December 2001

FRAME RELAY SUPPORT FOR SANGOMA CARDS

Hardware Interface Manual

LIMITED USE LICENSE AGREEMENT

Sangoma Technologies Inc. provides the computer software program contained on the medium in this package (hereinafter called the Program) and licenses its use.

THE LICENSEE SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE ATTEMPTING TO USE THIS PRODUCT. INSERTION OF ANY OF THE DISKETTES IN THIS PACKAGE INTO ANY MACHINE INDICATES THE LICENSEE'S ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF THE LICENSEE DOES NOT AGREE WITH THE TERMS AND CONDITIONS, THE LICENSEE SHOULD PROMPTLY RETURN THE PACKAGE WITHIN 15 DAYS UNUSED AND UNCOPIED IN ANY WAY SHAPE OR FORM, AND MONIES WILL BE REFUNDED.

LICENSE:

a. The purchaser of this license (hereinafter called the Licensee) is granted a personal, non-exclusive license to use the Program in accordance with the terms and conditions set out in this agreement.

b. The Program may be used only on a single computer per license granted.

c. The Licensee and the Licensee's agents and employees shall protect the confidentiality of the Program and shall not distribute or make available the Program or documentation to any third party.

d. The Licensee may copy the programs into machine readable or printed form for backup or modification purposes only in support of the Licensee's use on a single machine. The Licensee must reproduce and include the copyright notice on any copy, modification or portion merged into another program.

e. Any portion of the Program merged into or used in conjunction with another program will continue to be subject to the terms and conditions of this agreement.

f. The Licensee may not assign or transfer the license or the program to any third party without the express prior consent of Sangoma Technologies Inc.

g. The licensee acknowledges that this license is only a limited license to use the Program and documentation, and that Sangoma Technologies Inc. retains full title to the program and documentation.

h. The Licensee shall not use, copy, modify or transfer the Program or documentation or any copy, modification or merged portion, in whole or in part, except as expressly provided for in this license. If the Licensee transfers possession of any copy, modification or merged portion of the program to a third party, the license is automatically terminated under this agreement.

TERM:

The license is effective until terminated. The licensee may terminate the license at any time by destroying the Program together with all copies, modifications and merged portion in any form. The Licensee agrees upon such termination to destroy the Program together with all copies, modifications and merged portion in any form.

LIMITED WARRANTY:

The Program is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the performance of the Program is with the Licensee. Should the Program prove defective, the Licensee (and not Sangoma Technologies Inc. or an authorized dealer) shall assume the entire cost of all necessary servicing, or correction. However, Sangoma Technologies Inc. warrants the diskettes on which the Program is furnished will be free of defects in materials or workmanship under normal use for a period of 90 days from the date of delivery to the Licensee. In no event will Sangoma Technologies Inc. be liable for any damages, including incidental or consequential damages arising out of the use or inability to use the Program, even if Sangoma Technologies Inc. or an authorized dealer have been advised of the possibility
of such damages, or for any claim by any other party.

The Licensee acknowledges that the Licensee has read this agreement, understands it, and agrees to be bound by its terms and conditions. The Licensee further agrees that it is the complete and exclusive statement of the agreement between the parties and supersedes any proposal or prior agreement, oral or written, and any other communications between the parties relating to the subject matter of this agreement.

1. Introduction

The Sangoma cards are general co-processor communication adapters capable of supporting any communication protocol autonomously, and providing information transfer into PC work space.

This document describes the support of the Frame Relay protocol on the card, and the hardware level PC to Card API required for development of device drivers. Where drivers have been developed, the details of the hardware interface are transparent to the user, and this document becomes a reference for command and return codes. The hardware/software solution handles the link Information Rates and local signalling autonomously, without PC intervention. The PC accesses the system as required to send or receive data, configure the link or to obtain statistics.

The implementation of Frame Relay on the S508 adapter includes the following features:

! ANSI T1.617, Annex D, LMI or Q.933 Local in-channel signalling.

! Up to 100 DLCIs are supported.

! A maximum information packet length of 15356 bytes.

! Line speeds up to 4Mbps.

! Control of transmission Information Rates on a per DLCI basis to maximize line utilization while keeping each DLCI data rate strictly inside the set parameters.

! Dynamic Frame Relay configuration facilities.

! Complete Frame Relay statistics package.

! A built in datascope with frame time stamping.

! Configurable as both a CPE (customer premises equipment) and an Access Node.

Conventions used in this manual

Programming conventions used are as follows:

Variables described with an **0x** prefix or an **h** suffix are hexadecimal values. All other variables are decimal.

For bit mapping, the **least significant** (**low**) bit is denoted as bit **0**.

2. Hardware

S503

This is a short 4 layer card, compatible with the ISA bus and it supports hardware interrupts as well as operating in a passive polled mode. The RS232 or V.35/X.21 interface is jumper selectable.

Clock speed:

This is factory set by Jumper **JP1**. Do not change without consulting your Sangoma dealer.

I/O port address:

This is set by Jumper **JP3**.

Pins 5-6	Pins 3-4	Pins 1-2	I/O Address Selection
Not Jumpered	Jumpered	Jumpered	250-252 (Hex)
Jumpered	Jumpered	Jumpered	254-256 (Hex)
Not Jumpered	Jumpered	Not Jumpered	300-302 (Hex)
Jumpered	Jumpered	Not Jumpered	304-306 (Hex)
Not Jumpered	Not Jumpered	Jumpered	350-352 (Hex)
Jumpered	Not Jumpered	Jumpered	354-356 (Hex)
Not Jumpered	Not Jumpered	Not Jumpered	360-362 (Hex)☎
Jumpered	Not Jumpered	Not Jumpered	364-366 (Hex)

☎Factory default.

IRQ Selection

The optional IRQ is set using **JP2**.

Pins 1-2	Pins 3-4	Pins 5-6	Pins 7-8	Pins 9-10	Selection
In	Out	Out	Out	Out	IRQ 2
Out	In	Out	Out	Out	IRQ 3
Out	Out	In	Out	Out	IRQ 4
Out	Out	Out	In	Out	IRQ 5
Out	Out	Out	Out	In	IRQ 7☎

☎Factory default.

Interface Level Selection

This is set by Jumper **JP3**.

Pins 9-10	Interface Level
Jumpered	RS-232
Not Jumpered	V.35

S514 PCI card

No jumpers need to be set on this card as it is configured by the PC BIOS.

S508 ISA Card

Jumpers **`JP1`** on the S508 define the card I/O address range as specified in Table 3-1. The specified card I/O addresses must not conflict with I/O addresses in use by any other hardware installed on the server. Use the **`SNOOPER`** utility if you are in any doubt as to hardware settings.

Note that **`JP1-1`** on the S508 is furthest to the left if the board is held such that the connectors are to the right. **`JP1-4`** is reserved.

The 8k (2000 Hex) byte shared memory address and the IRQ level are set in software for the S508.

Internal Line Clocking

For back-to-back connections, the cards can provide their own Transmit and Receive clock signals, which, with the appropriate cable, can also provide the clock for third party devices.

All cards are capable of generating the transmit and receive clocks as long as the appropriate back-to-back cable is used. The generated line speed is set by software.

However, the cards have a very large configurable range and therefore cannot easily be tabulated. When asked for the line speed during setup, you may specify any value in kbps from 1 to 2600. The actual generated line speed will be reasonably close the specified value, but will deviate more as the line speed increases.

S514 Port Pinouts

NB: Port PA is the Primary 4Mbps port
Port PB is the Secondary 512Kbps port.

PIN #	PA:RS232	PA:V.35	PB:RS232	PB:V.35
1	RTS		RTS	
2	CTS		CTS	
3	GND	GND	GND	GND
4	DCD		DCD	
5	DTR		DTR (V.10)	
6	TXD			
7	RXD			
8	TXC			
9	RXC			
10		RTS		RTS
11		CTS		CTS
12		DCD		DCD
13		DTR		DTR (V.10)
14		TXD		
15		RXD		
16		TXC		
17		RXC		

PIN #	PA:RS232	PA:V.35	PB:RS232	PB:V.35
18			TXA	
19			TXB	
20			RXA	
21			RXB	
22			TX Clock A	
23			TX Clock B	
24			RX Clock A	
25			RX Clock B	
26			DTR A (V.11)	
27			DTR B (V.11)	
28				TXA
29				TXB
30				RXA
31				RXB
32				TX Clock A
33				TX Clock B
34				RX Clock A
35				RX Clock B
36				DTR A (V.11)
37				DTR B (V.11)

S503/S508 Port Pinouts

RS232

Pin #	**Function**
2	TxD
3	RxD
7	GND
4	RTS
5	CTS
20	DTR
6	DSR
8	DCD
15	TxC
17	RxC
24	BxC

V.35/X.21

Pin #	**Function**
4	RTS
5	CTS
6	DSR
7	GND
8	DCD
10	TxA
9	TxB
12	RxA
11	RxB
19	Tx Clock A
20	DTR (V10 signal)
13	DTRA (V11 signal)
14	DTRB (V11 signal)
21	Tx Clock B
22	RI
23	Rx Clock A
25	Rx Clock B
18	Aux. Clock A (On board clock source)
16	Aux. Clock B (On board clock source)

3. The programmer's interface

Using the Frame Relay Shared Memory Interface

The SDLA card is operated by reading and writing structures to positions in the shared memory window. For details of moving the structures to/from the board, see the section “Example Code”.

