first option mode
char option_b; a value indicating the second option mode
char option_c; a value indicating the third option mode
char option_d; a value indicating the fourth option mode

Applicable Boards

XDS Infinity H.100 & H.110 Loop-start Interface boards

Purpose

Sets options for a given Loop-start board port.

Message Sent

“SOxxabc” where **xx** is the port, **a** is the first option, **b** is the second option, **c** is the third option, and **d** is the fourth option.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an interface port was called for the board being used
ILL_MODE an invalid first, second, third, or fourth option mode was chosen
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

Sets the options o for port xx. Each option may be set to ‘Y’ for yes, ‘N’ for no, or ‘*’ for no change. The default is no. If option **a** is set to ‘Y’ (yes), an “SF” message will be sent if loop current is detected upon seizure of the loop. If option **b** is set to ‘Y’ (yes), changes in battery polarity will be reported with “Sr” and “Sn” messages. If option **c** is set to ‘Y’ (yes), the line will not be seized, but an audio path will be created when a *connect* or other command is issued. If option **d** is set to ‘Y’ (yes), the port will not be disconnected on a linebreak. If option **d** is set to ‘N’ (no), then the port will be disconnected on a linebreak.

Examples

```
xds_loop_ct_port_options(17, 0, 'Y', 'N', 'Y', 'N')
```

this will send an “SF” message if loop current is detected upon seizure of the loop and the line will not be seized, but an audio path will be created when a *connect* or other command is issued, and the port will be disconnected on a linebreak on port 0 on board 17

```
xds_loop_ct_port_options(17, 2, 'N', 'Y', 'N', '*')
```

this will report “Sr” and “Sn” messages when there are changes in battery polarity on port 2 on board 17

This page was intentionally left blank.

H.100/110 Station Board Functions

This page was intentionally left blank.

xds_station_ct_call_waiting

int xds_station_ct_call_waiting(board_number, port, tone);

unsigned char board_number; a value indicating which board the command is for
int port; a value indicating which port is being controlled
int tone; a value between 0 and 4 indicating which call progress tone
 to play

Applicable Boards

XDS Infinity Station boards

Purpose

This function can be used for the call waiting feature on the boards.

Message Sent

“CW xx t” where xx is the port number and t is the tone.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used
ILL_ARG an invalid tone was selected
ILL_PORT an invalid port number was called
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

A tone can be issued to a port that is already in the connect state to indicate that another call is waiting. This tone consists of a 440 Hz. signal in one of several patterns. The command to send this tone takes the form “**CWxxp**” where xx is the port number and p is the pattern. The pattern codes are:

- 1 - a single tone for 300 msec.
- 2 - two tone bursts of 100 msec. each with 100 msec. Between
- 3 - three tone bursts of 300 msec. each with 100 msec. Between
- 4 - 100 msec. tone, 300 msec., and 100 msec. Tones

To respond to this tone, the user sends a hook-flash signal. The application can then put the first call on hold and connect the user to the second caller.

Example

```
xds_station_ct_call_waiting(16, 2, 3);
```

this will play tone 3 to port 2 of board 16

xds_station_ct_return_polarity

int xds_station_ct_return_polarity (board_number, port);
unsigned char board_number; a value indicating which board the command is for
int port; a value indicating which port is being controlled

Applicable Boards

XDS Infinity Station boards

Purpose

This function is used to return the polarity on a given port back to normal.

Message Sent

“PNxx” where xx is the port.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used
ILL_PORT an invalid port number was called
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The polarity may only be reversed for ports that are not in the idle or ringing states. The polarity will automatically return to normal when a port is disconnected or goes on-hook.

Example

xds_station_ct_return_polarity(16, 2); this will return the polarity on port 2 for
board 16 back to normal

xds_station_ct_reverse_polarity

int xds_station_ct_reverse_polarity(board_number, port);

unsigned char board_number; a value indicating which board the command is for

int port; a value indicating which port is being controlled

Applicable Boards

XDS Infinity Station boards

Purpose

This function is used to reverse the polarity on a given port.

Message Sent

“PRxx” where **xx** is the port.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used

ILL_PORT an invalid port number was called

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The polarity may only be reversed for ports that are not in the idle or ringing states. The polarity will automatically return to normal when a port is disconnected or goes on-hook.

Example

xds_station_ct_reverse_polarity(16, 2); this will reverse the polarity on port 2 for
board 16

xds_station_ct_set_calling_name

int xds_station_ct_set_calling_name(board_number, port, calling_name);
unsigned char board_number; a value indicating which board the command is for
int port; a value indicating which port is being controlled
char *calling_name; a pointer to the calling name to be displayed

Applicable Boards

XDS Infinity Station boards

Purpose

This function is used to display a calling party's name.

Message Sent

“DNxx<name>” where **xx** is the port is the port and **name** is the name to be displayed.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used
ILL_ARG an invalid (NULL) calling name was selected
ILL_PORT an invalid port was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The calling party name may be added to the information with a command of the form “DNxxname” where **xx** is the port and name is the calling party name. The board will accept up to 28 characters in the name field, but current devices are limited to either 15 or 21 characters. In the case where the name is not available or private, the name may either be omitted or replaced with a “?O” if not available or “?P” if private. The “DN” command must be sent after the “DD” command.

Example

```
xds_station_ct_set_calling_name(16, 2, “John Smith”);
```

this sets port 2 on board 16, with “John Smith” calling

xds_station_ct_set_date_calling_number

int xds_station_ct_set_date_calling_number(board_number, port, date, calling_num);
unsigned char board_number; a value indicating which board the command is for
int port; a value indicating which port is being controlled
char *date; a pointer to the date to be displayed
char *calling_num; a pointer to the calling number to be displayed

Applicable Boards

XDS Infinity Station boards

Purpose

This function is used to display a date along with a calling party's number.

Message Sent

“DDxxmmddhhmm/#” where **xx** is the port is the port, **mmdd** is the month and date, **hhmm** is the hour and minute, and **#** is the calling party number to be displayed.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used
ILL_ARG an invalid (NULL) calling number was selected
ILL_PORT an invalid port was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

To send caller identity information, the time, date, and number must be set up before the ringing or call waiting command is issued. This is done with a message of the form “DDxxmmddhhmm/#” where **xx** is the port number, **mmdd** is the month and date, **hhmm** is the hour and minute, and **#** is the calling party number. The date and time are in decimal using a 24 hour format. The calling party number should be between 4 and 10 digits. Longer numbers may not display properly. In some cases, it may desired not to display the number either because it is not available or restricted. In this case, the number may be replaced with an “O” for out of area if not available, or with a “P” for private.

Example

```
xds_station_ct_set_date_calling_number(16, 2, “04030201”, “6085551234”);
```

this will display “April 3, 2:01 AM, 608-555-1234” on port 2 of board 16

xds_station_ct_set_port_profile

int xds_station_ct_set_port_profile(board_number, profile);

unsigned char board_number; a value indicating which board the command is for
char *profile; a pointer to a string of characters which sets the profile for
 all of the ports on the board

Applicable Boards

XDS Infinity Station boards

Purpose

This function is used to set the hookflash detect times for a specified port.

Message Sent

“SPpp...p” where **p** is the port profile to be used for the respective port

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used
ILL_ARG an invalid profile / profile string was selected
ILL_PORT an invalid port was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

Each port can be set to the required characteristics by selecting the appropriate parameter profile. Ten different profiles numbered 0-9 are provided. Six of these are fixed, while four can be customized for specific needs. The profiles are selected using the “**SP**” command. This command consists of “**SP**” followed by the profile number of each of the 24 ports. The profiles are:

- 0 - default does not use CRAM coefficients
- 1 - North American, 20 Hz. internal ringing
- 2 - North American, 30 Hz. internal ringing
- 3 - North American, 20 Hz. external ringing
- 4 - European, 25 Hz. internal ringing
- 5 - European, 25 Hz. external ringing
- 6 - User profile 1
- 7 - User profile 2
- 8 - User profile 3
- 9 - User profile 4

Example

```
xds_station_ct_set_port_profile(16, “111111111111222222222222”);
```

this would set the port profile for the first 12 ports to the North American 20 Hz. Internal ring, and the last 12 ports to the North American 30 Hz. internal ring

xds_station_ct_set_profile_parms

int xds_station_ct_set_profile_parms(board_number, profile_num, reg, parms);
unsigned char board_number; a value indicating which board the command is for
char profile_num; a character of the selected profile
int reg; the selected register
char *parms; a pointer to a string of characters which sets the parameters
for a specified profile

Applicable Boards

XDS Infinity Station boards

Purpose

This function is used to set the custom port profiles.

Message Sent

“SPVpr(vv...v)” where **p** is the port profile to be configured, **rr** is the register number to be used, and the **v**’s are the desired values for the port profiles.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used
ILL_ARG an invalid register or an invalid/NULL parameter string was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function takes a bit of knowledge in programming custom profiles, therefore you will need to refer to the *H.100 Station Board Manual* (Amtelco P/N 257M003).

Example

```
xds_station_ct_set_profile_parms(16, 6, 70, “4A0C2324C4922403”);
```

xds_station_ct_set_visual_message_waiting

int xds_station_ct_set_visual_message_waiting(board_number, port, enable);
unsigned char board_number; a value indicating which board the command is for
int port; a value indicating which port is being controlled
char enable; a value of 'F' or 'N' indicating the enable mode to be used

Applicable Boards

XDS Infinity Station boards

Purpose

This function is used to turn the *Visual Message Waiting Indicator* on or off.

Message Sent

“DVxxF” where **xx** is the port

or

“DVxxN” where **xx** is the port

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used
ILL_PORT an invalid port was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The Visual Message Waiting Indicator is controlled using a command of the form “**DVxxa**” where **xx** is the port and **a** is the action, either “F” to turn the indicator on or “N” to turn the indicator off. This command can only be issued when the port is in the on-hook idle condition.

Example

```
xds_station_ct_set_visual_message_waiting(16, 1, 'F');
```

this will turn Visual Message Waiting Indicator off on port 1 of board 16

xds_station_ct_set_voltage

int xds_station_ct_set_voltage(board_number, v1, v2);

unsigned char board_number; a value indicating which board the command is for
char v1; a value between 0 and 6
char v2; a value between 0 and 7

Applicable Boards

XDS Infinity Station boards

Purpose

This function is used to set the voltage levels of both fixed and user profiles on a board.

