

Operations Manual

December 2001

# **BSC 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 S-Series cards are general co-processor communication adapters capable of supporting any communication protocol autonomously, and providing information transfer into PC workspace. RS232, V.35 and X.21 interfaces are supported.

This manual describes the programming interface to the Binary Synchronous Communications (BSC) protocol code on the Sangoma card set. Code support is provided for both point-to-point and multipoint (3270) operation. Both primary and secondary modes of operation are supported for each protocol.

## Supported BSC Features

The BSC code handles all aspects of the protocol, including appropriate formatting of text block frames. The following summarizes the major aspects of BSC that are supported.

- ▼ EBCDIC and ASCII transmission codes.
- ▼ Framing format flexibility with a configurable number of preceding SYN and trailing PAD characters. Receiver can synchronize on one SYN character for locating start of frame.
- ▼ Constant RTS or switched CTS/RTS operation.
- ▼ User-provided heading blocks.
- ▼ Transparent text mode.
- ▼ Limited conversational mode.
- ▼ Switched-network (dial-up) operation (point-to-point).
- ▼ General or specific polling modes (multipoint).
- ▼ Large number of tunable retry counters and timers to adjust the behavior of the station.

## Conventions Used in this Manual

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

### 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 the table below. 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.

#### JP1 - I/O Address Selection

JP1-1	JP1-2	JP1-3	I/O Address
CLOSED	CLOSED	CLOSED	250-253 (Hex)
OPEN	CLOSED	CLOSED	270-273 (Hex)
CLOSED	OPEN	CLOSED	280-283 (Hex)
OPEN	OPEN	CLOSED	300-303 (Hex)
CLOSED	CLOSED	OPEN	350-353 (Hex)
OPEN	CLOSED	OPEN	360-363 (Hex) *
CLOSED	OPEN	OPEN	380-383 (Hex)
OPEN	OPEN	OPEN	390-393 (Hex)

\* Default

### S503 ISA Card

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.

#### I/O port address:

This is set by Jumper **JP3**.

Pins 5-6	Pins 3-4	Pins 1-2	I/O Address Selection
OPEN	CLOSED	CLOSED	250-252 (Hex)
CLOSED	CLOSED	CLOSED	254-256 (Hex)
OPEN	CLOSED	OPEN	300-302 (Hex)
CLOSED	CLOSED	OPEN	304-306 (Hex)
OPEN	OPEN	CLOSED	350-352 (Hex)
CLOSED	OPEN	CLOSED	354-356 (Hex)
OPEN	OPEN	OPEN	360-362 (Hex)*
CLOSED	OPEN	OPEN	364-366 (Hex)

\* 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
CLOSED	OPEN	OPEN	OPEN	OPEN	IRQ 2
OPEN	CLOSED	OPEN	OPEN	OPEN	IRQ 3
OPEN	OPEN	CLOSED	OPEN	OPEN	IRQ 4
OPEN	OPEN	OPEN	CLOSED	OPEN	IRQ 5
OPEN	OPEN	OPEN	OPEN	CLOSED	IRQ 7 <sup>Ⓜ</sup>

<sup>Ⓜ</sup> Factory default.

### Interface Level Selection

This is set by Jumper **JP3**.

Pins 9-10	Interface Level
CLOSED	RS-232
OPEN	V.35

## 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		
18			TXA	
19			TXB	
20			RXA	
21			RXB	

PIN #	PA:RS232	PA:V.35	PB:RS232	PB:V.35
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**

<b>Pin #</b>	<b>Function</b>
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 hardware level shared memory interface

Sangoma cards are operated by reading and writing bytes and structures from and to positions in the shared memory window. Under the different operating systems, different mechanisms are used to read/write to the shared memory area, but otherwise, all implementations are identical.

The size of the shared memory window is 8192 bytes, and it may be set to access any particular 8k segment on the board. The loader sets the window to the appropriate one for interfacing with the user, and all memory addresses in this manual are relative to the start of this window. It will not be necessary to change the shared memory window once the code is loaded.

Reading and writing blocks of memory is required for sending and receiving data or any other interaction with the card. It is also necessary to issue commands to the board, and wait for a response. A special control block structure exists for issuing commands and for returning error codes. The structure is read from and written to the shared memory window directly by the user application. The operation is started once all required structure elements are filled and a special OP\_FLAG byte is set to 0x01. The operation is completed by the board when this byte is reset to 0x00.

#### Shared Memory Control Block Structure

The control block structure is used to interface between the user application and the BSC code on the board, and it starts at the relative address 0x1000.

Parameter	Off-set	Lgth	Remarks
OP_FLAG	00H	1	A flag set by the user (to 0x01) to inform the SDLA processor that a COMMAND is pending. This flag is in turn reset (to 0x00) 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.

Parameter	Off-set	Lgth	Remarks
MISC_TX_RX_BITS	05H	1	Options set when sending and receiving text data.
HEADING_LENGTH	06H	2	Length of heading data included in text data.
NOTIFY	08H	2	Bits set to indicate events that took place since the last command call. Reset to 0 by the SDLA after each call.
Multipoint Fields	0AH	6	Fields reserved for use by Multipoint BSC code.
DATA	10H	4000	This is the transfer area for passing data to and from the application level.

## Multipoint-Specific Fields in Control Block Structure

Parameter	Off-set	Lgt h	Remarks
STATION	0AH	1	Station number associated with this command.
POLL_ ADDRESS	0BH	1	Poll address associated with this command.
SELECT_ ADDRESS	0CH	1	Select address associated with this command.
DEVICE_ ADDRESS	0DH	1	Device address associated with this command.
NOTIFY_ EXTENDED	0EH	1	Extended information pertaining to events set in NOTIFY field. Reset to 0xFF by the SDLA after each call.
RESERVED	0FH	1	Reserved.