The application program accesses the frame relay software by completing the required parameters within the control block defined below and then setting the OPP_FLAG. The SDLA processor will carry out the defined command and then update this control block with the required return code and, if applicable, the associated data buffer, data length etc. Once the command has been completed, the OPP_FLAG will be reset by the SDLA processor.

This control block structure is found at the physical offset **0xE000** on the adapter. For example if S508LOAD is executed with a command line definition of '- mDC', then the physical address of the control block is **0xDC00:0xE0000**.

The Shared Memory Control Block Structure

The control block structure is as follows:

Parameter	Offset	Lth	Remarks
OPP-FLAG	00H	1	A flag set by the application to inform the SDLA processor that a COMMAND is pending. This flag is in turn reset by the processor when the COMMAND has been completed.
COMMAND	01H	1	Command code.
BUFFER_LENGTH	02H	2	Length of the data buffer associated with this call.
RETURN_CODE	04H	1	Result of the previous command.
DLCI	05H	2	The Data Link Connection Identifier concerned with this command.
FECN_BECN_DE_CR_BITS	07H	1	FECN, BECN DE and C/R bits associated with the transmitted Information frame.
RESERVED	08H	8	Reserved for later use.
DATA	10H	1024	This is the transfer area for passing data associated with the various commands to and from the application level.

4. COMMAND Codes

The valid commands are:

0x01	INFORMATION_WRITE
0x03	ISSUE_IN_CHANNEL_SIGNALLING_FRAME
0x10	SET_DLCI_CONFIGURATION
0x11	READ_DLCI_CONFIGURATION
0x12	DISABLE_COMMUNICATIONS
0x13	ENABLE_COMMUNICATIONS
0x14	READ_DLC_STATUS
0x15	READ_DLC_STATISTICS
0x16	FLUSH_DLC_STATISTICS
0x17	LIST_ACTIVE_DLCIs
0x18	FLUSH_INFORMATION_BUFFERS
0x20	ADD_DLCIs
0x21	DELETE_DLCIs
0x22	ACTIVATE_DLCs
0x23	DEACTIVATE_DLCs
0x30	READ_MODEM_STATUS
0x31	SET_MODEM_STATUS
0x32	READ_COMMS_ERR_STATS
0x33	FLUSH_COMMS_ERR_STATS
0x40	READ_CODE_VERSION
0x41	DISCARD_INCOMMING_I_FRAMES
0x42	BRIDGE_TRANSMITTER_AND_RECEIVER
0x50	SET_INTERRUPT_TRIGGERS
0x51	READ_INTERRUPT_TRIGGERS
0x60	SET_TRACE_CONFIGURATION
0x61	READ_TRACE_STATISTICS

INFORMATION_WRITE (0x01)

Send data to the card for onward transmission to the network as an Information frame on the selected Data Link Connection Identifier (DLCI).

Control Block values to be set on entry:

DLCI: The DLCI associated with this Information frame.

BUFFER_
LENGTH The length of the data in the Information frame to be transmitted. The maximum data length is **15356** bytes and is dependent on the maximum I-frame length defined in the SET_DLCI_CONFIGURATION command OR the maximum number of bytes which may be transmitted in a single Tc interval.

FECN_BECN_
DE_CR_BITS: The status of the FECN, BECN, DE and C/R bits to be set in the transmitted I-frame (represented by bits 3, 2, 1 and 0 respectively).

If the Discard Eligibility (DE) Indicator is reset, then the I-frame will only be transmitted within the throughput limit defined by the Committed Information Rate (CIR). If this bit is set, then the throughput will be limited to the Uncommitted Information Rate.

For a device configured as Customer Premises Equipment (CPE), only the Discard Eligibility (DE) Indicator and the Command/Response (C/R) bits are valid. For a device configured as an Access Node, the Forward Explicit Congestion Notification (FECN), the Backward Explicit Congestion Notification (BECN) and the Command/Response (C/R) bits are valid.

DATA: The actual data to be transmitted.

Control Block values set on return:

RETURN_
CODE: 0x00 The data has been queued for transmission. Note that any return code other than **0x00** indicates that **no data was sent to the S508 card and this same data should be re-sent in its entirety.**

0x01 The ENABLE_COMMUNICATIONS command has not been executed and information transfer is not possible.

0x02 The channel is currently inoperative and information transfer is not possible.

0x03 The DLCI is currently inactive and information transfer is not possible on this particular DLCI.

0x04 The DLCI selected is invalid (i.e., it was not included in the DLCIs listed in the SET_DLCI_CONFIGURATION command).

0x05 The BUFFER_LENGTH set by the application exceeded the maximum I-frame length defined by the SET_DLCI_CONFIGURATION command. The BUFFER_LENGTH is set to the maximum permitted length.

0x06 The BUFFER_LENGTH set by the application exceeded the maximum number of bytes which may be transmitted in a single Tc interval. The returned BUFFER_LENGTH indicates the calculated maximum number of bytes that can be transmitted in a Tc period.

0x07 The current transmit throughput has exceeded that defined by the Committed Information Rate (CIR) or the Uncommitted Information Rate. The application is attempting to send data to the frame relay network at a rate faster than transmit throughput constraints dictate.

0x08 The data was not queued due to the fact that all the transmit buffers are currently occupied.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F

See Section "Notes on Return Codes" for further details.

DATA (valid if a **RETURN_CODE** of **0x00** is received):

Offset 0x00-0x04: a pointer to the Transmit Status Element to be used for this outgoing frame (see the section on Transmitting Information Frames for further details).

ISSUE_IN_CHANNEL_SIGNALLING_FRAME (0x03)

Issue a local In-channel Signalling message at a station configured as a CPE. Status enquiry messages are automatically issued by a CPE at a period of T391 seconds. However, the user may wish to force the transmission of these messages in cases such as recovery from an inoperative channel. This command permits the application to issue either a Full Status Enquiry or a Link Verification Enquiry message.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to 0x01.

DATA: Offset 0x00 indicates the type of Status Enquiry to be issued:

0x02 - issue a Link Verification Enquiry message.
0x03 - issue a Full Status Enquiry message.

Control Block values set on return:

RETURN_
CODE: 0x00 The action was performed successfully.

0x01 The ENABLE_COMMUNICATIONS command has not been executed and no frames may therefore be issued.

0x06 The type of Status Enquiry frame set at offset 0x00 in the DATA area is invalid.

0x07 A Status Enquiry message may not be issued at a station configured as an Access Node.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F

SET_DLCI_CONFIGURATION (0x10)

Set the frame relay code configuration. This command may be used as a global configuration command (on DLCI 0x00) or may be used to dynamically configure a specific DLCI.

After down loading the code to the adapter, a DISABLE_ COMMUNICATIONS and then a SET_DLCI_CONFIGURATION command (DLCI 0x00) should be issued. This SET_DLCI_ CONFIGURATION command configures 'global' parameters pertaining to the In-channel Signalling pragmatics such as T391 and N392, as well as DLCI-specific parameters such at the CIR, Bc and Be. Note that this command executed with the DLCI set to 0x00 applies the DLCI-specific configuration parameters to all DLCIs listed. Thereafter, if desired, each DLCI may be individually configured by using the SET_DLCI_CONFIGURATION command with the DLCI set to a specific (non-zero) value.

The use of this command with the DLCI set to zero also acts as a code 'reinitialization' - all transmit and receive buffers are flushed, statistics are reset and In-channel Signalling numbering and timers are zeroed. Note that a DISABLE_COMMUNICATIONS command should always be issued before a global SET_DLCI_CONFIGURATION command (DLCI 0x00) is used.

Control Block values to be set on entry:

DLCI: Set to **0x00** for a 'global' configuration or to a specific DLCI.

BUFFER_
LENGTH: For DLCI 0x00:

Set to **0x20** plus twice the number of DLCIs to be used by this station. Note that if automatic DLCI configuration is used at a CPE (bit 15 of the miscellaneous frame relay configuration bits is set), then the BUFFER_LENGTH should be set to 0x20.