Message Sent

“SVab” where **xx** is the port

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-Infinity Station board was used

ILL_ARG invalid voltage values were chosen

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

These are the available voltage levels for each board:

0 - automatic detection

1 - 28 V

2 - 48 V

3 - 72 V

4 - 96 V

5 - 120 V

6 - 144 V

7 - user value (programmable profiles only)

Example

xds_station_ct_set_voltage(16, 1, 2);

this will set the voltage level for fixed profiles to 28 Volts and user profiles to 48Volts on board 16.

This page was intentionally left blank.

**ISA / PCI / cPCI
Common Line-interface
Board Functions**

This page was intentionally left blank.

xds_line_cptones

int xds_line_cptones(board_number, port, tone);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
 board indicating which port should receive the tone
int tone; a value between 0 and 5 indicating which call progress tone
 to play

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

This function can be used to play any of the six precise call progress tones to a port. These tones are dial tone, reorder, busy, and audible ringback. In addition, silence and a calibration tone of 1004 Hz. can be played at a level corresponding to 1 milliwatt.

Message Sent

“CPxxt” where **xx** is the port number and **t** is the tone.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an interface port was called for the board being used
ILL_ARG an invalid call progress tone was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comment

Using **xds_line_cptones** will seize a port unless that port is an E&M or DID port that has is in the off-hook waiting or listen state. In those states, the seizure condition will remain unchanged. To force a seizure, **xds_line_hold** may be called before the **xds_line_cptones**. Any existing audio path to or from this port will be disabled by this function. There is no limit on the number of ports listening to any of the call progress tones.

The six tones produced by this function are selected with the tone argument. The tones produced are:

- 0 - dial tone
- 1 - reorder tone
- 2 - busy tone
- 3 - audible ringback
- 4 - digital milliwatt 1004 Hz steady tone.
- 5 - silence

The first four tones are as defined in RS-464.

Example

`xds_line_cptones(1, 2, 3);` this will play “busy tone” to port 2 of board 1

xds_line_detect_energy

int xds_line_detect_energy(board_number, port, duration);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
 board indicating which port should receive the tone
int duration; a value between 0 & 224 indicating the number of .1 of
 second energy must persist before being reported, if 0,
 detection is disabled

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

This function will set a port to detect audio energy. This can be used for detecting call progress tones or voice to determine whether a call has been answered.

Message Sent

“CExxdd” where **xx** is the port and **dd** is the duration in the range 20 - FF.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an interface port was called for the board being used

ILL_ARG an invalid value for the duration was used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The energy detect function can be used to detect whether a call has been answered, or to detect call progress signals either before or after dialing. Any connection that exists at the time detection is enabled will remain. If the port is not transmitting when the command is issued, it will be placed in the transmit state. If detection is disabled using the **xds_line_detect_energy** function with the duration set to 0, any connection will not be affected. This means that if the port is in the transmit state, it will remain in that state. Detection will also be disabled by any command that changes the state of the port, for example a connect or a hold function.

The minimum duration that the energy must persist before being reported is set using the duration value, which is in tenths of a second. When a signal has lasted for this amount of time it will be reported. Cessation of the signal will also be reported. It is not necessary to re-enable detection after a signal has been detected. Thus, cadences of regularly repeating signals can be determined. Also, the duration can be used to discriminate between signals of different duration. Thus, by setting the duration to 4, the call progress signal busy which consists of .5 seconds of tone followed by .5 seconds of silence will be detected and reported, while reorder, which consists of .25 sec. of tone followed by .25 sec. of silence will not be reported.

Energy detection will be reported in a message of type 1, sub type 15. If the argument is 1, energy has been detected, if the argument is 0, a detected signal has stopped.

Examples

`xds_line_detect_energy(1, 2, 4)`

this will detect energy of .4 seconds or greater duration on port 2 of board 1.

`xds_line_detect_energy(1, 2, 0)`

will disable energy detection on port 2 of board 1

xds_line_disconnect

int xds_line_disconnect(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
board indicating which port should receive the tone

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

This function will release a port. The line will be released and the transmit timeslot will be disabled.

Message Sent

“CDxx” where xx is the port number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an interface port was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The port will be returned to the idle state by this function. However, E&M ports and station ports set to type **phone** will not be returned to the idle state until after they have gone on-hook in response to the port release. A state change message of sub-type 7 will be returned by the board in response to this function.

Example

xds_line_disconnect(1, 2); this will disconnect port 2 of board 1

xds_line_hold

int xds_line_hold(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
 board indicating which port should receive the tone

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

This function will put the port in the hold state. The port will be seized but audio to and from the port will be disabled.

Message Sent

“CHxx” where **xx** is the port number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an interface port was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

Using **xds_line_hold** will seize a port unless that port is an E&M or Station port that has been issued an **xds_line_disconnect** and has not yet gone on-hook to return to the idle state. The board will respond to an **xds_line_hold** with a state change message of sub_type 6.

There are three valid ways for a port to reach the hold state. The first is in response to a call to the **xds_line_hold** function. The second is in response to a call to the **xds_line_seize** function if the port is in the idle state. The third is at the termination of sending a tone string.

Example

xds_line_hold(1, 2); this will place port 2 on board 1 in the hold state

xds_line_hookflash

xds_line_hookflash(board_number, port, mode, time);

unsigned char board_number;	a value indicating which board the command is for
int port;	a value between 0 and the maximum number of ports on the board indicating which port should receive the tone
char mode;	a value specifying what mode of time interval to use for the hookflash (10 millisecond steps or 100 millisecond steps)
int time;	a value between indicating the duration in .1 sec. or .01 sec, depending on the mode, of the hookflash

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

This function will send a hook-flash out of the B-channel. The duration of the hook-flash is determined in either .1 second or .01 second increments by the **time** argument, depending on the mode selected. Hook-flash signals may be used to get the attention of a PBX or other phone system for performing some action such as transferring a call.

Message Sent

“CFxxd” where **xx** is the B-channel number and **d** is the duration in .1 second increments.

“CFxxdd” where **xx** is the B-channel number and **dd** is the duration in .01 second increments.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT	an interface port was called for the board being used
ILL_MODE	an invalid time interval mode was selected
ILL_ARG	an invalid value for the hookflash time was selected
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

The range of hook-flashes that may be sent is .1 to 1.5 seconds or .01 to 2.5 seconds. If current is not flowing in the line circuit, this command will cause a port to hangup. No state change response message is sent for this function.

Example

xds_t1e1_hookflash(16, 2, 1, 7); will cause a hook-flash of .7 seconds on port 2 of board 16

xds_t1e1_hookflash(16, 2, 2, 7); will cause a hook-flash of .07 seconds on port 2 of board 16

xds_line_disable_output

xds_line_disable_output(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
board indicating which port should receive the tone

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

This function will disable audio output to the SCbus in those states where an idle pattern may be output. These states are hold, listen, off-hook waiting, or idle-pattern. In the listen state, the port will change to the hold state. In the idle pattern state, the port will change to the idle state. In the other states, the state of the port will not change. Calling this function for a port in any other state will have no effect.

Message Sent

“CIxx” where **xx** is the port number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an interface port was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

When a port changes from the connect or transmit state to the hold state, or when an on-hook condition is detected in either of those states, an idle pattern of silence will be output to the SCbus. It may be desirable in these cases to disable the output to the SCbus. This is done with the **xds_line_disable_output** function. The states will change as indicated above, with the appropriate state change message being sent by the board. In the 4.1 SC model, transmit timeslots are reserved for each port, so there is no need to free a timeslot with an idle pattern on it. Therefore, there is no benefit to using this function under this model.

Example

xds_line_disable_output(1, 2); will disable the idle pattern for port 2.

xds_line_state

xds_line_state(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
board indicating which port should receive the tone

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

This function will return the current line state of a port on an analog line board or of a B-channel on the Basic Rate ISDN Board. This information may be used for diagnostic purposes.

Message Sent

None.

Returns

This function will return with the port state code. If there is no board with the proper board number or the port is not a valid port on that board, a (-1) will be returned.

Comments

This function obtains information by reading a table in the dual-ported ram. This table is synchronized to an internal table. However, there may be cases where there are differences between the two.

The defined states for the analog line boards are:

- 0 - idle
- 1 - connected
- 2 - on hold
- 3 - ringing being detected
- 4 - transmitting, audio to the SCbus
- 5 - ringing being generated
- 6 - listening, audio from the SCbus or an internal source
- 7 - off-hook waiting to proceed
- 8 - conferenced (using resources local to the board only)
- 9 - off-hook waiting to go idle
- 10 - idle pattern - generating an idle pattern to the SCbus, behavior the same as idle

For the SCSA U-Interface Basic Rate ISDN Board, the line states reflect the state of the B-channels and not the call state. Note, that the state information is not available for the S/T-Interface board. The states for the SCSA BRI Boards are:

- 0 - idle, no audio path
- 1 - connect
- 2 - hold
- 3 - transmit, one way audio to the SCbus
- 4 - listen, audio from the SCbus or an internal source
- 5 - detecting, a DTMF or energy detector connected to the B-channel

Example

`xds_line_state(1, 2);`

this requests the state of port 2 on board 1, the function will return the state.

xds_line_detect_dtmf

int xds_line_detect_dtmf(board_number, port, mode);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
 board indicating which port should receive the tone
char mode; a character indicating listen mode is to be used

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

This function will set a port to detect DTMF digits. Dial-tone will be generated until the first digit is detected unless the “Quiet” option is chosen, in which case, no dial-tone will be played. The ‘M’ or “Monitor” option can be used to detect digits while the port is in the connect state. As each DTMF tone is detected, a state change message of sub-type 13 will be sent by the board with the argument indicating the tone.

Message Sent

“CLxx” where **xx** is the port number or “CLxxo” where **o** indicates an option.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT

an interface port was called for the board being used

IOCTL

an IOCTL was called and returns a return value from the IOCTL call

Comments

The DTMF detector is capable of detecting all 16 of the standard DTMF tones, 0-9, *, #, A-D. Each board is equipped with eight detectors, one for each line. Once a detector is assigned, it remains assigned until a command is issued for that port or an on-hook is detected.

The DTMF detect function will disable any audio paths to or from the port at the time it is called. If connected, the audio path will be broken and must be reestablished after the digits are detected. The port will be seized by this function.

If the monitor mode is used by selecting the 'M' option, existing connections or one-way audio paths will be maintained. The state of the port will also remain unchanged. Because connections are maintained, no dial-tone is played to the port. Digits detected will be reported in the usual manner. Detection may be stopped by using the `xds_line_detect_dtmf` function with the mode set to 'F'. Detection will also be stopped if the port changes state due to a command or change in hook status. The monitor mode is only supported in firmware versions 2.0 and higher.

if ' ', dial tone is played until the first DTMF tone is detected
if 'Q', no dial tone is played,
if 'M', DTMF is monitored without affecting existing connections
if 'F', DTMF detection is turned off

Example

`xds_line_detect_dtmf(1, 2, ' ');` this will set port 2 of board 1 to listen for DTMF tones, dial tone will be played until the first digit is detected

The board will respond with a message of type 1, subtype 9, port number 2. As each DTMF digit is detected, the board will return a message of type 1, subtype 13, port number 2. The argument will be the digits detected encoded as an ASCII character.