## 4. COMMAND Codes

The valid commands are:

### Common Operational Commands

0x00	BSC_READ
0x01	BSC_WRITE
0x02	OPEN_LINK
0x03	CLOSE_LINK
0x06	LINK_STATUS
0x0D	FLUSH_BSC_TEXT_BUFFERS
0x0E	SET_CONFIGURATION
0x0F	READ_CONFIGURATION
0x10	SET_MODEM_STATUS
0x11	READ_MODEM_STATUS
0x12	READ_CODE_VERSION
0x30	READ_STATE_DIAGNOSTICS

### Common Statistics Commands

0x07	READ_OPERATIONAL_STATISTICS
0x08	FLUSH_OPERATIONAL_STATISTICS
0x09	READ_COMMS_ERROR_STATISTICS
0x0A	FLUSH_COMMS_ERROR_STATISTICS
0x0B	READ_BSC_ERROR_STATISTICS
0x0C	FLUSH_BSC_ERROR_STATISTICS

### Additional Point-to-Point Commands

0x04	CAM_WRITE
0x05	CAM_READ

## **Additional Multipoint Commands**

0x20	ADD_STATION
0x21	DELETE_STATION
0x22	DELETE_ALL_STATIONS
0x23	LIST_STATIONS
0x24	SET_GENERAL_OR_SPECIFIC_POLL
0x25	SET_STATION_STATUS

## **BSC\_READ (0x00)**

This command reads a text frame from the receive buffer queue.

### **Control Block values to be set on entry:**

**BUFFER\_**

**LENGTH:** Set to 0x00.

**STATION:** **Multipoint Only**

Station number associated with read request.

### **Control Block values set on return:**

**RETURN\_**

**CODE:** 0x00 The action was performed successfully.

0x01 No data is available to read.

0x04 The link is currently closed.

0x05 **Point-to-Point Only**

Link is currently in circuit assurance mode.

0x07 **Multipoint Only**

Specified station is not configured.

### **Valid if RETURN\_CODE is 0x00:**

**BUFFER\_**

**LENGTH:** Set to total length of heading and text data.

HEADING\_

LENGTH: Set to length of heading data.

MISC\_TX\_

RX\_BITS: **Bit 0:** Set if text data is transparent.

**Bits 2,1:** 00 - ETX ended this block.

01 - ETB ended this block.

10 - ITB ended this block.

**Bit 4:** Set in **Multipoint Primary** if the first two bytes of text were assumed to be poll and device address, and were removed.

**Bits 7,6,5,3:** Reserved or not relevant.

DATA: The decoded heading and text data, without framing characters such as SYN, PAD, BCC, or extra DLEs. The first HEADING\_LENGTH bytes were located in the heading section of the frame, and the following (BUFFER\_LENGTH-HEADING\_LENGTH) text bytes were located after STX.

On **Multipoint Primary**, the first two bytes are assumed to be poll address and device address, which are **removed** before passing the frame to the application using this command.

## BSC\_WRITE (0x01)

This command writes a text frame to the transmit buffer queue.

Note that if ITB or ETB blocks are sent, and the configured TTD timer expires, TTD will be sent for the configured number of times until either an ETX block is sent, or the maximum number of TTD frames is reached in which case an early EOT is sent.

Although ITB is supported, its use may have unpredictable results due to the limited buffering capacity, in conjunction with the inability to NAK it when buffers are full until a subsequent ETB or ETX is received. ITB is therefore not recommended unless absolutely necessary.

### Control Block values to be set on entry:

#### BUFFER\_

LENGTH: Set to total length of heading and text data.

#### HEADING\_

LENGTH: Set to length of heading data.

#### MISC\_TX\_

RX\_BITS: **Bit 0:** Set if text data is transparent.

**Bits 2,1:** 00 - End this block with ETX.

01 - End this block with ETB.

10 - End this block with ITB.

#### **Bit 4:**

Set in **Multipoint Secondary** if the text frame is to be preformatted to include the poll and device addresses in the text. If not set, the frame will be reformatted to include them if necessary during data transfer. If set, but the addresses must not be included at the time of data transfer, the frame will be reformatted to remove them. It is generally recommended that Bit 4 be set for the first logical block of the message, and reset for the following blocks up to and including the ETX-terminated block. Assuming the entire message is transmitted in one poll sequence, this method will maximize average throughput.

STATION: **Multipoint Only**

Station number associated with write request.

**DATA:** The raw heading and text data, without framing characters such as SYN, PAD, BCC, or extra DLEs. The first `HEADING_LENGTH` bytes will be located in the heading section of the frame, and the following (`BUFFER_LENGTH-HEADING_LENGTH`) text bytes will be located after STX.

On **Multipoint Secondary**, the poll and device address associated with `STATION` will be inserted in front of the text data automatically when necessary. The user must not pass these bytes as part of the data.

### **Control Block values set on return:**

`RETURN_`

- `CODE:`
- `0x00` The frame has been queued for transmission.
  - `0x02` The transmit buffer is full. Increase transmit queue size or reduce the transmission rate.
  - `0x03` The frame is too long to be stored in a transmit buffer after formatting.
  - `0x04` The link is currently closed.
  - `0x05` **Point-to-Point Only**  
Link is currently in circuit assurance mode.
  - `0x06` **Multipoint Only**  
Specified station is not active.
  - `0x07` **Multipoint Only**  
Specified station is not configured.

## **OPEN\_LINK (0x02)**

This command raises DTR and enables the receiver. If operating in constant RTS mode, RTS is also raised. See **protocol\_options1** for **Point-to-Point** if changing DTR/RTS is not desirable.

If dial-up operation has been configured with **Point-to-Point**, the link will enter circuit assurance mode (CAM).

### **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 link has already been opened.

## **CLOSE\_LINK (0x03)**

This command drops DTR and RTS, disables the receiver, and flushes the transmit and receive buffers. See **protocol\_options1** for **Point-to-Point** if changing DTR/RTS is not desirable.

### **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.

0x03 The link has already been closed.

## **CAM\_WRITE (0x04)**

(Point-to-Point Only)

This command is used to send a circuit assurance message.

### **Control Block values to be set on entry:**

**BUFFER\_**

**LENGTH:** Set to length of ID sequence (2 to 15), or 0 if none.

**MISC\_TX\_**

**RX\_BITS:** Set to type of circuit assurance message to send:

0x01: I AM/WRU

0x02: (ID) ACK0

0x03: (ID) NAK

0x04: WACK

0x05: CONNECT

0x06: DISCONNECT

**DATA:** Identification sequence if applicable.

### **Control Block values set on return:**

**RETURN\_**

**CODE:** 0x00 The action was performed successfully.

0x01 Not in circuit assurance mode.

0x02 Circuit assurance message transmit buffer is full.

0x03 Dial-up operation has not been configured.

0x04 The link is currently closed.

0x06 Identification sequence is too long.

## **CAM\_READ (0x05)**

### **(Point-to-Point Only)**

This command is used to read a circuit assurance message. There is only one receive buffer for these messages. If it is full when another message is received, the newer one is stored and the older one is discarded.