For a non-zero DLCI:

Set to **0x0E**.

DATA: For DLCI **0x00** (all parameters are 2-byte, unsigned values):

Offset 0x00: the station configuration

0x00: Customer Premises Equipment (CPE).

0x01: Access Node (switch).

Offset 0x02: miscellaneous frame relay configuration bits as follows:

Bit 0: if reset, then the transmission data rates for Information frames will be kept within the limits set by the CIR, Bc and Be parameters. If set, then the transmission speed of Information frames will be restricted only by the physical baud rate and no CIR checking will be performed.

Bit 1: if reset, then no CIR checking will be performed on incoming Information frames. If set, then the throughput of the incoming Information frames will be compared against the defined backwards CIR and account will be kept of all frames exceeding this limit. If the station is configured as an Access Node, then all I-frames received in excess of the CIR will have the DE (Discard Eligibility) bit automatically set before passing to the application.

Bit 2: reserved for later use.

Bit 3: if reset, then Information frames passed to the board with an INFORMATION_WRITE command will not be queued if the Committed Burst Size or Excess Burst Size will be exceeded by the transmission of this frame. If set, then the Information frame will be stored in a transmit buffer (if there is a buffer available), irrespective of the current transmit throughput.

Bit 4: if reset, then throughput calculations will not be performed. If set, then transmit and receive throughput calculations will be performed for each DLCI and these results may be accessed by using the READ_DLC_STATISTICS command.

Bit 5: if reset, then exception condition return codes (0x10, 0x11, 0x12, 0x13 and 0x14) will be passed to the application on an INFORMATION_WRITE command. This bit may be set so as to disable the passing of these error codes when performing

INFORMATION_WRITE command, thus simplifying error processing in the application code, particularly in interrupt handlers.

Bit 6, 7: reserved for later use.

Bit 8: this bit should be set if the user is connecting to the MCI Frame Relay network.

Bits 9 - 11: reserved.

Bit 12: if this bit is set, then no In-Channel Signalling is used and Information frames may be transferred on all configured DLCIs after communications have been enabled.

Bits 13 - 14: the transmit/receive buffer ratios as follows:

If bits 13 and 14 are reset, then the transmit/receive buffer ratio is 50/50.

If bit 13 is set and bit 14 is reset, then the transmit/receive buffer ratio is 70/30.

If bit 13 is reset and bit 14 is set, then the transmit/receive buffer ratio is 30/70.

Bit 15: (valid only at a station configured as a CPE)

If reset, then a list of DLCIs to be handled by this station must be included in this SET_CONFIGURATION command. If set, then automatic DLCI configuration will occur, and the first100 DLCIs listed by the node will be configured. The application will be notified of theses DLCI additions by means of the return code 0x13.

Offset 0x04: the baud rate of the access line (kbps). This parameter must be set correctly even if external clocking is used, as this baud rate is used in CIR throughput checking. Valid values are from 1 to 2666 kbps.

Offset 0x06: the clocking source and electrical interface.

If bit 0 is reset, then external clocking is used.

If bit 0 is set, then internal clocking is used.

If bit 1 is reset, then the electrical interface is V.35.

If bit 1 is set, then the electrical interface is RS232.

Offset 0x08: the maximum data length of the Information frames to be transferred (valid entries from 300 to 15356 bytes).

Offset 0x0A: the T391 (Link Integrity Verification) timer. Valid entries are from 5 to 30 seconds and only pertain to a station configured as a CPE.

Offset 0x0C: the T392 (Polling Verification) timer. Valid entries are from 5 to 30 seconds and only pertain to a station configured as an Access Node.

Offset 0x0E: the N391 (Full Status Polling Cycle) counter. Valid entries are from 1 to 255 and only pertain to a station configured as a CPE.

Offset 0x10: the N392 (Error Threshold) counter. Valid entries are from 1 to 10 events and pertain to both a CPE and Access Node.

Offset 0x12: the N393 (Monitored Events) counter. Valid entries are from 1 to 10 events and pertain both a CPE and Access Node.

Offset 0x14: the forward Committed Information rate (CIR fwd). This is the rate (in kbps) at which the network agrees to transfer Information frames generated at this station under normal conditions. Valid values are between 1 and 512 kbps.

Offset 0x16: the forward Committed Burst Size (Bc fwd). This is the maximum amount of data generated at this station (in kbits) that the network agrees to transfer, under normal conditions, during a time interval Tc (the Committed Rate Interval). Valid values are between 1 and 512 kbits. In general, Bc fwd is set to equal the selected CIR parameter setting, resulting in a Tc of one second (Tc=Bc/CIR).

Offset 0x18: the forward Excess Burst Size (Be fwd). This is the maximum amount of uncommitted data generated at this station (in kbits) that the network will attempt to deliver during a time interval Tc. Valid values are between 0 and 511 kbits. The total of Bc plus Be should not exceed 512 kbits.

Offset 0x1A: the backward Committed Information rate (CIR bwd). This is the rate (in kbps) at which the network agrees to transfer Information frames generated at the remote CPE under normal conditions. Valid values are between 1 and 512 kbps.

Offset 0x1C: the backward Committed Burst Size (Bc bwd). This is the maximum amount of data generated at the remote CPE (in kbits) that the network agrees to transfer, under normal conditions, during a time interval Tc. Valid values are between 1 and 512 kbits.

Offset 0x1E: the backward Excess Burst Size (Be bwd). This is the maximum amount of uncommitted data generated at the remote station (in kbits) that the network will attempt to deliver during a time interval Tc. Valid values are between 0 and 511 kbits, but the total of Bc plus Be should not exceed 512 kbits.

Offset 0x20 - 0xN: a list of all the DLCIs to be handled by the station. Each DLCI is represented by a 2-byte unsigned value, and a maximum of 100 DLCIs may be listed. Valid assignments for DLCIs are from **16** to **991**.

Note that for a CPE station configured for automatic DLCI addition (bit 15 of the miscellaneous frame relay configuration bits set), no DLCIs should be listed here.

For a non-zero DLCI, the configuration data is as follows (all parameters are 2-byte, unsigned values):

Offset 0x00: miscellaneous frame relay configuration bits are:

Bit 0: if reset, then the transmission data rates for Information frames will be kept within the limits set by the CIR, Bc and Be parameters for this DLCI. If set, then the transmission speed of Information frames will be restricted only by the physical baud rate and no CIR checking will be performed.

Bit 1: if reset, then no CIR checking will be performed on incoming Information frames. If set, then the throughput of the incoming Information frames will be compared against the defined CIR bwd and account will be kept of all frames exceeding this limit.

Bit 2: Reserved

Bit 3: if reset, then Information frames passed to the board with an INFORMATION_WRITE command will not be queued if the Committed Burst Size or Excess Burst Size will be exceeded by the transmission of this frame. If set, then the Information frame will be stored in a transmit buffer (if there is a buffer available), irrespective of the current transmit throughput.

Bits 4 - 15: reserved.

Offset 0x02: the forward Committed Information rate (CIR fwd) for this DLCI. Valid values are between 1 and 512 kbps.

Offset 0x04: the forward Committed Burst Size (Bc fwd). Valid values are between 1 and 512 kbits.

Offset 0x06: the forward Excess Burst Size (Be fwd). Valid values are between 1 and 511 kbits, but the total of Bc plus Be should not exceed 512 kbps.

Offset 0x08: the backward Committed Information rate (CIR bwd). Valid values are between 1 and 512 kbps.

Offset 0x0A: the backward Committed Burst Size (Bc bwd). Valid values are between 1 and 512 kbits.

Offset 0x0C: the backward Excess Burst Size (Be bwd). Valid values are between 1 and 511 kbits, but the total of Bc plus Be should not exceed 512 kbps.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x01 Disable communications before executing this command on DLCI zero.

0x04 For a configuration command on DLCI 0x00:

An invalid DLCI was passed in the DLCI list, i.e., the DLCI was either zero or was outside the valid 16 to 991 range.

For a configuration command on a non-zero DLCI:

The DLCI selected is invalid (i.e., it was not included in the DLCIs listed in the global SET_DLCI_CONFIGURATION command).

0x05 The passed configuration data was of excessive length.

0x06 A configuration parameter was out of the valid range. The BUFFER_LENGTH indicates the offset within the DATA area of the invalid parameter.

0x07 The configured Bc plus Be is greater than the permitted maximum of 512 kbits.