`xds_line_detect_dtmf(1, 2, 'Q');` this will do the same as the first example, except dial tone will not be played.

`xds_listen_dtmf(1, 2, 'M');` this will turn on detection but maintain existing connections
`xds_listen_dtmf(1, 2, 'F');` this will turn off detection while maintaining existing connections

xds_line_seize

xds_line_seize(board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
board indicating which port should receive the tone

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

The purpose of this function is to seize an idle port and place it in the hold state. If a port is not idle, it will not be seized and a “busy” state change will be returned. This function can be used to minimize “glare” or collisions between incoming and outgoing calls.

Message Sent

“CBxx” where xx is the port number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

(-1) the port is busy
ILL_PORT an interface port was called for the board being used
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function may be used for reducing the possibility of “glare” by calling the function before making an outgoing call on a port. If the port is busy because an incoming call has been detected, the function will return a (-1) and the board will return a “SBxx” message. An additional state change message, such as ring detect or off-hook, indicating the incoming call on that port should also be received.

Example

xds_line_seize(1, 2); this will seize port 2 on board 1

xds_line_send_dial_string

xds_line_send_dial_string(board_number, port, tones);

unsigned char board_number; a value indicating which board the command is for
int port; a value between 0 and the maximum number of ports on the
 board indicating which port should receive the tone
char *tones; a pointer to an ASCII string of tones to be played

Applicable Boards

XDS line-interface and BRI-interface boards

Purpose

This function will cause a string of DTMF tones to be sent out a port. The tone string takes the form of a NULL terminated ASCII string. The port will be seized by this function. In addition to the standard DTMF tones, the string may consist of pauses and single frequency tones.

Message Sent

“CTxx<string>” where **xx** is the port.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_PORT an interface port was called for the board being used

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

Each board has DTMF generators. If an attempt is made to generate more tone strings than possible by a board, an error will be returned from this function.

The allowed tones may vary from board to board, so please refer to your hardware manual for a list of these.

Example

xds_line_send_dial_string(1, 2, “123P456”); this will cause the tone string “123P456” to
 be sent out port 2 of board 1.

This page was intentionally left blank.

**Common Enhanced Conference
Board Functions
(H.110 Library Version)**

This page was intentionally left blank.

xds_ct_cnf_timeslot

xds_ct_cnf_timeslot (board_number, port);

unsigned char board_number; a value indicating which board the command is for
int port; a value indicating the conference port outputting to the CT
 bus

Applicable boards

H.100/H.110 boards with conferencing resources

Purpose

This function returns the transmit timeslot for a specified conference port.

Message Sent

None

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

WRONG_BOARD a non-PCI based board was used for this function

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The number of available conference ports on each board varies, so please consult the board's reference manual for this number.

Example

xds_ct_cnf_timeslot (16, 4); This would return the transmit timeslot of conference port 4
 on board 16.

xds_ct_conference

xds_ct_conference(board_number, handle, ct_xmt_stream, ct_xmt_timeslot, ct_rcv_stream, ct_rcv_timeslot, atten, threshold);

unsigned char board_number;	a value indicating which board the command is for
char handle;	a value indicating which conference handle to assign
int ct_xmt_stream;	a value indicating which CT bus stream is serving as an output
int ct_xmt_timeslot;	a value indicating which CT bus timeslot is serving as an output
int ct_rcv_stream;	a value indicating which CT bus stream is serving as an input
int ct_rcv_timeslot;	a value indicating which CT bus timeslot is serving as an input
char atten;	a value between 0-7 that controls the input and output attenuation of the conference on the specified timeslots
char threshold;	a value between 0-3 specifying a noise threshold below which the input will not be added to a conference

Applicable boards

H.100/H.110 boards with conferencing resources and the SCSA Enhanced conference board

Purpose

This function adds a CT bus connection to the conference with the specified handle. The connection consists of a pair of CT bus timeslots, one for an input and one for an output. The attenuation and noise threshold parameters used by the conferencing hardware are specified.

Message Sent

“CAhhooooiiiiat” where **hh** is the conference handle, **oooo** is the CT bus output stream and timeslot, **iiii** is the CT bus input stream and timeslot, **a** is the attenuation parameter and **t** is the noise threshold parameter.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_SLOT	an illegal input and/or output stream/timeslot was used for the board
ILL_HANDLE	an invalid conference handle was chosen for the conference
ILL_ATTEN	an invalid attenuation value was chosen for the conference
ILL_THRESHOLD	an invalid threshold value was chosen for the conference
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

Each conference board is equipped with a different number of conference bridges, varying on board type and density. Each group of conferences share a set of outputs called “Conference Control Addresses” or CCA’s. To best determine the limit, please consult the hardware reference manual. The attenuation parameter can be used to attenuate the conference input, output, or both, in 3 dB steps. The following levels of attenuation can be achieved using these attenuation parameter values:

<u>value</u>	<u>input attenuation</u>	<u>output attenuation</u>
0	0 dB	0dB
1	0 dB	3 dB
2	3 dB	0 dB
3	3 dB	3 dB
4	6 dB	0 dB
5	6 dB	3 dB
6	9 dB	0 dB
7	9 dB	3 dB

As more parties are added to a conference, it may be necessary to increase the amount of attenuation to produced acceptable audio performance. The noise threshold parameter sets a threshold below which an input signal is not added to a conference. A value of 0 sets it for no threshold, while 3 sets the threshold to the highest value. The higher levels of noise threshold will introduce increasing levels of distortion and so should be used with caution. Typical values for small conferences are 0 for both the attenuation and noise threshold parameters. An **xds_ct_conference** call may be issued for a connection already conferenced to change the attenuation and noise threshold parameters for that connection. The **xds_conference_parameter** function can be used to change the parameters for all parties to a conference. It is the responsibility of the application to select a conference handle and to keep track of which handles are in use and which parties are assigned to each handle.

Valid values for the transmit streams and receive streams are 0 – 31. Valid values for the transmit timeslots and receive timeslots are 0 – 127.

`xds_ct_conference_mode()` must be called before doing any conferencing and calling and calling this function. This will enable the user’s desired conference mode and enable proper error-checking in the library code.

Example

```
xds_ct_conference(16, 4, 0, 14, 1, 3, 2, 1);
```

This would use conference handle 4 on board 16 for the conference. Stream 0, timeslot 14 would serve as the output and stream1, timeslot 3 would be the input. The attenuation parameter is set to 2 and the noise threshold is set to 1.

xds_ct_conference_disable

xds_ct_conference_disable(board_number, ct_xmt_stream, ct_xmt_timeslot);

unsigned char board_number; a value indicating which board the command is for
int ct_xmt_stream; a value indicating which CT bus stream is serving as an
 output
int ct_xmt_timeslot; a value indicating which CT bus timeslot is serving as an
 output

Applicable boards

H.100/H.110 boards with conferencing resources and the SCSA Enhanced conference board

Purpose

This function will remove the output from the conference bridge.

Message Sent

“CDoooo” where **oooo** is the CT bus stream and timeslot.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_SLOT an illegal output timeslot was used for the board
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function will remove a connection from a conference.

Valid values for the transmit streams are 0 – 31. Valid values for the transmit timeslots are 0 – 127.

Example

xds_ct_conference_disable(16, 0, 2); this would remove stream 0, timeslot 2 from
 a conference on board 16

xds_ct_conference_dtmf

xds_ct_conference_dtmf(board_number, ct_rcv_stream, ct_rcv_timeslot, time);
unsigned char board_number; a value indicating which board the command is for
int ct_rcv_stream; a value indicating which CT bus stream is serving as an
 input
int ct_rcv_timeslot; a value indicating which CT bus timeslot is serving as an
 input
int time; a value between 1 - 207 indicating the duration of the
 clamping feature, a value of 0 disables DTMF detection

Applicable Boards

H.100/H.110 boards with conferencing resources and the SCSA Enhanced conference board

Purpose

This function will enable the detection of DTMF tones on a specified CT bus timeslot. The timeslot must be an active input to a conference for detection to be possible. The “clamping” feature can be enabled which will suppress the specified input to the conference for the amount of time specified when a DTMF digit is detected.

Message Sent

“CEiiiiidd” where **iiii** is the CT bus stream and timeslot and **dd** is the clamping duration.
“CEiiiiD” if detection/clamping is to be disabled on CT bus timeslot **iiii**.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG an invalid value of time was used
ILL_SLOT an illegal input timeslot was used for the board
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

DTMF detection can be enabled simultaneously on all possible conference inputs. DTMF detection can only be enabled on active inputs to conferences. If the input is removed from the conference using the **xds_ct_conference_disable** function or the conference is dissolved using the **xds_ct_conference_dissolve** function, detection will be disabled.

A “clamping” feature is also provided as part of DTMF function. If this feature is enabled by specifying a time in the range of 1-207, the input to the conference will be suppressed if a DTMF digit is detected. The suppression will begin as soon as a DTMF digit is qualified (approximately 20 msec.) and will continue for the specified time. This time is given in multiples of 20 msec. with a maximum duration of 4.140 seconds. If the time parameter is 0, DTMF detection and clamping will both be disabled.

Valid values for the receive streams are 0 – 31. Valid values for the receive timeslots are 0 – 127.

Example

`xds_ct_conference_dtmf(16, 0, 3, 10);` this will enable DTMF detection on stream 0,
timeslot 3 with a clamping duration of 200 msec.

xds_ct_conference_monitor

xds_ct_conference_monitor(board_number, handle, ct_xmt_stream, ct_xmt_timeslot);

unsigned char board_number;	a value indicating which board the command is for
int handle;	a value indicating the conference handle to use
int ct_xmt_stream;	a value indicating which CT bus stream is serving as an output
int ct_xmt_timeslot;	a value indicating which CT bus timeslot is serving as an output

Applicable boards

H.100/H.110 boards with conferencing resources and the SCSA Enhanced conference board

Purpose

This function will establish a CT bus output from the conference specified by the handle. No input will be established.