#### **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 No circuit assurance messages have been received.  
0x03 Dial-up operation has not been configured.

#### **Valid if RETURN\_CODE is 0x00:**

**BUFFER\_**  
**LENGTH:** Set to length of identification sequence if included.

**MISC\_TX\_**  
**RX\_BITS:** Set to type of circuit assurance message received:  
0x01: I AM/WRU  
0x02: (ID) ACK0  
0x03: (ID) NAK  
0x04: WACK  
0x05: CONNECT  
0x06: DISCONNECT

**DATA:** Identification sequence if applicable.

## LINK\_STATUS (0x06)

Returns the state of the link and transmit and receive buffer usage.

### 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.

**BUFFER\_**

**LENGTH:** Set to 0x06.

**DATA:**

Offset 0x00

### Current state

(PP=Point-to-Point, MP=Multipoint Primary, MS=Multipoint Secondary)

<b>0x00</b>		Link not open
<b>0x01</b>		Open and idle
<b>0x02</b>	PP	Waiting for bid response
	MP	Waiting for select response
	MS	Waiting for EOT
<b>0x03</b>		Ready to transmit data
<b>0x04</b>		Ready to receive data
<b>0x05</b>		Waiting for tx data response
<b>0x06</b>		Waiting for final tx data response
<b>0x07</b>	PP	Waiting to idle after sent EOT
	MP/MS	Waiting for response to TTD
<b>0x08</b>		A bad conversational reply received
<b>0x09</b>		Ready to transmit conversational reply
<b>0x0A</b>		Waiting for response to conv-reply

<b>0x0B</b>		Waiting to retransmit NAK'd frame
<b>0x0C</b>	PP	Circuit assurance mode
<b>0x0D</b>	PP	Passive receive mode
<b>0x0E</b>		Reserved

These states are provided for reference when debugging the link, and may be used in conjunction with the results of READ\_STATE\_DIAGNOSTICS for reporting problems to Sangoma. For purposes of general use, only the values 0x00, 0x0C and any value between 0x01-0x0B, indicating normal operation, should be of concern. When idle over a sufficient period of time, the state must be either 0x00, 0x01, or 0x0C. Otherwise, it is trapped in a transitional state. If this happens, record this value and the results of READ\_STATE\_DIAGNOSTICS and send them to Sangoma.

Offset 0x01 Total number of receive buffers full.

Offset 0x02 **Point-to-Point**

Set to 0x01 if one or more frames are queued for transmission, or are pending acknowledgment; otherwise, 0x00.

**Multipoint**

Total number of transmit buffers full.

Offset 0x03 Set to 0x01 if at least one global transmit buffer is free; otherwise, 0x00.

Offset 0x04 Set to 0x01 if the receiver has halted; otherwise, 0x00. If the receiver remains halted, look at the statistics returned by READ\_BSC\_ERROR\_STATISTICS. A CLOSE\_LINK followed by OPEN\_LINK should resolve the short-term problem.

Offset 0x05 **Point-to-Point**

Reserved and set to 0x00.

**Multipoint Primary**

Set to 0x00 if in specific polling mode, or 0x01 if in general polling mode.

**Multipoint Secondary**

Always 0x00.

## READ\_OPERATIONAL\_STATISTICS (0x07)

Retrieve general statistics on the operation of this node.

### 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.

BUFFER\_

LENGTH: **Point-to-Point:** Set to 0x4E.

**Multipoint:** Set to 0x54.

DATA:

The following counters are 2-byte unsigned integers.

Offset	Counter
0x00	Received frames processed.
0x02	Message syncs parsed (SYN SYN or DLE SYN).
0x04	Received frames processed in circuit assurance mode ( <b>Point-to-Point</b> ).
0x06	Conversational replies received.
0x08	ETX frames transmitted as a conversational reply.
0x0A	ENQ transmitted.
0x0C	ACK0 transmitted.
0x0E	ACK1 transmitted.
0x10	WACK transmitted.
0x12	EOT transmitted.
0x14	ETX transmitted.
0x16	ETB transmitted.
0x18	ITB transmitted.

0x1A	RVI transmitted.
0x1C	TTD transmitted.
0x1E	NAK transmitted.
0x20	Disconnect (DLE EOT) transmitted ( <b>Point-to-Point</b> ).
0x22	Identification (ID ACK0) transmitted ( <b>Point-to-Point</b> ).
0x24	IAM/WRU (ID ENQ) transmitted ( <b>Point-to-Point</b> ).
0x26	Connect (EOT) transmitted ( <b>Point-to-Point</b> ).
0x28	Circuit assurance WACK transmitted ( <b>Point-to-Point</b> ).
0x2A	Circuit assurance NAK transmitted ( <b>Point-to-Point</b> ).
0x2C	ENQ received.
0x2E	ACK0 received.
0x30	ACK1 received.
0x32	WACK received.
0x34	EOT received.
0x36	ETX received.
0x38	ETB received.
0x3A	ITB received.
0x3C	RVI received.
0x3E	TTD received.
0x40	NAK received.
0x42	Disconnect (DLE EOT) received ( <b>Point-to-Point</b> ).
0x44	Identification (ID ACK0) received ( <b>Point-to-Point</b> ).
0x46	IAM/WRU (ID ENQ) received ( <b>Point-to-Point</b> ).
0x48	Connect (EOT) received ( <b>Point-to-Point</b> ).
0x4A	Circuit assurance WACK received ( <b>Point-to-Point</b> ).
0x4C	Circuit assurance NAK received ( <b>Point-to-Point</b> ).

**Multipoint Only:**

0x4E	Total number of polls (transmitted by <b>Primary</b> or received by <b>Secondary</b> ).
0x50	Total number of general polls (transmitted by <b>Primary</b> or received by <b>Secondary</b> ).
0x52	Total number of selects (transmitted by <b>Primary</b> or received by <b>Secondary</b> ).

## **FLUSH\_OPERATIONAL\_STATISTICS (0x08)**

The current values of the variables accessed by the **READ\_OPERATIONAL\_STATISTICS** 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 was performed successfully.

## READ\_COMMS\_ERROR\_STATISTICS (0x09)

Retrieve physical link error statistics for this node.

### 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.

BUFFER\_

LENGTH: Set to 0x08.

DATA:

The following counters are 1-byte unsigned integers.