0x08 The selected maximum data length is excessive for setting up the internal frame relay receive buffers. Offset 0x00-0x01 of the data area contains the permitted maximum data length for a successful configuration. If this returned data length is sufficient for the requirements of the application, then the SET_DLCI_CONFIGURATION should be re-issued using this value. Otherwise, decrease the transmit/receive buffer ratio in the miscellaneous frame relay configuration bits so as to increase the amount of receive buffer allocated.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

DATA (valid if a **RETURN_CODE** of **0x00** is received):

Offset 0x00-0x01: if internal clocking has been selected, then the actual generated baud rate may be different from the configured baud rate and is returned to the user in the mailbox data area.

READ_DLCI_CONFIGURATION (0x11)

Read the current configuration of the frame relay code. This command may be used to read the global configuration (on DLCI 0x00) or to read the configuration of a specific DLCI.

Control Block values to be set on entry:

DLCI: Set to **0x00** to read the 'global' configuration or set to a specific DLCI to read the configuration of that particular connection.

BUFFER_
LENGTH: set to 0x00.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x04 An invalid DLCI was selected.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

BUFFER_LENGTH (valid if a **RETURN_CODE** of **0x00** is received):

For DLCI 0x00: Set to **0x20** plus twice the number of DLCIs configured for use by this station.

For a non-zero DLCI: Set to **0x16**.

DATA (valid if a **RETURN_CODE** of **0x00** is received):

For DLCI **0x00**, the configuration data is returned in exactly the same format as for the SET_DLCI_CONFIGURATION (DLCI zero) described above. However, for a CPE station configured for automatic DLCI addition (bit 15 of the miscellaneous frame relay configuration bits set), offset 0x20 to 0xNN will list the DLCIs thus far automatically configured.

For a non-zero DLCI, the configuration data is returned in the same format as for the SET_DLCI_CONFIGURATION (non-zero DLCI) described above, with the following additions (all parameters are 2-byte, unsigned values):

Offset 0x0E: the forward Committed Rate Measurement Interval (Tc fwd, in seconds) calculated for this station.

Offset 0x10: the backward Committed Rate Measurement Interval (Tc bwd, in seconds) calculated for this station.

Offset 0x12: the maximum number of bytes that may be transmitted in a single Tc interval. If forward CIR checking is enabled, this value defines the maximum number of data bytes that can be passed by the INFORMATION_WRITE command.

0x14: the maximum number of bytes that may be received by the adapter in a single Tc interval and still remain inside the CIR bwd limit.

DISABLE_COMMUNICATIONS (0x12)

This command causes the frame relay code to no longer send frames or to respond to incoming frames.

Note that this command must be executed before using the SET_DLCI_CONFIGURATION command on DLCI zero.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to 0x00.

Control Block values set on return:

RETURN_
CODE: 0x00 The action was performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

ENABLE_COMMUNICATIONS (0x13)

This command allows the frame relay code to transmit frames and to respond to incoming frames.

DTR will be automatically raised once the **ENABLE_COMMUNICATIONS** command has been issued.

Note that after down loading the code to the adapter, the communications may not be enabled until the initial SET_DLCI_CONFIGURATION (DLCI zero) command has been issued.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to 0x00.

Control Block values set on return:

RETURN_
CODE: 0x00 The action was performed successfully.

0x01 The communications may not be enabled until the initial SET_DLCI_CONFIGURATION (DLCI zero) command has been issued.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

READ_DLC_STATUS (0x14)

Read the current status of all the configured DLCIs or return the status of a selected DLCI.

Control Block values to be set on entry:

DLCI: Set to **0x00** to read the status of all DLCIs or set to a specific DLCI to read the status of that particular connection.

BUFFER_
LENGTH: set to 0x00.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x04 An invalid DLCI was selected.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

BUFFER_LENGTH (valid if a **RETURN_CODE** of **0x00** is received):

For DLCI 0x00:

Set to **0x01** plus three times the number of DLCIs configured for use by this station.

For a non-zero DLCI:

Set to **0x01**.

DATA (valid if a **RETURN_CODE** of **0x00** is received):

For DLCI **0x00**, the status data is returned as follows:

Offset 0x00: The channel status.

0x00 - the channel is inoperative

0x01 - the channel is operative

Offset 0x01 - 0xNN: A list of the configured DLCIs and their associated status bytes in the following three byte format:

The first two bytes represent the DLCI and the next byte is the status of that DLCI as follows:

Bit 0 - if set, the DLCI has been deleted.

Bit 1 - if set, then the DLCI is active and information transfer is possible.

Bit 2 - if set, then the Access Node is waiting to issue a Full Status message before denoting the DLCI as being active (pertains only to a device configured as an Access Node).

Bit 3 - if set, then the DLCI is new.

Bit 6 - if set, then the DLCI was included in the last received (CPE) or transmitted (Access Node) Full Status message.

For a non-zero DLCI, the first byte of the returned DATA area represents the status of the DLCI. The bit settings for this status byte are the same as that returned for the READ_DLC_STATUS call on DLCI zero.

READ_DLC_STATISTICS (0x15)

Read the current performance statistics for this station.

Control Block values to be set on entry:

DLCI: Set to **0x00** to read the 'global' statistics or set to a specific DLCI to read the performance statistics of a particular connection.

BUFFER_
LENGTH: Set to **0x00** if the selected DLCI is ZERO, otherwise set to **0x01**.

DATA: Offset 0x00 - only valid for a non-zero DLCI setting as follows:

Set to 0x00 - the throughput statistics are not passed to the application.

Set to 0x01 - calculate and pass the throughput statistics to the application.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x04 An invalid DLCI was selected.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

BUFFER_LENGTH (valid if a **RETURN_CODE** of **0x00** is received):
For DLCI 0x00: Set to **0x38**.

For a non-zero DLCI: Set to **0x20** (no throughput statistics included) or **0x30** (statistics included).

DATA (valid if a **RETURN_CODE** of **0x00** is received):

For DLCI **0x00**, the statistics are returned as follows (all parameters are **2-byte, unsigned values**):

Offset Parameter Meaning

0x00 number of I-frames which were not transmitted (and discarded) after a non-DLCI specific transmit interrupt for one of the following reasons:

a) the selected DLCI was unconfigured..

b) the selected DLCI was inactive..

c) the passed frame length was invalid.

0x02 number of I-frames which were not transmitted (and discarded) after a non-DLCI specific transmit interrupt, as the CIR would have been exceeded by the transmission of this frame.

0x04 number of frames received of invalid length.

0x06 number of Information frames received with a DLCI not matching one of the DLCIs configured at this station. These frames have been discarded.

0x08 number of received Information frames discarded due to a format error.

0x0A number of I-frames which were discarded due to insufficient transmit buffering when operating in the bridging mode.

0x0C number of Full Status Enquiry messages sent (pertains to CPE only).

0x0E number of Link Integrity Verification Status Enquiry messages sent (pertains to CPE only).

0x10 number of Full Status messages received (pertains to CPE only).

0x12 number of Link Integrity Verification Status messages received (pertains to CPE only).

0x14 number of Full Status Enquiry messages received (pertains to an Access Node only).

0x16 number of Link Integrity Verification Status Enquiry messages received (pertains to an Access Node only).

0x18 number of Full Status messages sent (pertains to an Access Node only).

0x1A number of Link Integrity Verification Status messages sent (pertains to an Access Node only).

0x1C number of received In-channel Signalling frames discarded due to a format error.

0x1E number of unsolicited responses from the Access Node.

0x20 number of invalid Send Sequence Numbers received.

0x22 number of invalid Received Sequence Numbers received.

0x24 number of timeouts on the T391 timer (pertains to CPE only).

0x26 number of timeouts on the T392 timer (pertains to an Access Node only).

0x28 number of times that the N392 error threshold was reached during N393 monitored events.

0x2A number of CPE initializations (Send and Receive Sequence Numbers reset).

0x2C the current Send Sequence Number.

0x2E the current Receive Sequence Number.

0x30 the number of consecutive timeouts on the T391 timer (pertains to CPE only).

0x32 the number of consecutive timeouts on the T392 timer (pertains to an Access Node only).

0x34 the current N392 count.

0x36 the current N393 count.

For a non-zero DLCI, the statistics are returned as follows (all parameters are **4-byte, unsigned long values**) and pertain only to the specified DLCI:

Offset Parameter Meaning

0x00 number of Information frames transmitted.

0x04 number of Information bytes transmitted.

0x08 number of Information frames received.

0x0C number of Information bytes received.

0x10 number of I-frames which were not transmitted (and discarded) after a non-DLCI specific transmit interrupt, as the CIR would have been exceeded by the transmission of this frame.

0x14 number of received Information frames discarded, as the specified DLCI was inactive.

0x18 number of Information frames received with excessive throughput, i.e. the actual Bc or Be was found to exceed the configured values.

0x1C number of Information frames received with the Discard Eligibility (DE) indicator set.

0x20 the current transmit throughput (bps).