Message Sent

“CMhhoooo” where **oooo** is the CT bus output stream and timeslot and **hh** is the conference handle.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_SLOT	an illegal input and/or output timeslot was used for the board
ILL_HANDLE	an invalid conference handle was chosen for the conference
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

A conference monitor can be treated as a conference party with a silent input. As with a full conference party, a Conference Control Address and conference port is used. A conference monitor may be disabled by using the **xds_ct_conference_disolve** or **xds_ct_conference_disable** functions. If the **xds_ct_conference_monitor** function is called for a conference port that is in use, the input from that port will be removed from the conference. Similarly, if an **xds_ct_conference** function is called for a port that is monitoring, the port will become a full party to the conference.

Valid values for the transmit streams are 0 – 31. Valid values for the transmit timeslots are 0 – 127.

`xds_ct_conference_mode()` must be called before doing any conferencing and calling and calling this function. This will enable the user's desired conference mode and enable proper error-checking in the library code.

Example

```
xds_ct_conference_monitor(16, 2, 0, 3);
```

this will monitor the conference with handle 3 using stream 0, timeslot 2 on board 16.

xds_ct_mix_remove

xds_ct_mix_remove(board_number, handle, ct_rcv_stream, ct_rcv_timeslot);
unsigned char board_number; a value indicating which board the command is for
int handle; a value indicating the conference handle to use
int ct_rcv_stream; a value indicating which CT bus stream is serving as an
 input
int ct_rcv_timeslot; a value indicating which CT bus timeslot is serving as an
 input

Applicable boards

H.100/H.110 boards with conferencing resources and the SCSA Enhanced conference board

Purpose

This function is used to remove an input from a “mix”. It will remove only the input from the mix with the specified handle. Any other mixes the timeslot may be an input to will not be affected.

Message Sent

“CXhhiiii” where **hh** is the mix handle and **iiii** is the CT bus input stream and timeslot

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_SLOT an illegal input and/or output timeslot was used for the board
ILL_HANDLE an invalid conference handle was chosen for the conference
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The **xds_ct_mix_remove** function is used to remove inputs from mixes created using the **xds_ct_mix_input** function. See the description of that function for details of how a mix works.

xds_ct_conference_mode() must be called before doing any conferencing and calling and calling this function. This will enable the user’s desired conference mode and enable proper error-checking in the library code.

Example

xds_ct_mix_remove(16, 14, 4); this will remove timeslot 14 from the mix using
 handle 4 on board 16.

xds_ct_mix_input

xds_ct_mix_input(board_number, handle, ct_rcv_stream, ct_rcv_timeslot, atten, threshold);

unsigned char board_number;	a value indicating which board the command is for
int handle;	a value indicating the conference handle to use
int ct_rcv_stream;	a value indicating which CT bus stream is serving as an input
int ct_rcv_timeslot;	a value indicating which CT bus timeslot is serving as an input
char atten;	a value between 0-7 that controls the input and output attenuation of the conference on the specified timeslots
char threshold;	a value between 0-3 specifying a noise threshold below which the input will not be added to a conference

Applicable boards

H.100/H.110 boards with conferencing resources and the SCSA Enhanced conference board

Purpose

This function adds an input to a conference “mix”. Using this function, a CT bus timeslot can serve as an input to several different mix outputs, each of which is a combination of a different set of inputs.

Message Sent

“Cihhiiiiat” where **iiii** is the CT bus input stream and timeslot, **hh** is the conference handle being used for the mix, **a** is the attenuation parameter and **t** is the noise threshold parameter.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_SLOT	an illegal input and/or output timeslot was used for the board
ILL_HANDLE	an invalid conference handle was chosen for the conference
ILL_ATTEN	an invalid attenuation value was chosen for the conference
ILL_THRESHOLD	an invalid threshold value was chosen for the conference
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

The conference facilities can be used to create a set of mixes where each mix consists of a different set of inputs, but where each input timeslot can serve as an input to one or more different mixes. Each mix is independent of the other mixes and inputs can be added or deleted at any time.

A mix output is established by using the **xds_ct_conference** or **xds_ct_conference_monitor** functions to create a conference output. Inputs are then added to the mix by using the **xds_ct_mix_input** function. Inputs may be removed from a mix by using the **xds_ct_mix_clear** function. Removing an input from one mix will not remove it from any other mixes it is a party to.

The **xds_ct_conference_dissolve** will disable the mix outputs and any inputs to the mix. It will not affect any inputs that may also be part of other mixes. Each mix input to each mix requires a separate Conference Control Address (CCA).

`xds_ct_conference_mode()` must be called before doing any conferencing and calling and calling this function. This will enable the user's desired conference mode and enable proper error-checking in the library code.

Example

```
xds_ct_mix_input(16, 14, 1, 4, 1, 0);
```

this adds stream 1, timeslot 4 on the CT bus as an input to the mix using handle 14. The attenuation parameter is 1 and the noise threshold parameter is 0.

xds_ct_conference_mode

xds_ct_conference_mode(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
char mode; a character indicating whether conferencing should be
 enabled 'E', enhanced enabled '2', and disabled 'D'

Applicable boards

H.100/H.110 boards with conferencing resources and the SCSA Enhanced conference board

Purpose

This function is used to enable or disable conferencing on the board.

Message Sent

“SKm” where **m** is the conferencing mode

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE an invalid conference mode was chosen

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

If conferencing is not enabled, other conference commands may not be successfully sent to the board. On the 256 port H.100 conference board, the user may enable 128 ports or 256 ports of conferencing. Mode 'E' enables 128 ports, and '2' enables 256 ports.

Example

xds_ct_conference_mode(16, 'E'); this enables conferencing on board 16

xds_ct_conference_dissolve

xds_ct_conference_dissolve(board_number, handle);

unsigned char board_number; a value indicating which board the command is for

int handle; a value indicating the conference handle to use

Applicable boards

All XDS boards with conferencing

Purpose

This function dissolves an entire conference. All inputs and outputs associated with the specified handle are disabled.

Message Sent

“CUhh” where **hh** is the handle number.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_HANDLE an invalid conference handle was chosen for the conference

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This function affects all parties to a conference. All inputs and outputs associated with the handle are disabled and the “CCA’s” are returned to the available pool. This function does not disable an output established using the **xds_ct_conference_monitor** function.

xds_ct_conference_mode() must be called before doing any conferencing and calling and calling this function. This will enable the user’s desired conference mode and enable proper error-checking in the library code.

Example

xds_ct_conference_dissolve(1, 3); this will dissolve the conference with a handle of 3.

xds_ct_conference_param

xds_ct_conference_param(board_number, handle, atten, threshold);

unsigned char board_number;	a value indicating which board the command is for
int handle;	a value indicating the conference handle to use
char atten;	a value between 0-7 that controls the input and output attenuation of the conference on the specified timeslots
char threshold;	a value between 0-3 specifying a noise threshold below which the input will not be added to a conference

Applicable boards

All XDS boards with conferencing

Purpose

This function is used to set the attenuation and noise threshold parameters of all parties to a conference at one time. This may be desirable when adding or deleting parties to a conference to optimize audio performance.

Message Sent

“SAhhat” where **hh** is the conference handle, **a** is the attenuation parameter, and **t** is the noise threshold parameter.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_HANDLE	an invalid conference handle was chosen for the conference
ILL_ATTEN	an invalid attenuation value was chosen for the conference
ILL_THRESHOLD	an invalid threshold value was chosen for the conference
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

As parties are added to a conference, the noise levels and feedback may increase to unacceptable levels. This affect can be reduced by adding attenuation to the input and output levels. The **xds_ct_conference_param** function may be used to globally control the attenuation parameters for a single conference. The amount of attenuation needed, and the number of parties to the conference for which additional attenuation should be added, are dependent on the specific characteristics of the inputs, and will have to be determined for each application. The **xds_ct_conference_param** function affects all inputs and outputs associated with a particular conference handle. It will also affect inputs to mixes established using the **xds_ct_mix_input** function.

`xds_ct_conference_mode()` must be called before doing any conferencing and calling and calling this function. This will enable the user's desired conference mode and enable proper error-checking in the library code.

Example

`xds_ct_conference_param(16, 4, 3, 0);` this will set the attenuation parameter to 3 and the noise threshold parameter to 0 for the conference with handle 4 on board 16.

xds_ct_conference_energy_det

xds_ct_conference_energy_det(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
int mode; a value of 'E' if detection is to be enabled or 'D' if
 detection is to be disabled

Applicable Boards

All XDS boards with conferencing

Purpose

This function will enable or disable the energy detection feature. This feature acts globally on all active conference inputs. The energy levels are reported in a table, which consists of one entry for each CT bus timeslot.

Message Sent

“SDD” if energy detection is to be disabled

“SDE” if energy detection is to be enabled

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE an invalid detection mode was chosen

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

Energy detection can be enabled simultaneously on all possible conference inputs. Energy detection can only be enabled on active inputs to conferences. If the input is removed from the conference using the **xds_ct_conference_disconnect** function or the conference is dissolved using the **xds_ct_conference_dissolve** function, detection will be disabled. The results of energy detection are reported in an array by using the **xds_query_energy_table** function.

Example

`xds_ct_conference_energy_det(1, 'E');` this will enable energy detection on board 1

xds_ct_query_energy_table

xds_ct_query_energy_table(board_number, &energy);

unsigned char board_number; a value indicating which board the command is for

unsigned char *array; a pointer to an array to place the energy values in

Applicable Boards

All XDS boards with **enhanced** conferencing

Purpose

If energy detection is enabled, this function will fill an array with the energy levels of active conference inputs. The table is arranged by CT bus timeslots.

Message Sent

none

Returns

This function will return the following values:

(-1)	the energy table flag was unable to be reset
SUCCESS	the energy table flag has been updated
NO_UPDATE	the energy table has no updated information for user / application
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

Energy levels are reported by the board in a table in dual-ported memory, one for each CT bus timeslot. A flag is used to indicate that the table has been updated. A value of 1 indicates an update. This function checks the flag, and if it is 0x1 it transfers the contents of the table to the array and sets the flag to 0x0. If the flag is 0, the function returns with a NO_UPDATE return code. The latter indicates that either energy detection has not been enabled using the **xds_ct_conference_energy_det** function, or not enough time has been allowed since the last attempt to read the table.

The energy levels reported will be in the range 0x0 to 0x1F where each step corresponds to 3 dB. If a CT bus timeslot is not an input to a conference, the energy value for that timeslot will be 0.

Example

xds_query_energy_table(1, &energy); this will transfer the energy level values to the *energy* array

xds_ct_conference_set_gain

xds_ct_conference_set_gain(board_number, ct_xmt_stream, polarity, gain);

unsigned char board_number;	a value indicating which board the command is for
int ct_xmt_stream;	the output stream used
int ct_xmt_timeslot;	the output timeslot used
char polarity;	a value of either '-' or '+', indicating the sign of the polarity for the gain
char gain;	a value between 0x0 and 0xF

Applicable boards

H.110 512 port conference board

Purpose

This function is used to set the output gain and polarity of a conference.