Offset	Counter
0x00	Receiver overrun errors.
0x01	Transmit underrun errors.
0x02	CTS dropped before transmitting (unexpected).
0x03	CTS dropped while transmitting.
0x04	Transmit timeout while waiting for CTS.
0x05	CTS high before transmitting (switched mode).
0x06	Number of times DCD changed.
0x07	Number of times CTS changed.

## **FLUSH\_COMMS\_ERROR\_STATISTICS (0x0A)**

The current values of the variables accessed by the **READ\_COMMS\_ERROR\_STATISTICS** 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 was performed successfully.

## READ\_BSC\_ERROR\_STATISTICS (0x0B)

Retrieve protocol error statistics for this node.

### 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.

BUFFER\_  
LENGTH: **Point-to-Point:** Set to 0x20.  
**Multipoint:** Set to 0x2E.

DATA:

The following counters are 2-byte unsigned integers.

Offset	Counter
0x00	Unable to decode basic frame type.
0x02	Receiver discards, no buffers.
0x04	Discards recorded above (offset 0x02) which were conversational replies and were treated as ACKs.
0x06	Received circuit assurance message discarded, buffer full ( <b>Point-to-Point</b> ).
0x08	Received circuit assurance messages discarded, bad format ( <b>Point-to-Point</b> ).
0x0A	Receiver discards, too long.
0x0C	Receiver ITB discards, no buffers.
0x0E	Receiver text discards, bad BCC.
0x10	Receiver text discards, incorrect state.

0x12	Premature transmit aborts received (TTD).
0x14	Transmitter text discards, retries exceeded.
0x16	NAK received after transmission of ETB/ETX.
0x18	Response timeouts to text transmission.
0x1A	Number of times receiver was halted (interrupt buffer overflow or link was closed).
0x1C	Number of catastrophic receiver restarts (resynchronization of interrupt and application receiver handlers).
0x1E	Frame parser reached end of buffer prematurely (normally the result of a poorly-terminated or otherwise badly-formatted frame).

**Multipoint Only:**

0x20	Badly formatted poll/select received ( <b>Secondary</b> ).
0x22	Unknown poll/select address received ( <b>Secondary</b> ).
0x24	Unknown device polled ( <b>Secondary</b> ). No response on poll ( <b>Primary</b> ).
0x26	Unknown device selected ( <b>Secondary</b> ). No response on select ( <b>Primary</b> ).
0x28	Receiver text discards, embedded addresses don't match those set in transmitted poll sequence ( <b>Primary</b> ).
0x2A	Receiver text discards, unknown embedded device address during general poll phase ( <b>Primary</b> ).
0x2C	Transmitter text discards, selection retry limit exceeded on device ( <b>Primary</b> ).

## **FLUSH\_BSC\_ERROR\_STATISTICS (0x0C)**

The current values of the variables accessed by the **READ\_BSC\_ERROR\_STATISTICS** 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 was performed successfully.

## FLUSH\_BSC\_TEXT\_BUFFERS (0x0D)

Discards all text frames in the transmit or receive queues.

### Control Block values to be set on entry:

BUFFER\_

LENGTH: Set to 0x01.

STATION: **Multipoint Only**

Station number whose buffers will be flushed.

DATA:

Offset 0x00 **Flush control**

0x01 Transmit queue flushed only.

0x02 Receive queue flushed only.

0x03 Both queues are flushed.

### Control Block values set on return:

RETURN\_

CODE: 0x00 The action has been performed successfully.

0x07 **Multipoint Only**

Specified station has not been configured.

## SET\_CONFIGURATION (0x0E)

This command sets the configuration for all BSC options.

### Control Block values to be set on entry:

**BUFFER\_**

**LENGTH:**   **Point-to-Point:**       Set to 0x25.

**Multipoint:**         Set to 0x2f.

**DATA:**

Offset 0x00	<b>line_speed_number</b>	
	<b>number</b>	<b>approximate bit rate</b>
	0x00	External clocking
	0x01	1200
	0x02	2400
	0x03	4800
	0x04	9600
	0x05	19200
	0x06	38400
	0x07	45000
	0x08	56000
	0x09	64000
	0x0A	74000
	0x0B	112000
	0x0C	128000
	0x0D	156000

Offset 0x01-0x02    **max\_data\_frame\_size**

Maximum text frame size after expansion due to formatting and framing (including SYNs and PADs). The maximum permitted size is 4000. If a value greater than 4000 is specified, it will be silently set to 4000.

Offset 0x03           **secondary\_station**

Set to 0x01 for **Secondary** operation, or 0x00 for **Primary** operation. If **Point-to-**

**Point** is used, one peer must be **Primary** and the other **Secondary**.

Offset 0x04                    **num\_consec\_PAD\_eof**

Number of PADs (FF) to append to end of frame on transmission, and number expected at end of frame on reception for purposes of synchronization. The recommended value is 5. Valid range is 2 to 255.

Offset 0x05                    **num\_add\_lead\_SYN**

Number of additional leading SYN characters to prepend to transmitted frames. If set to zero, one SYN is sent. The receiver synchronizes on one SYN character, and any following SYNs are ignored. The recommended value is 1 or 2. Valid range is 0 to 255.

Offset 0x06                    **conversational\_mode**

If set to 0x01, limited conversational mode will be enabled. Set to 0x00 to disable it. When enabled, one ETX frame will be sent as a positive acknowledgement to a received ETX frame, if queued for transmission at the time. Conversational replies are always permitted on reception, but they will not be transmitted unless this mode is enabled.

Offset 0x07                    **pp\_dial\_up\_operation            (Point-to-Point)**

Set to 0x01 to enable circuit assurance mode over a switched point-to-point link, or 0x00 for leased-line operation. If enabled, this mode is entered immediately following an OPEN\_LINK command, and a Connect message must be sent by either peer to enter normal data transfer mode. This parameter is ignored in **Multipoint**.

Offset 0x08                    **switched\_CTS\_RTS**

Set to 0x01 to enable switched CTS/RTS operation, or 0x00 for constant RTS operation. In switched mode, RTS is raised and transmission begins when CTS is raised; RTS is dropped when transmission ends.

Offset 0x09                    **EBCDIC\_encoding**

Set to 0x01 for EBCDIC encoding, or 0x00 for ASCII encoding.

Offset 0x0A

**auto\_open**

If set to 0x01, the link will be opened when the configuration is set, avoiding the need to issue OPEN\_LINK. Set to 0x00 to have manual control.