0x24 the number of milliseconds expired since the transmit throughput measurement was activated by the first outgoing I-frame.

0x28 the current receive throughput (bps).

.

0x2C the number of milliseconds expired since the receive throughput measurement was activated by the first incoming I-frame.

FLUSH_DLCI_STATISTICS (0x16)

The current values of the variables accessed by the **READ_DLC_STATISTICS** command are reset to zero. Note that if the 'global' statistics are flushed, then the active statistics such as the current Send and Receive Sequence Numbers, the number of consecutive timeouts on the T391/T392 timer and the current N392/N392 counts will not be flushed.

Control Block values to be set on entry:

DLCI: Set to **0x00** to flush the 'global' statistics or to flush the statistics for all the configured DLCIs, or set to a specific DLCI to flush the performance statistics of a particular connection.

BUFFER_
LENGTH: Set to **0x01** if the DLCI is set to zero.
Set to **0x00** if the DLCI is set to a non-zero value.

DATA: Offset 0 - only required if the DLCI is set to zero.

Indicates the type of statistics flushing to be performed as follows.

bit 0 set - flush the global statistics.

bit 1 set - flush the statistics for all the configured DLCIs.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x04 An invalid DLCI was selected.

0x06 The bit settings passed at offset 0x00 in the DATA area are invalid.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F

See Section "Notes on Return Codes" for further details.

LIST_ACTIVE_DLCIs (0x17)

Return a list of all the DLCIs which have been marked as active in the last Full Status Report transmitted by the Access Node. Information transfer may only take place on active DLCIs.

Note that for a device configured as an Access Node, the DLCIs will only be marked as being active once the status of these DLCIs has been reported to the CPE by means of a transmitted Full Status message.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to 0x00.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

BUFFER_
LENGTH (valid if a **RETURN_CODE** of **0x00** is received):
Set to twice the number of active DLCIs.

DATA (valid if a **RETURN_CODE** of **0x00** is received):

Offset 0x00 - 0xNN: a list of all active DLCIs. Each DLCI is represented by a 2-byte unsigned hexadecimal value, and a maximum of 100 DLCIs will be listed.

FLUSH_INFORMATION_BUFFERS (0x18)

Flush (discard) all queued Information buffers.

If DLCI zero is selected, then all queued Information frames (both incoming and outgoing) for all configured DLCIs are flushed. If a specific DLCI is selected, then only the transmit buffers for that DLCI are flushed (no buffered received Information frames are cleared).

Control Block values to be set on entry:

DLCI: Set to **0x00** to flush all transmit and receive buffers, or set to a specific DLCI to flush the transmit buffers for that DLCI.

BUFFER_
LENGTH: Set to 0x00.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x04 An invalid DLCI was selected.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

ADD_DLCIs (0x20)

This command pertains only to a station configured as an Access Node and is used to define the DLCIs which will be returned to the CPE in a Full Status message.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to twice the number of DLCIs to be added.

DATA: Offset 0x00 - 0xNN: a list of all DLCIs to be added. Each DLCI is represented by a 2-byte unsigned hexadecimal value, and a maximum of 100 DLCIs may be listed. Note that these DLCIs must have been defined in the SET_DLCI_CONFIGURATION command.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x04 An invalid DLCI was listed.

0x05 The length of the DLCI list was excessive.

0x07 The command is invalid for a station configured as a CPE.

0x08 A listed DLCI has already been 'added' and is already included in the Full Status message.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

DELETE_DLCIs (0x21)

This command pertains only to a station configured as an Access Node and is used to delete DLCIs which are currently included in a Full Status message.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to twice the number of DLCIs to be deleted.

DATA: Offset 0x00 - 0xNN: a list of all DLCIs to be deleted. Each DLCI is represented by a 2-byte unsigned hexadecimal value, and a maximum of 100 DLCIs may be listed. Note that these DLCIs must have been previously 'added' by using the ADD_DLCIs command.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x04 An invalid DLCI was listed.

0x05 The length of the DLCI list was excessive.

0x07 The command is invalid for a station configured as a CPE.

0x08 A listed DLCI has not been 'added' and cannot therefore be 'deleted'.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

ACTIVATE_DLCIs (0x22)

This command pertains only to a station configured as an Access Node and is used to define which DLCIs will be reported as 'active' in the next returned Full Status message.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to twice the number of DLCIs to be activated.

DATA: Offset 0x00 - 0xNN: a list of all DLCIs to be activated. Each DLCI is represented by a 2- byte unsigned hexadecimal value, and a maximum of 100 DLCIs may be listed. Note that DLCIs must have been 'added' before being 'activated'.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x04 An invalid DLCI was listed.

0x05 The length of the DLCI list was excessive.

0x07 The command is invalid for a station configured as a CPE.

0x08 A listed DLCI has not been 'added' and may therefore not be 'activated'.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

DEACTIVATE_DLCIs (0x23)

This command pertains only to a station configured as an Access Node and is used to define which 'active' DLCIs will be no longer be reported as 'active' in the next returned Full Status message.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to twice the number of DLCIs to deactivate.

DATA: Offset 0x00 - 0xNN: a list of all DLCIs to be deactivated. Each DLCI is represented by a 2- byte unsigned hexadecimal value, and a maximum of 100 DLCIs may be listed.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x04 An invalid DLCI was listed.

0x05 The length of the DLCI list was excessive.

0x07 The command is invalid for a station configured as a CPE.

0x08 A listed DLCI has not been 'added' and may therefore not be 'deactivated'.

0x10, 0x11, 0x12, 0x13, 0x14, 0x1F
See Section "Notes on Return Codes" for further details.

READ_MODEM_STATUS (0x30)

Read the current CTS and DCD status.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to **0x00**.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14
See Section "Notes on Return Codes" for further details.

BUFFER_LENGTH (valid if a **RETURN_CODE** of **0x00** is received):
Set to **0x01**.

DATA (valid if a **RETURN_CODE** of **0x00** is received):

Offset 0x00: The current CTS and DCD status as follows:

If bit 3 is set, then DCD is high.

If bit 5 is set, then CTS is high.

SET_MODEM_STATUS (0x31)

Set the status of DTR and RTS.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to 0x01.

DATA: Offset 0x00 defines the DTR and RTS settings.

If bit 0 is set, then DTR will be set high.

If bit 1 is set, then RTS will be set high.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14
See Section "Notes on Return Codes" for further details.

READ_COMMS_ERR_STATS (0x32)

Retrieve the communications error statistics for the link.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to **0x00**.

Control Block values set on return:

RETURN_
CODE: 0x00 The action was performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14
See Section "Notes on Return Codes" for further details.

BUFFER_
LENGTH (valid if a **RETURN_CODE** of **0x00** is received):
Set to **0x0A**.

DATA (valid if a **RETURN_CODE** of **0x00** is received):

Offset 0x00:	number of receiver overrun errors.
Offset 0x01:	number of receiver CRC errors.
Offset 0x02:	number of abort frames received.

Offset 0x03: number of times that the receiver had to be disabled due to inadequate receive buffering.

Offset 0x04: reserved for later use.

Offset 0x05: number of abort frames transmitted.

Offset 0x06: number of missed transmit underrun interrupts.

Offset 0x07: reserved for later use.

Offset 0x08: number of times DCD changed state.

Offset 0x09: number of times CTS changed state.

FLUSH_COMMS_ERR_STATS (0x33)

The current values of the variables accessed by the **READ_COMMS_ERR_STATS** command are reset to zero.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to **0x00**.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14
See Section "Notes on Return Codes" for further details.

READ_CODE_VERSION (0x40)

Return the code version of the frame relay code.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to **0x00**.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14
See Section "Notes on Return Codes" for further details.

BUFFER_
LENGTH: Set to **0x04** if a **RETURN_CODE** of **0x00** is received.

DATA (valid if a **RETURN_CODE** of **0x00** is received):

The code versions are of the ASCII format: main-version.sub-version

Offset 0x00-0x03: the frame relay code version.

DISCARD_INCOMMING_I_FRAMES (0x41)

Configure the adapter to discard or store received Information frames.

By default, the adapter is set to store all valid incoming I-frames and make them available for the application once communications have been enabled. If the application is not able to process these queued frames, then this may result in the receiver on the adapter being temporarily disabled to avoid buffer overruns. Therefore, the application may wish to temporarily discard incoming I-frames if the host PC is unable to process them.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to 0x01.

DATA: Offset 0x00 is set as follows:

0x00 - buffer incoming I-frames for passing to the application.
0x01- discard incoming I-frames.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14
See Section "Notes on Return Codes" for further details.

BRIDGE_TRANSMITTER_AND_RECEIVER (0x42)

Configure the adapter to automatically transmit all Information frames which are received.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to 0x01.