Message Sent

“SGssttab” where **ss** is the output stream, **tt** is the output timeslot, **a** is the sign of the polarity, and the **b** is the value of the gain.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE	an invalid polarity sign was chosen
ILL_ARG	an illegal gain value was selected
ILL_SLOT	an illegal stream and/or timeslot was used for the board
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This is the XDS Native style command for setting the output gain parameters on a conference. This will be used to control the gain of conferences created using the “CA”, “CM”, or “CI” commands.

Legal values for the polarity and gain are:

+0x1 to +0xF (+3 dB to +45 dB)

±0x0 (0 dB)

-0x1 to -0xE (-3 dB to -42 dB)

-0xF (-45 dB (silence))

Example

`xds_ct_conference_set_gain(2, 0, 1, '+', 3);` this will set the output gain on stream 0, timeslot 1 to +9 dB on board 2.

`xds_ct_conference_set_gain(2, 0, 1, '-', 1);` this will set the output gain on stream 0, timeslot 1 to -3 dB on board 2.

xds_ct_conference_control_gain

xds_ct_conference_control_gain(board_number, ct_xmt_stream, polarity, gain);
unsigned char board_number; a value indicating which board the command is for
int ct_xmt_stream; the output stream used
int ct_xmt_timeslot; the output timeslot used
char polarity; a value of either '-' or '+', indicating the sign of the polarity
 for the gain
char gain; a value between 0x0 and 0xF

Applicable boards

H.110 512 port conference board

Purpose

This function is used to set the output gain and polarity of a conference.

Message Sent

“SIsttab” where **ss** is the output stream, **tt** is the output timeslot, **a** is the sign of the polarity, and the **b** is the value of the gain.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE	an invalid polarity sign was chosen
ILL_ARG	an illegal gain value was selected
ILL_SLOT	an illegal stream and/or timeslot was used for the board
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This is the XDS Native style command for setting the input gain parameters on a conference. This will be used to control the gain of conferences created using the “CA”, “CM”, or “CI” commands.

Legal values for the polarity and gain are:

+0x1 to +0xF (+3 dB to +45 dB)

±0x0 (0 dB)

-0x1 to -0xE (-3 dB to -42 dB)

-0xF (-45 dB (silence))

Example

`xds_ct_conference_control_gain(2, 0, 1, '+', 3);` this will set the input gain on stream 0, timeslot 1 to +9 dB on board 2.

`xds_ct_conference_control_gain(2, 0, 1, '-', 1);` this will set the input gain on stream 0, timeslot 1 to -3 dB on board 2.

xds_ct_conference_distortion_control

xds_ct_conference_distortion_control(board_number, ct_xmt_stream, ct_xmt_timeslot, distortion_level);

unsigned char board_number; a value indicating which board the command is for
int ct_xmt_stream; the input stream used
int ct_xmt_timeslot; the input timeslot used
char distortion_level; an integer value for the distortion level

Applicable boards

H.110 512 port conference board *(with Distortion board firmware)

Purpose

This function is used to set the distortion level on an input stream and timeslot in a conference.

Message Sent

“CLiiiiI” where the first two **ii**’s are the input stream to be used, the second two **ii**’s are the input timeslot, and the **I** is the distortion level to use.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG an illegal distortion value was selected
ILL_SLOT an illegal stream and/or timeslot was used for the board
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The valid range for the distortion level is from 0 to 8. A value of 4 is the default and introduces no distortion. Decreasing the value from 4 will lower the pitch of the distorted party and increasing the value will raise the pitch of the distorted party. Thus, a value of 0 is the lowest pitch and 8 is the highest pitch.

This command can be used once a conference has been established.

Example

xds_ct_conference_distortion_control(2, 0, 1, 0); this will set the input gain on stream 0, timeslot 1 to the lowest level of distortion.

xds_ct_conference_distortion_control(2, 0, 1, 4); this will set the input gain on stream 0, timeslot 1 to the no distortion.

xds_ct_conference_dtmf_reporting_mode

xds_ct_conference_dtmf_reporting_mode(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
BOOLEAN mode; the reporting mode to be used

Applicable boards

H.110 512 port conference board

Purpose

This function is used to turn on or off the reporting of DTMF from the board to the user application.

Message Sent

“CLiiii” where the first two **ii**’s are the input stream to be used, the second two **ii**’s are the input timeslot, and the **I** is the distortion level to use.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This is a global command for the entire board. This command can not be used on a per conference basis.

A Boolean value of TRUE will turn on this feature, as FALSE will disable it.

Tone clamping itself can be enabled/disabled on a per-conference basis, but if six of the ten current conferences have tone clamping enabled, all six of these will either report the digits that are being clamped, or not, depending upon the use of this new command.

This will decrease message traffic from the board to the user application, therefore increasing performance and efficiency.

Example

`xds_ct_conference_dtmf_reporting_mode(2, TRUE);`

this will globally enable the reporting of DTMF detection from the board to the user application.

`xds_ct_conference_dtmf_reporting_mode(2, FALSE);`

this will globally disable the reporting of DTMF detection from the board to the user application.

This page was intentionally left blank.

Music On Hold
Audio Port Functions

This page was intentionally left blank.

xds_audio_disable_music_rcv

xds_audio_disable_music_rcv(board_number);
unsigned char board_number; a value indicating which board the command is for

Applicable boards

Infinity Series boards that have a music on hold audio port

Purpose

This function is used to disable the music on hold audio port output from the CT bus.

Message Sent

“ARX”

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The Infinity Series MC3 and Conference boards are equipped with an audio port that can operate in either the transmit or receive direction, i.e. output from the analog port to a CT bus timeslot or output from the CT bus to the analog port. This function disables output from the port to the bus.

Example

xds_audio_disable_music_rcv(16, 1); this will disable the audio port receive on board 16.

xds_audio_disable_music_xmt

xds_audio_disable_music_xmt(board_number);

unsigned char board_number; a value indicating which board the command is for

Applicable boards

Infinity Series boards that have a music on hold audio port

Purpose

This function is used to disable the music on hold audio port output to the CT bus.

Message Sent

“AD”

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The Infinity Series MC3 and Conference boards are equipped with an audio port that can operate in either the transmit or receive direction, i.e. output from the analog port to a CT bus timeslot or output from the CT bus to the analog port. This function disables output from the bus to the port.

Example

xds_audio_disable_music_xmt(16); this will disable the audio port transmit on board 16.

xds_audio_enable_music_rcv

xds_audio_enable_music_rcv(board_number, timeslot);

unsigned char board_number; a value indicating which board the command is for
int timeslot; a value indicating which timeslot on the CT bus is serving
 as an input to the audio port

Applicable boards

Infinity Series boards that have a music on hold audio port

Purpose

This function is used to enable the input from the CT bus to the music on hold audio port.

Message Sent

“ARsstt” where **sstt** is the CT bus stream and timeslot

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_SLOT an invalid timeslot value was called for the board being used

WRONG_BOARD an non-PCI based board was called

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The Infinity Series MC3 and Conference boards are equipped with an audio port that can operate in either the transmit or receive direction, i.e. output from the analog port to an CT bus timeslot or output from the CT bus to the analog port. This function enables an input from the CT bus to the port.

Example

xds_audio_enable_music_rcv(16, 27); this will enable input from timeslot 27 to the
 audio port on board 16.

xds_audio_enable_music_xmt

xds_audio_enable_music_xmt(board_number, timeslot);

unsigned char board_number; a value indicating which board the command is for
int timeslot; a value indicating which timeslot on the CT bus the audio
 port is outputting to

Applicable boards

Infinity Series boards that have a music on hold audio port

Purpose

This function is used to enable output from the music on hold audio port to the CT bus.

Message Sent

“AESstt” where **sstt** is the CT bus stream and timeslot.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_SLOT an invalid timeslot value was called for the board being used
WRONG_BOARD an non-PCI based board was called
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The Infinity Series MC3 and Conference boards are equipped with an audio port that can operate in either the transmit or receive direction, i.e. output from the analog port to an CT bus timeslot or output from the CT bus to the analog port. This function enables output from the port to the bus.

Example

xds_audio_enable_music_xmt(16); this will enable the audio port transmit on board 16.

xds_audio_music_atten

xds_audio_music_atten(board_number, xmt_attn, rcv_attn);

unsigned char board_number;	a value indicating which board the command is for
int xmt_attn;	a value between 0x00 and 0xFF indicating the transmit attenuation in .1 dB steps
int rcv_attn;	a value between 0x00 and 0xFF indicating the transmit attenuation in .1 dB steps

Applicable boards

Infinity Series boards that have a music on hold audio port

Purpose

This function is used to control the levels of the music on hold audio port.

Message Sent

“AGttr” where **tt** is the transmit attenuation and **rr** is the receive attenuation.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The Infinity Series MC3 and Conference boards are equipped with an audio port that can operate in either the transmit or receive direction, i.e. output from the analog port to an CT bus timeslot or output from the CT bus to the analog port. This function controls the audio levels in the transmit and receive directions. The amount of attenuation is controllable in .1 dB steps over a range of 0 dB to -25.5 dB.

Example

xds_audio_music_atten(16, 30, 0);	this will set the audio port transmit attenuation to 3 dB and the receive attenuation to 0 dB on board 16.
-----------------------------------	--

xds_audio_music_enable

xds_audio_music_enable(board_number, enable);

unsigned char board_number; a value indicating which board the command is for
char enable; a value of either '0' or '1' indicating the enable mode

Applicable boards

Infinity Series boards that have a music on hold audio port

Purpose

This function is the MVIP-compatibility mode function used to control the enabling and disabling of the music on hold audio port.

Message Sent

“MAx” where x is the enable mode to be used.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

The Infinity Series MC3 and Conference boards are equipped with an audio port that can be enable or disabled using this function.

Example

xds_audio_music_atten(16, 1); this will enable the audio port on board 16

H.100 / H.110
MVIP-compatibility
Commands

This page was intentionally left blank.

xds_ct_mvip_conference_det_dtmf

xds_ct_mvip_conference_det_dtmf(board_number, cca, mode, time);

unsigned char board_number; a value indicating which board the command is for
int cca; a value indicating which conference to enable or disable
 detection on
char mode; a value of either 'D' or 'E', indicating which mode to use
 for detection
int time; a value between 0 and 0xCF determining the duration of
 tone clamping to be used

Applicable boards

H.100/H.110 boards with conferencing resources and the MVIP MC3 / Enhanced Conference board

Purpose

This function is used to enable or disable DTMF detection on a "CCA".