Offset 0x0B

**misc\_bits**

Bit 0: **Point-to-Point**

Set to 1 to enable **Passive Receive Mode**. This mode is entered on LINK\_OPEN where it will receive only. Text frames received may be retrieved using BSC\_READ. All control frames are ignored, but the appropriate statistics will still be incremented. Do not attempt to use BSC\_WRITE in this mode.

Bit 1: **Point-to-Point**

Set to 1 to enable **Streaming Receive Mode**. Bit 0 must be set also, to keep the state machine in passive mode. This is a feature of **Passive Receive Mode** where all received frames are passed to the user on a BSC\_READ command. The data consists of everything past the first SYN character up to the end of frame marked by PADs. See the section in this manual describing this mode for more information.

Bits 2-7 are reserved for future use.

Offset 0x0C

**protocol\_options1**

Bit 0: **Multipoint Secondary**

Set to 1 to allow the reception and processing of poll/select sequences without being prefixed by an EOT as part of the same logical frame. If the secondary does not respond to polls, and the EOT received statistic goes up, and the 'Unable to decode basic frame type' statistic also goes up, then set this bit.

Bit 0: **Point-to-Point**

Set to 1 to prevent OPEN\_LINK and CLOSE\_LINK from changing DTR and RTS.

Bit 1: **Point-to-Point**

Set to 1 to indicate changes in DCD/CTS in the NOTIFY mailbox field. Note that changes may not be indicated during a CLOSE\_LINK command, if any. To be sure, use READ\_MODEM\_STATUS after CLOSE\_LINK if

necessary.

Bits 1-7 are reserved for future use.

Offset 0x0D            **protocol\_options2**

Bits 0-7 are reserved for future use.

Offset 0x0E-0x0F    **reserved**

Offset 0x10            **max\_retransmissions**

Number of times a text frame will be retransmitted when a NAK or no response is received before discarding it. Valid range is 0 to 255.

Offset 0x11-0x12    **fast\_poll\_retries**

Number of consecutive fast polls that are sent before switching to slow polling if no response is received. This is used by **Point-to-Point** for line bidding, and **Multipoint Primary** for device polling. This parameter is ignored by **Multipoint Secondary**. Valid range is 0 to 65535.

Offset 0x13-0x14    **TTD\_retries**

Number of TTDs sent before giving up the line while waiting for the user to submit an ETX frame to the transmit queue. If set to zero, an EOT will be sent immediately following the last frame sent if no more are queued at the time - be sure to set **TTD\_timer** to zero in this case. Valid range is 0 to 65535.

Offset 0x15-0x16    **restart\_timer**

Time between successive line bids (**Point-to-Point**) or polls (**Multipoint Primary**) in 100ths of a second. Valid range is 0 to 13107.

Offset 0x17-0x18    **pp\_slow\_restart\_timer**    (**Point-to-Point**)

Time between successive line bids after **fast\_poll\_retries** has expired in 100ths of a second. Valid range is 0 to 13107. This parameter is ignored in **Multipoint**.

Offset 0x19-0x1A    **TTD\_timer**

Time duration in 100ths of a second to wait before transmitting another TTD after the last NAK response, and the time to wait initially for an ETX frame from the user before starting to send TTDs. Valid range is 0 to 13107. Set this value to zero if **TTD\_retries** is set to zero.

Offset 0x1B-0x1C     **pp\_delay\_between\_EOT\_ENQ**  
                          **(Point-to-Point)**

Time to wait in 100ths of a second between EOT and the next line bid. The peer may bid for the line during this time. Valid range is 0 to 13107.

Offset 0x1D-0x1E     **response\_timer**

Time to wait in 100ths of a second for a response to text transmission, or to a select in the case of **Multipoint Primary**.

Offset 0x1F-0x20     **rx\_data\_timer**

Time to wait in 100ths of a second for text data or an EOT to arrive before returning to an idle state. This is also a poll timeout delay in the case of **Multipoint Primary**. Valid range is 0 to 13107.

Offset 0x21-0x22     **NAK\_retrans\_delay\_timer**

Time to wait in 100ths of a second to retransmit a NAK'd text frame. This timer starts on reception of the NAK. Valid range is 0 to 13107.

Offset 0x23-0x24     **wait\_CTS\_timer**

Time to wait in 100ths of a second for CTS to be raised in response to RTS in switched CTS/RTS mode. If this timer expires, the transmission will be aborted. The state machine will assume it was transmitted, and retransmission may occur if the relevant response timer expires. Valid range is 0 to 13107.

### **Multipoint Only:**

Offset 0x25           **mp\_max\_consec\_ETX**

Maximum number of consecutive ETX frames that may be sent per poll or select, per device. Valid range is 1 to 255.

Offset 0x26           **mp\_general\_poll\_address**

The device address used when general polling mode is enabled. It is typically 0x7F in EBCDIC or 0x22 in ASCII.

Offset 0x27-0x28     **sec\_poll\_timeout**     **(Secondary)**

Time to wait in 100ths of a second from last poll before declaring the station inactive. Valid range is 0 to 13107. This parameter is ignored by **Primary**.

Offset 0x29           **pri\_poll\_skips\_inactive**     **(Primary)**

Number of consecutive polls to skip for an inactive station. Valid range is 0 to 255. This parameter is ignored by **Secondary**.

Offset 0x2A            **sec\_additional\_stn\_send\_gpoll**  
                          **(Secondary)**

Number of additional consecutive devices that may send text blocks on a general poll. A value of zero allows one device to respond per general poll. Valid range is 0 to 255. This parameter is ignored by **Primary**.

Offset 0x2B            **pri\_select\_retries**    **(Primary)**

Number of selects to retry after no response or a NAK before giving up and discarding the frames in the transmit queue. Valid range is 0 to 255. This parameter is ignored by **Secondary**.

Offset 0x2C            **mp\_multipoint\_options**

Bits 0-7 are reserved for future use.

Offset 0x2D-0x2E    **reserved**

### **Control Block values set on return:**

RETURN\_

CODE:    0x00    The action was performed successfully.

          0x01    The link is currently open, and must be closed before the configuration can be set.

## READ\_CONFIGURATION (0x0F)

Retrieves the current configuration.

### 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.

BUFFER\_

LENGTH: **Point-to-Point:** Set to 0x27.

**Multipoint:** Set to 0x31.

DATA:

See SET\_CONFIGURATION for returned data. The information returned is exactly as set, except for a possible adjustment of **max\_data\_frame\_size**. In addition to this information, the number of buffers allocated is returned:

### Point-to-Point:

Offset 0x25 Total number of transmit buffers allocated.