DATA: Offset 0x00 is set as follows:
0x00 - do not bridge the transmitter and receiver.
0x01- bridge the transmitter and receiver.

Control Block values set on return:

RETURN_
CODE: 0x00 The action has been performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14
See Section "Notes on Return Codes" for further details.

SET_INTERRUPT_TRIGGERS (0x50)

Set the occurrences which will cause the SDLA adapter to trigger a hardware interrupt on the PC.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to 0x03.

DATA: Offset 0x00 defines the interrupt triggers as follows:

bit 0 - the receive interrupt bit.

If this bit is set, then an interrupt will be triggered if there is an incoming Information frame available for reception by the application.

bit 1 - the transmit interrupt bit.

The transmit interrupt may be used in two different ways:

a) DLCI specific transmit interrupt, where an interrupt will be triggered if an Information frame of a specified length may be transmitted on a specific DLCI. The length of the frame to be transmitted is defined at offset 0x01 to 0x02 of the DATA area of this control block, and the DLCI is specified in the DLCI definition area.

b) non-DLCI specific transmit interrupt, where an interrupt will be triggered if at least one transmit buffer is available on the adapter. DLCI zero is specified in the DLCI definition area.

bit 2 - the modem status interrupt.

If this bit is set, then an interrupt will be triggered when a change in the state of CTS or DCD occurs. The details of this modem status change may be established by using a READ_DLC_STATUS command to illicit a return code of 0x10 and then examining the byte at offset 0x00 in the structure data area.

bit 3 - the ‘command complete’ interrupt bit.

If bit 3 is set, then an interrupt will be triggered on completion of an interface command, i.e. when the 'opp_flag' has been reset.

bit 4 - the channel/DLCI status interrupt bit.

If bit 4 is set, then an interrupt will be triggered on a change in status of the channel or a DLCI. The details of this status change may be established by using an READ_DLC_STATUS command to illicit a return code of 0x11, 0x12, 0x13 or 0x14 as documented in the section "Notes on Return Codes and Status Bytes".

bits 5, 6, 7 - reserved for later use.

Offset 0x01-0x02: (only applicable when enabling transmit interrupts in the DLCI-specific mode) - the length of the outgoing frame for which a transmit interrupt should be triggered. Note that this parameter is represented by a two byte, unsigned short variable.

DLCI: used in conjunction with transmit interrupts and is set to zero to enable non DLCI specific interrupts or to a specific DLCI on which the Information frame is to be transmitted.

Control Block values set on return:

RETURN_

CODE: 0x00 The action has been performed successfully.

0x04 (only applicable when enabling transmit interrupts) - the DLCI selected is invalid (i.e., it was not included in the DLCIs listed in the SET_DLCI_CONFIGURATION command).

0x05 (only applicable when enabling transmit interrupts) - the length of the frame to be transmitted exceeded the maximum I-frame length defined by the SET_DLCI_CONFIGURATION command.

0x06 (only applicable when enabling transmit interrupts) - the length of the frame to be transmitted exceeded the maximum number of bytes which may be transmitted in a single Tc interval.

0x10, 0x11, 0x12, 0x13, 0x14
See Section "Notes on Return Codes" for further details.

READ_INTERRUPT_TRIGGERS (0x51)

Read the current interrupt trigger configuration as set in the SET_INTERRUPT_TRIGGERS command.

Control Block values to be set on entry:

BUFFER_
LENGTH: Set to 0x00.

Control Block values set on return:

RETURN
_CODE: 0x00 The action has been performed successfully.

0x10, 0x11, 0x12, 0x13, 0x14
See Section "Notes on Return Codes" for further details.

BUFFER_
LENGTH: Set to 0x03 if a RETURN_CODE of 0x00 is received.

DATA: The interrupt trigger configuration as defined in the SET_INTERRUPT_TRIGGERS command.

5. Notes on Return Codes and Status Bytes

General return codes

There are return codes common to frame relay interface commands which require additional discussion. These return codes (hexadecimal values) are:

0x10 A modem failure occurred - DCD and/or CTS were found to be unexpectedly low.

The cause of this failure is returned at offset **0x00** of the data area and may be one of the following:

0x01 - DCD was found to be unexpectedly low.

0x02 - CTS was found to be unexpectedly low.

0x11 A change in the channel status occurred - the channel moved from **operative** to being **inoperative**.

0x12 A change in the channel status occurred - the channel moved from **inoperative** to being **operative**.

0x13 The Access Node has reported a change in the status of a DLCI or a number of DLCIs. The BUFFER_LENGTH is set to three times the number of DLCIs included in this report. The DATA area consists of a list of three byte structures as follows:

Offset 0x00 - 0x01: The DLCI (two byte hexadecimal value).

Offset 0x02: The DLCI status as follows:

Bit 0 - if set, the DLCI has been deleted.

Bit 1 - if set, then the DLCI is active and information transfer is possible.

Bit 3 - if set, then the DLCI is new.

0x14 The last Full Status Report received from the Access Node included a DLCI or a number of DLCIs which were not included in the DLCI list passed when performing the original SET_DLCI_CONFIGURATION command (DLCI zero).

The user should reconfigure the frame relay code to include these additional DLCIs or contact the network provider to establish the validity of these DLCIs.

The BUFFER_LENGTH is set to twice the number of unconfigured DLCIs included in this report. The DATA area consists of a list of these DLCIs, each being represented by a 2-byte, unsigned, hexadecimal value.

0x1F The frame relay command used is invalid.

6. Memory Interface and I/O Port Status Bytes

There are a number of bytes within the shared PC/SDLA memory area which may be useful for programmers using the shared memory interface.

These bytes are:

The **EXCEPTION_CONDITION_INTERFACE_BYTE** (offset 0xF003 on the adapter) - this byte indicates any exception conditions which may have occurred. If any of these bits are set and the adapter is not configured to generate an interrupt on a modem status change or a channel/DLCI status change, then the interface should be polled with a command such as a READ_DLC_STATUS to recover the exception return code of 0x10, 0x11, 0x12, 0x13 or 0x14.

The EXCEPTION_CONDITION_INTERFACE_BYTE is as follows:

If bit 0 is set, then the channel status has changed.

If bit 1 is set, then a change has occurred in the status of a DLCI or a number of DLCIs.

If bit 2 is set, then the last received Full Status Report included a DLCI or a number of DLCIs which were not configured at this CPE station.

If bit 6 is set, then DCD or CTS has been found to be low.

The **CURRENT_MODEM_STATUS_INTERFACE_BYTE** (offset 0xF004 on the adapter) - this byte indicates the current status of CTS and DCD and is updated on any change in the status of these two leads. The CURRENT_MODEM_STATUS_INTERFACE_BYTE is set as follows:

If bit 3 is set, then DCD is high.
If bit 5 is set, then CTS is high.

The I/O base port register may be read to reflect the status of the frame relay code as follows:

If bit 0 is set, then there is at least one received Information frame queued for reception by the application.

If bit 1 is set, then there is at least one transmit buffer available for use by the application.

7. Interrupt Usage

Once the interrupt vector and the PIC are initialized on the PC, the application informs the frame relay code of the required interrupt configuration by using the SET_INTERRUPT_TRIGGERS command.

Thereafter, in order to permit the adapter to generate interrupts to the PC, **the application must write 0x16 out to the adapter I/O port base address**.

When an interrupt occurs on the PC, the interrupt handler should examine
the INTERRUPT_INTERFACE_BYTE at offset 0xF010 on the adapter. This byte is as set as follows:

If bit 0 is set, then the interrupt has been triggered due to an Information frame being available for reception by the application.

If bit 1 is set, then the interrupt has been triggered as a frame may now be passed by the application to the adapter for transmission.
See Section 9 under Transmitting Information Frames for further details.

If bit 2 is set, then an interrupt has been triggered due to a change in the state of CTS or DCD. The details of this modem status change may be established by using a READ_DLC_STATUS command to illicit a return code of 0x10, and then examining the byte at offset 0x00 in the structure data area.

If bit 3 is set, then an interrupt has been triggered due to the completion of the last interface command, i.e. the 'opp_flag' has been reset.

If bit 4 is set, then the interrupt has been triggered due to a change in status of the channel or a DLCI. The details of this status change may be established by using a READ_DLC_STATUS command to illicit a return code of 0x11, 0x12, 0x13 or 0x14, and then appropriately processing this return code.