Message Sent

"MDcccddd" where **ccc** is the CCA of the conference, **m** is the detection mode, and the **ddd** is the duration of detection.

or

"MDccddd" where **cc** is the CCA of the conference, **m** is the detection mode, and the **ddd** is the duration of detection.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE	an invalid detection mode was chosen
ILL_ARG	an illegal clamping time value was selected
ILL_CCA	an illegal CCA value was used for the board
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This is the "MVIP" style command for enabling DTMF detection on a specified "CCA".

xds_ct_conference_dtmf is also available for use.

Example

<code>xds_ct_mvip_conference_det_dtmf(16, 14, 'E', 0);</code>	this will disable DTMF detection on timeslot 14 from the mix using handle 4 on board 16.
---	--

xds_ct_mvip_conference

int xds_ct_mvip_conference(board_number, handle, cca, mode, atten, threshold);
unsigned char board_number; a value indicating which board the command is for
int handle; a value indicating the conference number
int cca; the selected conference control address to be used
char mode; the conference mode to be used
char atten; a value between 0-7 that controls the input and output
 attenuation of the conference on the specified timeslots
char threshold; a value between 0-3 specifying a noise threshold below
 which the input will not be added to a conference

Applicable Boards

H.100/H.110 boards with conferencing resources

Purpose

This will create an MVIP compliant conference command and send it to the board.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ATTEN invalid attenuation value was chosen
ILL_THRESHOLD invalid threshold value was chosen
ILL_HANDLE conference handle out of range was chosen
ILL_ARG invalid conference mode or combination of CCA and handle is illegal
ILL_MODE invalid conference mode
ILL_CCA conference control address out of range was chosen
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Message Sent

MKhhccmat – where the **hh** is the conference handle to be used, **cc** is the CCA, **m** is the mode, **a** indicates the attenuation value, and the **t** is the threshold value.

Comments

For a more detailed explanation of any of the conferencing limitations of your board, please consult the board's reference manual. A chart of available combinations of CCAs and handles is available on page 7-3 of the "*Infinity Series H.110 512/256 Port Conference Board*" manual (258M006). This will give the user an idea of how the conference blocks are set up. The following is a brief list of valid parameter ranges:

Example

`xds_ct_mvip_conference(16, 127, 384, 'E', 6, 2);` this will enable conferencing on board 16 with a handle of 127 and CCA of 384, with attenuation will be set to a value of 6 and the threshold set to 2

xds_ct_mvip_conference_output_disable

**int xds_ct_mvip_conference_output_disable(board_number, output_bus,
output_stream, output_timeslot);**

unsigned char board_number; a value indicating which board the command is for
char output_bus; a value indicating which bus to control
int output_stream; the selected output stream
int output_timeslot; the selected output timeslot

Applicable Boards

H.100/H.110 boards with conferencing resources

Purpose

This will disable output in a conference on the selected MVIP-95 terminus.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG illegal output bus chosen

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Message Sent

MObssttD – where the **b** is the terminus bus, **ss** is the stream, and **tt** is the timeslot.

Comments

This function can be used to send an MVIP-95 compliant conference disable message to an XDS Infinity Series conference board. For information on the MVIP-95 terminus and valid values, please refer to the MVIP-95 specification.

Example

xds_ct_mvip_conference(16, 'L', 0, 3); this will disable the conference output on stream 0,
timeslot 3 of the local bus on board 16

xds_ct_mvip_conference_output_enable

**int xds_ct_mvip_conference_output_enable(board_number, output_bus,
output_stream, output_timeslot, input_bus, input_stream, input_timeslot);**

unsigned char board_number;	a value indicating which board the command is for
char output_bus;	the output bus
int output_stream;	the selected output stream
int output_timeslot;	the selected output timeslot
char input_bus;	the input bus
int input_stream;	the selected input stream
int input_timeslot;	the selected input timeslot

Applicable Boards

H.100/H.110 boards with conferencing resources

Purpose

This will enable a conference on the selected CT bus, stream, and timeslot

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG illegal input and/or output bus chosen

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Message Sent

M**O**bsstEbsstt – where the first **b** is the output terminus bus, first **ss** is the output stream, and first **tt** is the output timeslot, and the second **b** is the input terminus bus, second **ss** is the input stream, and second **tt** is the input timeslot.

Comments

This function can be used to send an MVIP-95 compliant conference enable message to an XDS Infinity Series conference board. For information on the MVIP-95 terminus and valid values, please refer to the MVIP-95 specification.

Example

```
xds_ct_mvip_conference_output_enable(16, 'L', 0, 3, 'H', 0, 4);
```

This will conference input stream 0, timeslot 4 on the H.110 bus with the output stream 0, timeslot 3 of the local bus on board 16.

xds_ct_mvip_conference_output_pattern

int xds_ct_mvip_conference_output_pattern(board_number, output_bus, output_stream, output_timeslot, pattern);

unsigned char board_number;	a value indicating which board the command is for
char output_bus;	the output bus
int output_stream;	the selected output stream
int output_timeslot;	the selected output timeslot
int pattern	a pattern value

Applicable Boards

H.100/H.110 boards with conferencing resources

Purpose

This will output a pattern value on a selected MVIP-95 terminus.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG illegal input and/or output bus chosen

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Message Sent

MObssttPpp – where the **b** is the output terminus bus, **ss** is the output stream, and **tt** is the output timeslot, and **pp** is the pattern value to output

Comments

This function can be used to send an MVIP-95 compliant output pattern message to an XDS Infinity Series conference board. For information on the MVIP-95 terminus and valid values, please refer to the MVIP-95 specification.

Example

```
xds_ct_mvip_conference_output_enable(16, 'L', 0, 3, 55);
```

This will output the pattern “55” on output stream 0, timeslot 3 on the local bus of board 16.

xds_ct_mvip_line_det_dtmf

xds_ct_mvip_line_det_dtmf(board_number, detector, mode);
unsigned char board_number; a value indicating which board the command is for
int detector; the detector number to control
char mode; a value of either 'D' or 'E', indicating which mode to use
 for detection

Applicable boards

H.100/H.110 line interface boards

Purpose

This function is used to enable or disable DTMF detection on a line board.

Message Sent

“MDhhd” where **hh** is the detector and **d** is the mode of detection.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE an invalid detection mode was chosen
ILL_ARG an illegal detector was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This is the “MVIP” style command for enabling DTMF detection on a detector.

Example

xds_ct_mvip_line_det_dtmf(1, 2, 'E'); this will enable DTMF detection on
 detector 2 on board 1.

xds_ct_mvip_line_det_dtmf(1, 2, 'D'); this will disable DTMF detection on
 detector 2 on board 1.

xds_ct_mvip_send_dial_string

xds_ct_mvip_detect_energy(board_number, generator, dial_string);
unsigned char board_number; a value indicating which board the command is for
int generator; the generator number to control
char *dial_string; the dial string of digits to be sent

Applicable boards

H.100/H.110 line interface boards

Purpose

This function is used to enable or disable DTMF detection on a line board.

Message Sent

“MGhh(digits)” where **hh** is the generator and (**digits**) is the string of digits to be sent.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_ARG an illegal generator was selected

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This is the “MVIP” style command for enabling energy detection on a detector.

Example

```
xds_ct_mvip_detect_energy(1, 2, “0123”);
```

this will send the digits “0123” out from generator 2 on board 1 for 220 milliseconds.

xds_ct_mvip_conference_set_gain

xds_ct_mvip_conference_set_gain(board_number, cca, polarity, gain);

unsigned char board_number; a value indicating which board the command is for
int cca; a value indicating which conference to enable or disable
 detection on
char polarity; a value of either '-' or '+', indicating the sign of the polarity
 for the gain
char gain; a value between 0 and 0xF

Applicable boards

H.110 512 port conference board

Purpose

This function is used to set the output gain and polarity of a conference.

Message Sent

“MGcccab” where **ccc** is the CCA of the conference, **a** is the sign of the polarity, and the **b** is the value of the gain.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE	an invalid polarity sign was chosen
ILL_ARG	an illegal gain value was selected
ILL_CCA	an illegal CCA value was used for the board
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This is the “MVIP” style command for setting the gain parameters on a conference. This will be used to set the output gain of conferences created using the “MK” MVIP-compatible command. Note that in the ccc argument, the last two digits are the same as used in the MK command and the first digit is 0 if handles 01-54 were used and 1 if handles 55-A8 were used.

Legal values for the polarity and gain are:

+0x1 to +0xF (+3 dB to +45 dB)

±0x0 (0 dB)

-0x1 to -0xE (-3 dB to -42 dB)

-0xF (-45 dB (silence))

Example

`xds_ct_mvip_conference_set_gain(2, 1, '+', 3);` this will set the output gain on CCA 1 to +9 dB on board 2.

`xds_ct_mvip_conference_set_gain(2, 1, '-', 1);` this will set the output gain on CCA 1 to -3 dB on board 2.

xds_ct_mvip_conference_control_gain

xds_ct_mvip_conference_control_gain(board_number, cca, polarity, gain);

unsigned char board_number; a value indicating which board the command is for
int cca; a value indicating which conference to enable or disable
 detection on
char polarity; a value of either '-' or '+', indicating the sign of the polarity
 for the gain
char gain; a value between 0 and 0xF

Applicable boards

H.110 512 port conference board

Purpose

This function is used to control the input gain and polarity of a conference.

Message Sent

“MIccab” where **ccc** is the CCA of the conference, **a** is the sign of the polarity, and the **b** is the value of the gain.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE	an invalid polarity sign was chosen
ILL_ARG	an illegal gain value was selected
ILL_CCA	an illegal CCA value was used for the board
IOCTL	an IOCTL was called and returns a return value from the IOCTL call

Comments

This is the “MVIP” style command for setting the gain parameters on a conference. This will be used to control the gain of conferences created using the “MK” MVIP-compatible command. Note that in the ccc argument, the last two digits are the same as used in the MK command and the first digit is 0 if handles 01-54 were used and 1 if handles 55-A8 were used.

Legal values for the polarity and gain are:

+0x1 to +0xF (+3 dB to +45 dB)

±0x0 (0 dB)

-0x1 to -0xE (-3 dB to -42 dB)

-0xF (-45 dB (silence))

Example

```
xds_ct_mvip_conference_control_gain(2, 1, '+', 3);
```

this will set the gain on CCA 1 to +9 dB on board 2.

```
xds_ct_mvip_conference_control_gain(2, 1, '-', 1);
```

this will set the gain on CCA 1 to -3 dB on board 2.

xds_ct_mvip_dial_pulse_detect

xds_ct_mvip_dial_pulse_detect(board_number, port, mode);

unsigned char board_number; a value indicating which board the command is for
int port; the port to detect pulse on
char mode; the detection mode to be used, 'E' for enable, 'D' for
 disable

Applicable boards

H.100/H.110 line interface boards

Purpose

This function is used to enable or disable dial pulse detection on a line board.