Offset 0x26 Total number of receive buffers allocated.

**Multipoint:**

Offset 0x2F	Total number of transmit buffers allocated, to be shared by all configured stations.
Offset 0x30	Total number of receive buffers allocated, to be shared by all configured stations.

It is recommended that the above offsets are not hard-coded in your application. Use the value (BUFFER\_LENGTH-2) as the starting offset for this information, in the event that the configuration structure is lengthened in the future.

## **SET\_MODEM\_STATUS (0x10)**

Provides manual control of the state of DTR and RTS.

### **Control Block values to be set on entry:**

**BUFFER\_**

**LENGTH:** Set to 0x01.

**DATA:**

Offset 0x00

#### **Modem byte**

Bit 0 Set to 1 to maintain DTR high; set to 0 to maintain DTR low.

Bit 1 Set to 1 to maintain RTS high; set to 0 to maintain RTS low.

### **Control Block values set on return:**

**RETURN\_**

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

## **READ\_MODEM\_STATUS (0x11)**

Reads the current state of DCD and CTS.

### **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.

**BUFFER\_**

**LENGTH:** Set to 0x01.

**DATA:**

Offset 0x00

#### **Modem byte**

Bit 3 Set to 1 to indicate DCD is high; set to 0 to indicate DCD is low.

Bit 5 Set to 1 to indicate CTS is high; set to 0 to indicate CTS is low.

## READ\_CODE\_VERSION (0x12)

Return the code version for the BSC code on the SDLA card.

### 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.

BUFFER\_

LENGTH: Set to 0x06.

DATA:

Offset 0x00-0x01 Code type: "PP" for **Point-to-Point**, and "MP" for **Multipoint**.

Offset 0x02-0x05 Version information in the format: X.YY, where X is a major version number, and YY is a minor version number.

## **ADD\_STATION (0x20)**

### **(Multipoint Only)**

Configures a new tributary station. If configured on **Primary**, polls will be sent to this station. If configured on **Secondary**, polls from the **Primary** will be responded to.

#### **Control Block values to be set on entry:**

**BUFFER\_**

**LENGTH:** Set to 0x03.

**CU\_POLL\_**

**ADDRESS:** Set to the control unit polling address of this station.

**CU\_SELECT\_**

**ADDRESS:** Set to the control unit selection address of this station.

**CU\_DEVICE\_**

**ADDRESS:** Set to the device address of this station.

**DATA:**

Offset 0x00 Maximum size of transmit queue in number of frames. Set to 0 to use the physical limit only. If the number of frames specified exceeds the total number of transmit buffers, the physical limit will be used.

Offset 0x01 Maximum size of receive queue in number of frames. Set to 0 to use the physical limit only. If the number of frames specified exceeds the total number of receive buffers, the physical limit will be used.

Offset 0x02 **Station flags**

Bit 0: 1 = Station initially assumed to be active.

0 = Station activated when polled (**Secondary**), or received a response to a poll (**Primary**).

Bit 7: 1 = Station is enabled (responds to or issues polls).  
0 = Station is disabled (does not respond to or issue polls until later enabled).

**Control Block values set on return:**

RETURN\_

CODE: 0x00 The action has been performed successfully.

0x02 The specified addresses reference a station that is already configured.

0x07 The maximum number of configured stations has already been reached (32).

**Valid if RETURN\_CODE is 0x00:**

STATION: Set to the station number to be used for all subsequent station operations.

## **DELETE\_STATION (0x21)**

### **(Multipoint Only)**

This command is used to delete a station configuration. Any frames currently queued for transmission or reception on this station are discarded.

#### **Control Block values to be set on entry:**

**BUFFER\_**

**LENGTH:** Set to 0x00.

**STATION:** Set to station number to delete.

#### **Control Block values set on return:**

**RETURN\_**

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

0x07 The specified station number is invalid.

## **DELETE\_ALL\_STATIONS (0x22)**

### **(Multipoint Only)**

This command is used to delete all configured stations. Any frames currently queued for transmission or reception on these stations are discarded.

#### **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.

## **LIST\_STATIONS (0x23)**

### **(Multipoint Only)**

Lists all configured stations and includes configuration, status, and transmit and receive queue information for each.

#### **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.

BUFFER\_  
LENGTH: Set to the number of configured stations multiplied by 10 (0x0A).

DATA:

The information is returned as an array of structures. An information structure is defined as follows:

Offset 0x00 Station number.  
Offset 0x01 Station status flags:  
Bit 0: Set to 1 if station is active.  
Bit 1: Set to 1 if a general poll is issued to this station (**Primary**). All stations will have this bit set if in general polling mode.  
Bit 6: Set to 1 if configured (always).  
Bit 7: Set to 1 if enabled, or 0 if disabled.

Offset 0x02 Polling address.  
Offset 0x03 Selection address.  
Offset 0x04 Device address.  
Offset 0x05 Number of text frames discarded on transmission.  
Offset 0x06 Maximum transmission queue size in frames.  
Offset 0x07 Maximum reception queue size in frames.  
Offset 0x08 Number of text frames queued for transmission.  
Offset 0x09 Number of text frames queued for reception.

## **SET\_GENERAL\_OR\_SPECIFIC\_POLL (0x24)**

### **(Multipoint Primary Only)**

Use this command to set the polling method by the **Primary**. Specific device polling is enabled by default.

This is the only means of changing the polling mode. SET\_CONFIGURATION will not reset the mode to specific polling if changed to general polling using this command.

**Warning:** If operating in general poll mode, all configured stations that have the same poll address will be marked active if the general poll is responded to by the **Secondary**. This does not necessarily indicate that all these stations actually exist, as the response may be for any one station only, for example.

### **Control Block values to be set on entry:**

BUFFER\_

LENGTH: Set to 0x00.

DEVICE\_

ADDRESS: Set to 0x00 to poll specific device addresses.

Set to 0x01 to use the general polling address.

### **Control Block values set on return:**

RETURN\_

CODE: 0x00 The action has been performed successfully.

## SET\_STATION\_STATUS (0x25)

### (Multipoint Only)

This command is used to set the status flags of a particular station. It should normally not be used except under unusual circumstances.

#### Control Block values to be set on entry:

BUFFER\_

LENGTH: Set to 0x01.

STATION: Station number to set flags for.

DATA:

Offset 0x00

#### Station flags

Bit 0: 1 = Station assumed to be active until otherwise deemed inactive.

0 = Station assumed to be inactive until activated when polled  
(**Secondary**), or until received a response to a poll (**Primary**).