To reset the interrupt and to permit the next interrupt to be triggered, the interrupt handler should:

reset the PIC on the PC

reset the INTERRUPT_INTERFACE_BYTE byte (i.e., set this byte to 0x00)

Once the interrupts have been enabled by using the SET_INTERRUPT_TRIGGERS command, the interrupts may be temporarily disabled by using the INTERRUPT_PERMISSION_BYTE at offset 0xF011 on the adapter. The interrupt bit map for the INTERRUPT_PERMISSION_BYTE is the same as that used for the SET_INTERRUPT_TRIGGERS command, i.e,

bit 0 is used for frame reception

bit 1 is used for frame transmission

bit 2 is used for modem transitions

bit 3 is used for command completions

bit 4 is used for channel/DLCI status changes

If the bit is reset by the application, then an interrupt of that type will **not** occur until the bit is set again.

8. Transmitting and receiving information frames

Accessing memory on the S508 adapter

The adapter presents an 8K memory window to the host PC and this window may be moved by the application in order to be able to access various areas of adapter memory.

The 8K window is adjusted by writing a specified value out to the port address

adapter base I/O port + 2

as follows:

Value to be written to I/O base port + 2	Adapter memory accessed
0x00	0x00000 - 0x01FFF
0x01	0x02000 - 0x03FFF
0x02	0x04000 - 0x05FFF
0x03	0x06000 - 0x07FFF
0x04	0x08000 - 0x09FFF
0x05	0x0A000 - 0x0BFFF
0x06	0x0C000 - 0x0DFFF
0x07	0x0E000 - 0x0FFFF
0x08	0x10000 - 0x11FFF
0x09	0x12000 - 0x13FFF
0x0A	0x14000 - 0x15FFF
0x0B	0x16000 - 0x17FFF

0x0C	0x18000 - 0x19FFF
0x0D	0x1A000 - 0x1BFFF
0x0E	0x1C000 - 0x1DFFF
0x0F	0x1E000 - 0x1FFFF
0x10	0x20000 - 0x21FFF
0x11	0x22000 - 0x23FFF
0x12	0x24000 - 0x25FFF
0x13	0x26000 - 0x27FFF
0x14	0x28000 - 0x29FFF
0x15	0x2A000 - 0x2BFFF
0x16	0x2C000 - 0x2DFFF
0x17	0x2E000 - 0x2FFFF
0x18	0x30000 - 0x31FFF
0x19	0x32000 - 0x33FFF
0x1A	0x34000 - 0x35FFF
0x1B	0x36000 - 0x37FFF
0x1C	0x38000 - 0x39FFF
0x1D	0x3A000 - 0x3BFFF
0x1E	0x3C000 - 0x3DFFF
0x1F	0x3E000 - 0x3FFFF

An easy way of adjusting this 8K memory window would be through a procedure listed below:

```
void set_adapter_8K_memory_window(unsigned long adapter_memory_offset)
{
        outp(IO_port_base_address + 2, (adapter_memory_offset >> 13) & 0xFF);
}
```

Accessing the transmit and receive buffering information

Access to Information frame transmit and receive buffers is gained through a list of Transmit and Receive Status Elements.

The use of these Status Elements is explained the sections “Receiving Information Frames” and “Transmitting Information Frames”.

The location of these Status Elements and various buffer configuration parameters must be read after the initial SET_DLCI_CONFIGURATION has been performed. The buffer configuration information is as follows:

Parameter Definition	**Lengt h**	**Physical Offset On Adapter**
Number of Transmit Status Elements	2	0xF100
Base address of the Transmit Status Element list	4	0xF102
Number of Receive Status Elements	2	0xF120
Base address of the Receive Status Element list	4	0xF122
The address of the next Receive Status Element to be used	4	0xF126
Base address of the actual receive buffer area	4	0xF12A
End address of the actual receive buffer area	4	0xF130

Receiving information frames

General

The Receive Status Element is a 16 byte array configured as follows:

Parameter Definition	Lengt h	Offset Within Receive Status Element
Status flag	1	0x00
Length of I-frame data field	2	0x01
DLCI on which the I-frame was received	2	0x03
FECN/BECN/DE/CR bits in the received I-frame (represented by bits 3, 2, 1 and 0 respectively)	1	0x05
Reserved	6	0x06
Pointer to the actual received I-frame data area	4	0x0C

These Receive Status Element are used sequentially as Information frames are received by the adapter. This sequence is followed by the application, as the base address of these Receive Status Elements and the number of such elements may be established as described in “ACCESSING THE TRANSMIT AND RECEIVE BUFFERING INFORMATION”

Non-interrupt driven receive

1) The Status Flag of current Receive Status Element to be used is examined. When this flag is set to 0x01 by the adapter, then there is an I-frame available for the application.

2) The application reads the length of the data field, the DLCI, the FECN, BECN, CR and DE bits and the pointer to the actual data from the Receive Status Element.

3) The received I-frame data is read from the adapter as follows:

a) The 8K memory window is set to the physical memory area so as to access the base of the received data field.

b) The data is read until either:

i) all the data has been read

ii) the upper boundary of the current 8K memory segment has been reached

iii) the end address of the receive buffer has been reached

c) If not all the data has been read, then the 8K memory segment access is adjusted and step b) is repeated.

It is important to note that the receive buffer is a rotational buffer and so the base and end addresses of this buffer are made available to the application as described in the Section “Accessing the Transmit and Receive Buffering Information”.

4) The 8K memory window is set so as to access the current Receive Status Element and the Status Flag is now set to 0x00 so as to indicate to the adapter that the frame has been received by the application.

5) The application sets up to examine the next sequential Receive Status Element.

Interrupt driven receive

1) The SET_INTERRUPT_TRIGGERS command is used to enable receive interrupts.

2) Once a receive interrupt occurs, the logic for handling the incoming frame and the Receive Status Elements is the same as for the non-interrupt driven code described above.

Transmitting information frames

General

The Transmit Status Element is a 16 byte array configured as follows:

Parameter Definition	Length	Offset Within Transmit Status Element
Status flag	1	0x00
Reserved	4	0x01
Length of I-frame data field	2	0x05
DLCI on which the I-frame is to be transmitted	2	0x07
FECN/BECN/DE/CR bits in the outgoing I-frame (represented by bits 3, 2, 1 and 0 respectively)	1	0x09
Reserved	2	0x0A
Pointer to the actual I-frame data to be transmitted	4	0x0C

Non-interrupt driven transmissions

1) The I/O base port register is read until bit 1 is set, implying that there is at least one transmit buffer available for use by the application.

2) An INFORMATION_WRITE command is performed. If the return code from this command is zero (successful command), a pointer to the Transmit Status Element to be used is passed to the application in the first four bytes of the mailbox data area.

3) The pointer to the I-frame data area is read from this Transmit Status Element and the data to be transmitted is written to the adapter as follows:

> a) The 8K memory window is set to the physical memory area so as to access the base address of the transmit data area.
>
> b) The transmit data is written to the adapter until either:
>
> > i) all the data has been written
> >
> > ii) the upper boundary of the current 8K memory segment has been reached
>
> c) If not all the data has been written, then the 8K memory segment access is adjusted and step b) is repeated.

It is important to note that the transmit buffer is not a rotational buffer and so no buffer wrapping occurs as per the receive buffer.

Interrupt driven transmissions

There are two distinct modes of transmit interrupt usage:

1) DLCI-specific transmit interrupts - the adapter is set to issue a transmit interrupt when an Information frame of specified length may be transmitted on a specified DLCI. The interrupt will only be issued when the adapter has checked that the frame to be passed to the adapter will be transmitted, i.e. the channel and selected DLCI are active and any CIR restrictions have been satisfied.

2) Non DLCI-specific transmit interrupts - the adapter is set to issue a transmit interrupt when at least one buffer is available on the adapter for use by the application.

Transmit interrupts are used as follows:

1) The SET_INTERRUPT_TRIGGERS command is used to enable transmit interrupts and to define the type of transmit interrupt to be generated (DLCI or non-DLCI specific interrupts).

2) Once a transmit interrupt occurs, the application reads the four byte pointer to the Transmit Status Element to be used at offset 0xF012 on the adapter.

3) If DLCI-specific interrupts are being used, then the length of the data to be transmitted is set in the Transmit Status Element.

If non DLCI-specific interrupts are being used, then the length of the data to be transmitted, the DLCI to be used, as well as the FECN, BECN, CR and DE bits must be set in the Transmit Status Element.

3) The data to be transmitted is copied to the adapter as per non-interrupt driven I-frame transmissions.

4) The application sets bit 7 of the status flag in this Transmit Status Element (status flag set to 0x80) to indicate to the adapter that the frame may now be transmitted.