Message Sent

“MPxxm” where **xx** is the port and **m** is the detection mode.

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE an illegal detection mode was selected
ILL_PORT an illegal port was selected
IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This is the “MVIP” style command for enabling pulse detection on a port.

Example

```
xds_ct_mvip_dial_pulse_detect(1, 2, 'E');
```

this will enable dial pulse detection on port 2 of board 1.

```
xds_ct_mvip_dial_pulse_detect(1, 2, 'D');
```

this will disable dial pulse detection on port 2 of board 1.

xds_ct_mvip_tristate

xds_ct_mvip_tristate(board_number, mode);

unsigned char board_number; a value indicating which board the command is for
char mode; the tristate mode to be used, 1 for enable, 0 for disable

Applicable boards

H.100/H.110 line interface boards

Purpose

This function is used to enable or disable dial pulse detection on a line board.

Message Sent

“MTD”

or

“MTE”

Returns

This function will return a 0 if successful. A non-zero return value indicates an error condition:

ILL_MODE an illegal tristate mode was selected

IOCTL an IOCTL was called and returns a return value from the IOCTL call

Comments

This is the “MVIP” style command for tristating the CT bus.

Example

```
xds_ct_mvip_tristate(1, 0);
```

this will disable CT bus output on board 1.

```
xds_ct_mvip_tristate(1, 1);
```

this will enable CT bus output on board 1.

This page was intentionally left blank.

XDS Receive Messages

(H.110 Library Version)

This page was intentionally left blank.

Received Message Types

Conference Errors (512/256 port board)

“EKxx01”

Handle Error - an illegal handle number was used

msg_type = 6, msg_sub = 1, port_number = -1, argument = handle, msg_str = message

“EKxx02”

No Conference Control Address - no free conference inputs

msg_type = 6, msg_sub = 2, port_number = -1, argument = handle, msg_str = message

“EKxx03”

No input to a conference

msg_type = 6, msg_sub = 3, port_number = -1, argument = handle, msg_str = message

State Change Messages (Line Boards Only)

“SBxx”

Busy Port - Port found busy in response to a **XDS_SEIZE**

msg_type = 1, msg_sub = 1, port_number = port, argument = 0, msg_str = message

“SCxx”

Connect Acknowledge - Port placed in the connect state by a connect message

msg_type = 1, msg_sub = 2, port_number = port, argument = 0, msg_str = message

“SDxxxxddd”

DID - DID address digits detected on a port

msg_type = 1, msg_sub = 3, port_number = port, argument = 0, msg_str = message

“SExx”

Tone String Ended - A DTMF string sent by **XDS_SEND_DTMF** has completed

msg_type = 1, msg_sub = 4, port_number = port, argument = 0, msg_str = message

“SFxx”

Off-Hook - a port has been detected going off hook

msg_type = 1, msg_sub = 5, port_number = port, argument = 0, msg_str = message

“SHxx”

Hold - a port has been put in the hold state by an **XDS_SEIZE**

msg_type = 1, msg_sub = 6, port_number = port, argument = 0, msg_str = message

“SIxx”

Idle - a port has been put in the idle state by an **XDS_RLS**

msg_type = 1, msg_sub = 7, port_number = port, argument = 0, msg_str = message

“SKxx”

Conference - a port has been conferenced using the local facilities

msg_type = 1, msg_sub = 8, port_number = port, argument = 0, msg_str = message

“SLxx”

Listen Acknowledge - a port has been placed in the listen state by **XDS_CPTONES** or **XDS_SEND_DTMF**

msg_type = 1, msg_sub = 9, port_number = port, argument = 0, msg_str = message

“SNxx”

On-Hook - a port has been detected going on-hook while seized or connected

msg_type = 1, msg_sub = 10, port_number = port, argument = 0, msg_str = message

“SQxx”

Hook-Flash- a hook flash has been detected on a station or E&M port

msg_type = 1, msg_sub = 11, port_number = port, argument = 0, msg_str = message

“SRxx”

Ring Detect - ringing has been detected on a loop or ground start port

msg_type = 1, msg_sub = 12, port_number = port, argument = cadence (in ASCII)

msg_str = message

“STxxt”

DTMF tone - a DTMF tone has been detected on a port set to listen by **XDS_LISTEN_DTMF**

msg_type = 1, msg_sub = 13, port_number = port, argument = tone (in ASCII)

msg_str = message

“SXxx”

Transmit Acknowledge - a port has been placed in the transmit state by **XDS_LISTEN_DTMF**

msg_type = 1, msg_sub = 14, port_number = port, argument = 0, msg_str = message

“SMxx”

MVIP Connect Acknowledge - a port has been placed in the MVIP connect state by

XDS_MVIP_CONNECT

msg_type = 1, msg_sub = 15, port_number = port, argument = 0, msg_str = message

“SEXh” or “SEXhh”

MVIP DTMF String Ended - a tone string generated using the **MG** message has completed returned message is of the form **SEXh** where h is the generator number

msg_type = 1, msg_sub = 16, port_number = generator, argument = 0, msg_str = message

“STXxt” or “STXxt”

MVIP DTMF Tone - a DTMF tone has been detected from a detector enabled using the “**MDE**” message, returned message is of the form **STXhy** where h is the detector and y the digit

msg_type = 1, msg_sub = 17, port_number = detector, argument = tone(in ASCII),

msg_str = message

“SPxXe”

Energy Detected - Energy has been detected from a detector enabled by

XDS_ENERGY_DETECT

msg_type = 1, msg_sub = 18, port_number = port, argument = 0 if energy ended, = 1 if detected,

msg_str = message

“SPXxe” or “SPXxe”

MVIP Energy Detected - Energy has been detected fro a detector enabled using the “**ME**”

message, returned message is of the form “**SPXhy**” where h is the detector and y the results

msg_type = 1, msg_sub = 19, port_number = detector, argument = 0 if energy ended, = 1 if

detected, msg_str = message

“SZxxS”

Non-existent Port Command Acknowledge - a state change command has been issued for a non-existent port

msg_type = 1, msg_sub = 20, port_number = port, argument = command, msg_str = message

Acknowledgements

“RA”

Reset All - acknowledges a **XDS_RESET_ALL**

msg_type = 2, msg_sub = 1, port_number = -1, argument = 0, msg_str = “RA”

“IA”

Interrupt On - acknowledges a **XDS_MSG_ON**

msg_type = 2, msg_sub = 2, port_number = -1, argument = 0, msg_str = “IA”

“A(msg)”

Message Acknowledge - acknowledges that a message has been received

msg_type = 2, msg_sub = 3, port_number = -1, argument = 0, msg_str = msg

“RD”

Reset DSP - acknowledges an **XDS_RESET_DSP**

msg_type = 2, msg_sub = 4, port_number = -1, argument = 0, msg_str = message “RD”

“RPxxpp”

Reset PORT - acknowledges an **XDS_RESET_PORT**

msg_type = 2, msg_sub = 5, port_number = port, argument = 0, msg_str = message

Clock Errors

“EC1”, “EC8”, “EC9”, “EC5”, “EC6”

Clock Failure - a clock failure has been detected

msg_type = 3, msg_sub = 1, port_number = -1, argument = 0, msg_str = message

“EC0” or “EC4”

Clock Restored - a clock restoration has been detected

msg_type = 3, msg_sub = 2, port_number = -1, argument = 0, msg_str = message

“EC2”

Next Clock - an XDS Bridge Board has become next clock on SCbus

msg_type = 3, msg_sub = 3, port_number = -1, argument = 0, msg_str = message

“EC3”

SCbus Master Failure - an XDS Bridge Board has failed to become SCbus clock master as another board was already master

msg_type = 3, msg_sub = 4, port_number = -1, argument = 0, msg_str = message

“EFr”

MC3 Ring Failure - an MC3 ring failure has been detected

msg_type = 3, msg_sub = 5, port_number = ring, argument = 0, msg_str = message

“ERr”

MC3 Ring Restored - an MC3 ring restoration has been detected

msg_type = 3, msg_sub = 6, port_number = ring, argument = 0, msg_str = message

“ESrss”

MC3 Ring Status - an MC3 ring status change has been detected

msg_type = 3, msg_sub = 7, port_number = ring, argument = status, msg_str = message

Argument Errors

“EPxxxx”

Path Error - the connection can not be made due to an existing connection in the opposite direction

msg_type = 4, msg_sub = 1, port_number = -1, argument = stream & timeslot, msg_str = message

Line Board Resource Error

“EDxx”

No DTMF Detector - All DTMF detectors were in use when one was requested

msg_type = 5, msg_sub = 1, port_number = port, argument = 0, msg_str = message

“EGxx”

No DTMF Generator - all DTMF generators were in use when one was requested

msg_type = 5, msg_sub = 2, port_number = port, argument = 0, msg_str = message

“ETxx”

DTMF DID Timeout - a timeout occurred while expecting DTMF DID address digits

msg_type = 5, msg_sub = 3, port_number = port, argument = 0, msg_str = message

“EWxx”

Wink Detect Timeout - a timeout occurred while expecting a wink on an outgoing call

msg_type = 5, msg_sub = 4, port_number = port, argument = 0, msg_str = message

Unparseable Message

“U(msg)”

Unparseable Message - an unparseable message was received by the board

msg_type = 8, msg_sub = 1, port_number = -1, argument = 0, msg_str = “U<message>”

Basic Rate ISDN Board Responses

“LCxxsstt” or “LRxxsstt”

Layer 3 Message Received

msg_type = 9, msg_sub = 1, port_number = -1, argument = TEI,

msg_str = message, augRxLen = layer 3 msg length, augRxMesg = layer 3 message

“DX”

Layer 3 “D” Message Received

MSG_TYPE = 9, msg_sub = 2, port_number = B channel, argument = Q.931 message type

msg_str = message

arg	message	Q.931 message type	additional information
1	DAxxps	ALERTing	progress indicator, signal