Bit 7: 1 = Station is enabled (responds to or issues polls).

0 = Station is disabled (does not respond to or issue polls until later  
enabled).

#### Control Block values set on return:

RETURN\_

CODE: 0x00 The action has been performed successfully.

0x07 The specified station has not been configured.

## **READ\_STATE\_DIAGNOSTICS (0x30)**

Command to read information from the state machine that indicates events that took place which were not handled, due to them not being expected. This command may be ignored unless unusual protocol problems occur, in which case this will return information useful for debugging purposes and should be passed on to Sangoma along with a description of the problem.

### **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.

**BUFFER\_**

**LENGTH:** Set to 0x1E.

**DATA:**

The first 15 bytes is an array indexed on state number, whose elements contain the number of unhandled events. The next 15 bytes is an array indexed on state number, whose elements contain the code of the last unhandled event. The state numbers are the same as those found under the LINK\_STATUS command description. The event codes are internal numbers to be decoded by Sangoma.

## 5. Event Codes Returned in NOTIFY

After each command call, independent of the type of call or the return code, the NOTIFY mailbox field may be set to indicate events that took place since the last command. In some cases in **Multipoint**, the NOTIFY\_EXTENDED field may also be set to return additional information on an event. Events are indicated by bits being set in NOTIFY, and therefore more than one different type of event may be returned on the same command call. However, successive events of the same type will not be reflected in NOTIFY as only one bit is devoted to any one event.

### Point-to-Point

Bit Set	Event Description
0	Transmit retry limit exceeded on a text frame due to no response or NAK. The frame was discarded.
1	No response to line bid. Entering slow poll mode.
2	No EOT received in text receive mode, and will no longer expect one. Returning to an idle state. Any further text received will be discarded until the line is bid for again.
3	Received a DISCONNECT circuit assurance message. The current state is now circuit assurance mode.
4	Timeout occurred while waiting for CTS on transmission of a control frame.
5	Timeout occurred while waiting for CTS on transmission of a data frame.
6	Reserved.
7	A change in the state of DCD and/or CTS occurred. This is only set if Bit 1 is set in <b>protocol_options1</b> .
8	Current state of DCD (0=low, 1=high). Only valid if Bit 7 is set.
9	Current state of CTS (0=low, 1=high). Only valid if Bit 7 is set.

## Multipoint

Bit Set	Event Description
0	Transmit retry limit exceeded on a text frame due to no response or NAK. The frame was discarded. NOTIFY_EXTENDED is set to the associated station number.
1	Retry limit exceeded on selection of a device as no response was received. The transmit queue for this station will be flushed. NOTIFY_EXTENDED is set to the associated station number.
2	No EOT received in text receive mode, and will no longer expect one. Returning to an idle state. Any further text received will be discarded until another poll/select occurs.
3	Transmit buffers were flushed in the station queue, usually due to a station going inactive. NOTIFY_EXTENDED is set to the associated station number.
4	Timeout occurred while waiting for CTS on transmission of a control frame.
5	Timeout occurred while waiting for CTS on transmission of a data frame.
6	A station is going inactive after not having received polls on the <b>Secondary</b> , or responses to polls on the <b>Primary</b> . NOTIFY_EXTENDED is set to the associated station number.

## 6. Streaming Receive Mode

This receive mode stores for the user all data received following the first SYN character and up to a fixed number of consecutive PAD characters, marking the end of frame. It does not parse the frame, and does not check the BCC. It is used in conjunction with passive receive mode.

The interface to this mode is the same as described in this manual for the BSC.502 **Point-to-Point** code, with the following exceptions.

### *Configuration*

The following configuration parameters that are used for **Point-to-Point** are also used by the **Streaming Receive Mode**:

io\_port  
mem\_segment  
mem\_window  
multi\_point (=0)  
cpu\_speed  
line\_speed  
max\_data\_frame\_size  
num\_consec\_PAD\_eof  
switched\_CTS\_RTS (minor role: used only to determine if RTS should be raised on BSC\_OPEN command)  
EBCDIC\_encoding (determines SYN character value)  
auto\_open  
misc\_bits

All other parameters must still be present for BLOAD to be used but are ignored by the Streaming mode. If SET\_CONFIGURATION is used, the extraneous parameters may be set to anything.

Passive streaming receive is activated by setting bits 0 and 1 in **misc\_bits**.

## *Commands*

### **BSC\_READ**

HEADING\_LENGTH is always zero.

BUFFER\_LENGTH is the total frame size, which may include extraneous prefixed SYN characters, and will include all PAD characters at the end less one.

MISC\_TX\_RX\_BITS will always be 0x80. This does not imply that the frame ends with ETX, as it would with **Point-to-Point**.

DATA includes all received bytes as described above.

### **LINK STATUS**

The current state will always be either 0x00 (not open) or 0x0D (passive receive). The transmit-related fields will show a transmit buffer is available, and no frames are queued for transmission. One transmit buffer is always allocated but not used.

### **READ OPERATIONAL STATISTICS**

The only relevant statistics are:

Offset 0x00-0x01: Received frames processed. If buffers are full, this value still reflects the number of received frames, whether they have been discarded or not.

Offset 0x36-0x37: What was ETX received now is the number of frames received and successfully stored in the BSC\_READ queue.

### **READ COMMS ERROR STATISTICS**

The only relevant statistics are:

Offset 0x00: Receiver overruns

Offset 0x06: DCD changes

Offset 0x07: CTS changes

### **READ BSC ERROR STATISTICS**

The only relevant statistics are:

Offset 0x02-0x03: Receiver discards, no buffers.

Offset 0x0A-0x0B: Receiver discards, too long.

Offset 0x1A-0x1B: Number of times receiver was halted.

Offset 0x1C-0x1D: Number of catastrophic receiver restarts.

Offset 0x1E-0x1F: Frame parser reached end of buffer prematurely (not expected).

## *Mailbox*

**Field NOTIFY**

Will always be zero, as no events take place in this mode.

## 7. BSC Feature Limitations

The following is a list of some specific features of BSC that are not supported or are limited by this implementation. This list may change often as specific needs arise.

Successive intermediate text blocks (ITBs) must be preceded with STX or SOH on reception. They must also be separated by at least the configured number of PAD characters that mark end-of-frame.

Reverse Interrupt (RVI) is handled on reception, but is not transmitted.

Alternating ACKs (ACK0, ACK1) are accepted but are interpreted as an individual ACK. No check is made that they indeed alternate.