Note that in addition to setting the high bit of this status flag, there are additional bit settings which may be of use:

a) If the application does not currently have a frame to transmit but wishes to keep transmit interrupts enabled, then the status flag should be set to 0xA0.

b) If the application does not currently have a frame to transmit and wishes to disable transmit interrupts, then the status flag should be set to 0x90. Transmit interrupts may later be re-enabled by setting the appropriate bit in the INTERRUPT_PERMISSION_BYTE.

c) (Applicable to non DLCI specific interrupts only)

If the application has queued a frame for transmission in response to this interrupt and the current transmit throughput for this DLCI exceeds the defined CIR, then this frame will be discarded. However, if the user sets the status flag to 0xC0, then the frame will not be discarded, but will be queued for transmission once the CIR permits the release of this frame.

9. Using Frame Relay

Back-to-Back CPE/Switch Usage

It is useful to have two Sangoma Frame Relay boards in a back-to-back configuration, with one acting as the CPE and the other acting as the Access Node (switch). In order to achieve this setup, do the following:

The two boards should be connected by a Sangoma DB25 back-to-back cable.

Run FLOAD on both boards to download the Frame Relay code and then run FR_TEST above both stations.

Run the SET_DLCI_CONFIGURATION command on both machines and select DLCI 0. Set the station configuration on one machine as CPE, the other as NODE. Set both sides to provide internal clocking. The number of DLCIs configured should be set at both stations and DLCIs should be added to complete the configured DLCI list. Press 'S' to save the configuration.

Execute ENABLE_COMMUNICATIONS on both machines.

On the NODE machine, perform the ADD_DLCIs and then ACTIVATE_DLCs commands.

After a while, the CPE machine will show that a status change has occurred for the DLCIs. Note that the CPE polls the node at a period of T391 seconds and the PVC status is only reported every N391 polls. This means that if T391 is 10 seconds and N391 is 6, then it will take 60 seconds before the status of the configured DLCIs is reported.

The READ_DLC_STATISTICS command (DLCI zero) is useful for monitoring the In-channel polling as the counts of the various signalling frames are displayed.

The READ_DLC_STATUS and LIST_ACTIVE_DLCIs commands may be used to check the DLCI status. Any DLCIs which are listed as 'active' may be used for information transfer.

10. Managing Information Rates

The Sangoma Frame Relay implementation is designed to allow user control of the information rates. You can limit the information rate on any or all logical channels to the CIR, while maintaining optimum use of the line.

Where the card is being used on a network configured for all zero CIR, or one where the CIRs are at the line speed, there is no point in limiting the CIRs. The Backward and Forward CIR, Bc and Be can be set at the maximum, or preferably, CIR checking may disabled when configuring the code. The card will then act as a "dumb" router, sending and receiving data without trying to control the rate at which data is sent on any DLCI.

On networks where the CIR for each DLCI is a finite value less than line speed, set the Backward and Forward CIR, Bc and Be by using the SET_DLCI_CONFIGURATION call. The card will then only transmit on each DLCI at a rate such that the network will never set the DE bit. It will select frames from the transmit buffer so that as far as possible, line utilization is maximized while still staying below individual Forward CIRs. When in the CPE mode, the Backward CIR, Bc and Be have no effect on operation, they are simply used to keep statistics of incoming frame CIRs.

The data obtained from the READ_DLC_STATISTICS call can be used to monitor system performance and "tune" the network to reduce subscribed facilities to the minimum.

Buffering

The SDLA card has input and output buffering of frames. The smaller the maximum frame size set in the SET_DLC_CONFIGURATION call, the more efficiently the buffering is used.

The buffering can only deal with short term bursts. Obviously, the system can only work if the long term total data rate is below the line speed, and if the long term data rate on any DLCI is less than the CIR.

When the system is congested, and board buffers are full, data will not be accepted by the board. It is up to the application to either retry the INFORMATION_WRITE command or to discard the data and recover at the protocol level.

If there is consistent congestion on any DLCI, the Forward CIR should be raised. Note that you can always configure the board to have a higher CIR than the network, but then you are in danger of having the network discard frames.

11. MS-DOS Software modules

General

MS-DOS software support for Frame Relay is provided for the S508 only, and consists of the following modules:

S508LOAD.EXE is the Frame Relay microcode onto the S508 adapter, configures it, runs through a self test and starts the program.

FRM_REL.508 is the run time on board Frame Relay module loaded by S508LOAD.EXE.

FR_TEST.EXE is a high-level interface to the Frame Relay code which allows the user to perform individual interface commands.

S508LOAD.EXE

S508LOAD.EXE is a PC-DOS program and has the following command line syntax:

```
S508LOAD -c<FRAME  RELAY CODE MODULE> -p<I/O PORT>
-m<MEMORY>
```

where:

FRAME RELAY
CODE
MODULE is the run time Frame Relay module. This argument must be included, and the module **FRM_REL.508** is the one generally listed.

I/O PORT is the I/O port base address of the SDLA card. Valid values are 0x250, 0x270, 0x280, 0x300, 0x350, 0x360, 0x380 and 0x390. If this argument is not included, the I/O port defaults to 0x360. Note that the selection of this parameter must match the jumper settings on the adapter.

MEMORY is the location of the start of the 8k shared memory window of the SDLA card. Valid values are Ax, Cx, Dx, Ex, where x = 0, 2, 4, 6, 8, A, C. For example CA means that the shared memory area is from CA000 to CBFFF. If this argument is not included, the memory window defaults to C8.

For example:

```
S508LOAD -cFRM_REL.508 -mC8 -p300
```

S508LOAD will perform a system test, read the CODE file and load and configure the adapter. If S508LOAD does not execute successfully, an error message will be displayed and an exit code will be returned. A description of these messages and corresponding exit codes is given in The section, "Error Messages".

Optionally, a ‘-h’ option may be used with S508LOAD.EXE. This will initiate a full hardware check of the adapter and the PC environment, and report on the I/O port address for which the adapter is configured, the available 8K memory addresses on the PC and the IRQs available for use.

FRM_REL.508

This is the Frame Relay support code which is loaded onto the card, executes and establishes the communications. FRM_REL.508 is **NOT** a PC-DOS program and is not executable under DOS.

FR_TEST.EXE

FR_TEST is a high-level interface to the Frame Relay code which allows the user to perform individual interface commands.

FR_TEST.EXE is loaded with the following command line:

```
FR_TEST -m<MEMORY>
```

where:

MEMORY is the location of the start of the 8k shared memory window of the SDLA card. Valid values are Ax, Cx, Dx, Ex, where x = 0, 2, 4, 6, 8, A, C. For example CA means that the shared memory area is from CA000 to CBFFF. Note that this argument must be the same as that used when executing S508LOAD.EXE.

If FR_TEST does not execute successfully, an error message will be displayed and an exit code will be returned. A description of the error messages and corresponding exit codes is given in the section, "Error Messages".

12. MS-DOS Error Messages

FLOAD.EXE

If FLOAD does not execute successfully, an error message will be displayed and an exit code will be returned. The error messages and corresponding exit codes (DOS ERRORLEVEL) are as follows:

"**A command line error was found when executing FLOAD**" - an invalid command line argument was used (exit code of 0x01).

The file FILENAME was not found" - a filename listed in the command line arguments was not found in the defined directory (exit code of 0x02).

"**The I/O port address PORT ADDRESS read from the command line is invalid**" - an invalid I/O port parameter was specified in the command line arguments (exit code of 0x03).

"**The memory segment address MEMORY SEGMENT read from the command line is invalid**" - an invalid memory segment address was specified in the command line arguments (exit code of 0x04).

"**The memory offset address MEMORY OFFSET read from the command line is invalid**" - an invalid memory offset address was specified in the command line arguments (exit code of 0x05).

"**The code running on the adapter is not the same as the original downloaded code**" - there is a memory or I/O port address conflict in your PC. Change the I/O port address and/or the memory segment and memory window parameters (exit code of 0x06).

"The downloaded code is not running on the adapter. Select a different memory address" - select a different memory address and run DLOAD again (exit code of 0x07).

"**The downloaded code is not running on the adapter**" - the card CPU has halted. Contact your Sangoma representative (exit code of 0x07).

"**The S508 adapter is configured with the incorrect serial communications chip**" - contact your Sangoma representative (exit code of 0x08).

"**The S508 adapter has not been located at the defined I/O port address**" - the loader could not locate the S508 hardware at the specified I/O port address. Check the jumper settings (exit code of 0x09).

FR_TEST.EXE

The error messages and corresponding exit codes for FR_TEST are as follows:

"**A COMMAND LINE ERROR WAS FOUND WHEN LOADING FR_TEST**" - an invalid command line argument was used (exit code of 0x01).

"**THE FRIP TSR HAS NOT BEEN FOUND AT INTERRUPT VECTOR <VECTOR>**" - (exit code of 0x02).

"**AN SDLA TIMEOUT OCCURRED (THE FRAME RELAY CODE IS NOT RUNNING)**" - contact your Sangoma representative (exit code of 0x0A).