2	DCxxps	CONNect	progress indicator, signal
3	DCxxA	CONNect ACK	
4	DDxxccsrr	DISConnect	cause, signal, call reference
5	DPxxps	CALL PROCeeding	progress indicator, signal
6	DPxxPccps	PROGress	cause, progress indicator, signal
7	DRxxR(rr)	RELEase	
8	DRxxccrr	RELEase COMplete	cause, call reference
9	DSxxb(K)#	SETUP (overlap)	bearer capability, called number (NT only)
10	DSxxbC#	SETUP (enbloc)	bearer capability, called number (NT only)
11	DSxxbps#(/#)	SETUP	bearer capability, progress indicator, signal, calling #, optional called # (NT only)
12	DHxxrr	HOLD	call reference
13	DHxxArr	HOLD ACK	call reference
14	DHxxRccrr	HOLD REJect	cause, call reference
15	DGxxrr	RETriev	call reference
16	DGxxArr	RETriev ACK	call reference
17	DGxxRccrr	RETriev REJect	cause, call reference
18	DKxxkINFO,	keypad	keypad element
19	DFxxffINFO,	feature key	feature key
20	DFxxffa	INFO, feature ind.	feature indicator, status
21	DIxxCccdd	INFO, cause	
22	DIxxEuutt	INFO, endpoint	USID, TID (multiples?)
23	DIxxS#	INFO, SPID	SPID
24	DIxxTtext	INFO, text	text
25	DIxxLytext	INFO, line y text	text
26	DIxxQqq	INFO, request	requested item
27	DXxxccsrr	STATUS	cause, state, reference
28	DNxxn	NOTIFY	notification indicator
29	DYxx	REGister	
30	DIxxFee	INFO, fujitsu	element
31	DIxxA(SPID)	AT&T MIM	SPID
32	DExxArr	KEY SETUP ACK	call reference
33	DExxHrr	KEY HOLD	call reference
34	DExxRrr	KEY RELEase	call reference
35	DExxSrrb(/#)A=ca	KEY SETUP	
36	DIxxBcrr	INFO message	channel ID, call reference
37	DIxxPps	INFO message	progress ind. signal
38	DIxxRr#	INFO redirect	reason
39	DNxxT#	NOTIFY	connected number
40	DSxxApsrr	SETUP ACK	prog. ind., signal, call reference
41	DSxxrrbps#(/#)	SETUP	bearer capability, progress indicator, signal, calling #, optional called # (TE only)

42	DSxxb(D#)	SETUP	bearer cap, optional calling #, NT
43	DSxxbKk(D#)A=ca	SETUP	EKTS NT
44	DSxxbA=ca	SETUP	EKTS NT
45	DSxxbD#A=ca	SETUP	EKTS NT
46	DSxxbD#C#A=ca	SETUP	EKTS NT
47	DSxxbps#A=ca	SETUP	EKTS TE
48	DSxxrbpsA=ca	SETUP	EKTS TE
49	DDxx	DISconnect	EURO
50	DGxxRcc	RETreive REJect	EURO, cause
51	DGxxA	RESume ACK	EURO
52	DGxxRccrr	RESume REJect	EURO, cause, call reference
53	DHxxArr	HOLD ACK	call reference
54	DHxxA	SUSpend ACK	
55	DHxxRcc	HOLD REJect	cause
56	DIxxKkk	Info, with key	EURO, keypad digits
57	DIxxccKkk	Info with key & cause	EURO, cause, keypad digits
58	DKxxC(k)	Info sending complete	optional digits
59	DSxxb(p)	SETUP	bearer cap, optional prog ind.
60	DSxxbp(#)/C#	SETUP	bearer cap, prog ind., optional calling number, called number
61	DSxxbpKk,(#)/(#)	SETUP	bearer cap, prog ind., keypad digits optional calling number, optional called number
62	DSxxAp	SETUP ACK	progress indicator
63	DTxxTl(text)	TEXT line	line (1 or 2), text message
64	DZxxA0	REStart	
65	DIxxcc	Info with cause	cause
66	DSxxbp(#)/(#)	SETUP	bearer cap, prog ind., optional calling number, optional called number
67	DFxxFEidee	FACILITY	invoke ID, error
68	DFxxFlidopb(#)/#	FACILITY	invoke ID, operation, procedure, basic service, opt. forwarding #, subscriber #
69	DFxxFRid	FACILITY	invoke ID
70	DFxxQlido	FACILITY	invoke ID, operation
71	DFxxcrrid(tt)/(#)	FACILITY	component, call ref., invoke ID, optional tag value, optional number value
72	DHxxRccrr	HOLD REJect	EURO, cause, call reference
73	DHxxS(id)	SUSpend	EURO, optional call identity
74	DHxxrr	HOLD	EURO, call reference
75	DGxxArr	RETreive ACK	EURO, call reference
76	DGxxS(id)	RESume	EURO, optional ID
77	DGxxrr	RETreive	EURO, call reference
78	DDxxcc(p)(rr)	DISconnect	ERUO, cause, optional prog. ind., optional call reference

79	DNxxR	NOTIFY, call resumed	
80	DNxxS	NOTIFY, call suspended	
81	DNxxnn	NOTIFY	notification indicator
96	DBxxCrr	CONference	AT&T (NT), call reference
97	DBxxDrr	DROP	AT&T (NT), call reference
98	DBxxFrr	Feature ACTivation	AT&T (NT), feature
99	DBxxGrrA	Associated ACK	AT&T (NT), call reference
100	DBxxHrr	HOLD	AT&T (NT), call reference
101	DBxxMEuutt	Call Management	AT&T (NT), USID, TID
102	DBxxMS(spId)	Call Management	AT&T (NT), optional SPID
103	DBxxMO	Call Management	AT&T (NT)
104	DBxxRrr	Reconnect	AT&T (NT), call reference
105	DBxxSs	Switchhook	AT&T (NT), hook status
106	DBxxTrr	Transfer	AT&T (NT), call reference
107	DBxxA=ca(a)	Call Appearance	AT&T (NT), call appearance, optional adjunct ctrl.
108	DBxxCArr	CONference ACK	AT&T (TE), call reference
109	DBxxCRccrr	CONference REJECT	AT&T (TE), cause, call reference
110	DBxxDArr	DROP ACKnowledge	AT&T (TE), call reference
111	DBxxDRrr	DROP REJECT	AT&T (TE), call reference
112	DBxxFbmsff(i)	Feature Indication	AT&T (TE), button type, module, status type, feature number, opt. indicator status
113	DBxxGrrt(A=ca)	Associated Call	AT&T (TE), call reference, associated type, optional call appearance #
114	DBxxHArr	HOLD ACKnowledge	AT&T (TE), call reference
115	DBxxHRccrr	HOLD REJECT	AT&T (TE), cause, call reference
116	DBxxKrr	Redirect	AT&T (TE), call reference
117	DBxxMC	Call Management	AT&T (TE)
118	DBxxMS	Call Manage INFO	AT&T (TE)
119	DBxxRA	Reconnect ACK	AT&T (TE)
120	DBxxRRccrr	Reconnect REJECT	AT&T (TE), cause, call reference
121	DBxxTArr	Transfer ACK	AT&T (TE), call reference
122	DBxxTRccrr	Transfer REJECT	AT&T (TE), cause, call reference

“SM0” or “SM1”

Configuration Memory Response

MSG_TYPE = 9, msg_sub = 3, port_number = 0, argument = 0 or 1 msg_str = message

“TAxxt”

TEI Assigned

MSG_TYPE = 9, msg_sub = 4, port_number = port, argument = TEI msg_str = message

“TRxxt”

TEI Removed

MSG_TYPE = 9, msg_sub = 5, port_number = port, argument = TEI msg_str = message

“TQxx”

TEI Query Response

MSG_TYPE = 9, msg_sub = 6, port_number = port, argument = 0, msg_str = message

“ELxxe”

Layer 1 or Layer 2 Error

MSG_TYPE = 9, msg_sub = 7, port_number = port, argument = error, msg_str = message

“EMxxe”

Date Link Error

MSG_TYPE = 9, msg_sub = 8, port_number = port, argument = error, msg_str = message

“EDxxS”

SPID with illegal format received Error

MSG_TYPE = 9, msg_sub = 9, port_number = B-Channel, argument = 0, msg_str = message

“DQxx”

SPID/DN Query Response

MSG_TYPE = 9, msg_sub = 10, port_number = B-Channel, argument = 0, msg_str = message

“ETxxm”

U-Interface maintenance mode message

MSG_TYPE = 9, msg_sub = 11, port_number = port, argument = mode m, msg_str = message

“XMxxaabb”

U-Interface monitor message

MSG_TYPE = 9, msg_sub = 12, port_number = port, argument = aa, msg_str = message

Unknown Message

Unknown Message - an unknown message type was received from a board

msg_type = 0, msg_sub = 0, port_number = -1, argument = 0, msg_str = message

Query Messages

Unknown Response - an unknown message type on the query queue

msg_type = 16, msg_sub = 0, port_number = -1, argument = 0, msg_str = “U(message)”

“VC”

Board Version - a version request response

msg_type = 16, msg_sub = 1, port_number = -1, argument = 0, msg_str = message

“VD”

DSP Version - a DSP version request response

msg_type = 16, msg_sub = 2, port_number = -1, argument = 0, msg_str = message

“VA”

Alternate Version - an alternate segment version request response

msg_type = 16, msg_sub = 3, port_number = -1, argument = 0, msg_str = message

”QD”

Dump Switch - a query dump switch response message

msg_type = 16, msg_sub = 4, port_number = port, argument = 0, msg_str = message

“QI”

Sample Input - a sample input query response message

msg_type = 16, msg_sub = 5, port_number = port, argument = 0, msg_str = message

“QO”

Query Output - a query output response message

msg_type = 16, msg_sub = 6, port_number = port, argument = 0, msg_str = message

“QC” (all analog line boards, all ISA BRI boards, the PBX/Combo, and Bridge Board)

Query Switch Caps - a query switch caps response message msg_type = 16, msg_sub = 7,

port_number = port, argument = 0, msg_str = message

“QC” (all other XDS boards)

Query Clock - a query clock response message

msg_type = 16, msg_sub = 11, port_number = port, argument = 0, msg_str = message

“QP”

DSP Query - a DSP diagnostics query response message

msg_type = 16, msg_sub = 8, port_number = port, argument = 0, msg_str = message

“QR”

Query State - a query state response message from the Switch Matrix board

msg_type = 16, msg_sub = 9, port_number = stream & timeslot queried, argument = stream & timeslot of input, msg_str = message

Queue Errors

Full Receive Queue - the Receive queue was full

msg_type = 17, msg_sub = 1, port_number = -1, argument = 0, msg_str = “FULL QUEUE”
board_number = -1

Full Query Queue - the Query queue was full

msg_type = 17, msg_sub = 2, port_number = -1, argument = 0, msg_str = "FULL QUEUE"

board_number = -1

This page was intentionally left blank.