Sync-idle sequences are filtered out on reception, but are not transmitted.

Limited conversational mode is always enabled on reception. Only one ETX block constitutes a conversational reply. ITB/ETB is silently discarded as a reply. Once the reply is acknowledged, the poll/select/line-bid phase terminates without EOT being sent or expected. If no buffers are available to store a conversational reply, it is NAK'd at which point it is assumed that it will either be sent again, or an EOT is sent instead.

When in circuit assurance mode, the user must send a CONNECT message before text can be sent. However, if text is received in circuit assurance mode, the mode exits as it would if a CONNECT message was received.

In some cases an SOH-only frame, without following text data, is supported, but not by design. It should be avoided if possible.

Due to flaws in the BSC protocol specification, **NO TEXT FRAME IS GUARANTEED TO BE DELIVERED BY THE SDLA BSC CODE**. It is also remotely possible that duplicate frames may be received by the user application due to a bad link or inappropriately-set retransmission timers.

## 8. MS-DOS Software

For the ISA S503 there is a set of MS-DOS utilities that can be used to work with the protocol.

### **BLOAD.EXE**

**BLOAD.EXE** loads the BSC code onto the adapter, and has the following command line syntax:

```
bload [-cCODEFILE] [-fCONFIGFILE]
```

where:

**CODEFILE** The name of the file containing the code for BSC, e.g. BSC.502 for **Point-to-Point**, or BSCMP.502 for **Multipoint**.

**CONFIGFILE** The name of the file containing configuration information. See the section **Configuration Parameters**.

The default name for **CODEFILE** is BSC.502. The default name for **CONFIGFILE** is BSC.SDL.

### **BSC.502/BSCMP.502**

This is the BSC support code which is loaded onto the card and establishes the link. They are **NOT** MS-DOS executable programs. BSC.502 is the **Point-to-Point** code, and BSCMP.502 is the **Multipoint** code.

### **BTEST.EXE**

**BTEST.EXE** is a powerful testing tool, and has the following command line syntax:

```
btest [-mMEM] [-pPORT] [-b]
```

where:

**MEM** is the memory location where the BSC code was loaded in the format **QR**. **Q** refers to the segment:

Value of Q	Segment
A	0xA000
C	0xC000
D	0xD000
E	0xE000

**R** refers to the window, identical to the **mem\_window** value as described in the **Configuration Parameters** section. An example memory specification is **-md2**, when the code was loaded in segment 0xD000 window number 2 (or 0xD000:0x2000).

**PORT** is the base port address: 250, 300, 350, or 360.

The **-b** option specifies the use of black and white only for the choice of colours in the program display.

**Example:** `btest -md2 -p250 -b`

## Error Messages

### BLOAD.EXE

If BLOAD 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 BLOAD"

An invalid command line argument was used (exit code of 1).

"The file FILENAME was not found"

A filename listed in the command line arguments was not found in the defined directory (exit code of 2).

"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 4).

"The downloaded code is not running on the adapter"

The SDLA CPU has halted. Contact your Sangoma representative (exit code of 5).

# 9. Configuration Parameters under MS-DOS

## Setting Configuration Using Parameter File

The BSC configuration must be set using a parameter file given to the code loader BLOAD. Subsequent changes to the configuration are made using the SET\_CONFIGURATION command. A parameter entry is a line of text:

`<name><space>=<space><value>`

The format of the value field depends on the parameter type. Examine the sample file below to observe these formats. Note that the loader does not verify the parameter values - just the presence of them. Read the section on the command SET\_CONFIGURATION carefully to determine the correct ranges for these values. The **name** string is identical to the field names shown under SET\_CONFIGURATION.

The code loader BLOAD requires additional parameters to be located in the file:

- io\_port**                    The I/O base port address of the SDLA card as jumpered. The standard addresses are: 0x250, 0x300, 0x350, 0x360.
- mem\_segment**               Defines the segment location of the 8K shared memory window which the PC sees as being occupied by the SDLA card. Valid segments are 0xA000, 0xC000, 0xD000, and 0xE000.

**mem\_window**               Defines the particular 8K window location inside the segment. Valid windows are:

Value	PC memory window (hex)
0	0000 to 1FFF
2	2000 to 3FFF
4	4000 to 5FFF
6	6000 to 7FFF
8	8000 to 9FFF
A	A000 to BFFF
C	C000 to DFFF

**cpu\_speed**               Set to the CPU speed of the SDLA card. If a crystal is present on the card, the

CPU speed is half the value indicated. If no crystal is present, it is 7.2 MHz.

<b>cpu_speed</b>	<b>CPU speed</b>
1	3.6 MHz
2	7.2 MHz
3	8.0 MHz
4	10.0 MHz

**line\_speed** Although the line speed is set using an index number in SET\_CONFIGURATION, the loader requires an actual value in bps. Valid line speeds are: 0 (external), 1200, 2400, 4800, 9600, 19200, 38400, 45000, 56000, 74000.

**multi\_point** To tell the loader which parameters to look for, this needs to be set to 1 if the **Multipoint** code is being loaded, or 0 if the **Point-to-Point** code is being loaded.

## Sample Parameter File

Note that all entries are shown whether or not they are needed for the specified mode of operation.

```
io_port                = 0x360
mem_segment            = 0xD000
mem_window             = 0
multi_point            = 1
cpu_speed              = 2
line_speed             = 19200
max_data_frame_size   = 500
secondary_station     = 0
num_consec_PAD_eof    = 5
num_add_lead_SYN      = 2
conversational_mode   = 1
pp_dial_up_operation  = 0
switched_CTS_RTS     = 0
EBCDIC_encoding       = 1
auto_open             = 1
misc_bits              = 0x00
protocol_options1     = 0x00
protocol_options2     = 0x00
```

```
max_retransmissions      = 10
fast_poll_retries        = 10
TTD_retries               = 3
restart_timer            = 10
pp_slow_restart_timer    = 1000
TTD_timer                 = 50
pp_delay_between_EOT_ENQ = 100
response_timer           = 50
rx_data_timer            = 100
NAK_retrans_delay_timer  = 40
wait_CTS_timer           = 50
mp_max_consec_ETX        = 5
mp_general_poll_address  = 0x7F
sec_poll_timeout         = 2000
sec_additional_stn_send_gpoll = 3
pri_poll_skips_inactive  = 20
pri_select_retries       = 10
mp_multipoint_options    = 0x00
```