

**MULTIPROTOCOL SOFTWARE
USER'S MANUAL**

**MULTIPROTOCOL
USER'S MANUAL
MCX, MCX-LITE/S AND MCX-LITE/570 CARDS**

COPYRIGHT (©) ACKSYS 1993-1998

This document contains information that is protected by copyright.

No part of this publication may be reproduced, transcribed, stored on any computer system or other system, translated into any language or into any computer language without prior written permission from ACKSYS, 3 & 5 rue du Stade, BP 80, 78302 POISSY CEDEX, FRANCE.

REGISTERED TRADEMARKS ®

- ACKSYS is a registered trademark of ACKSYS.
- IBM P.C, AT are registered trademarks of International Business Machines Corporation.

NOTICE

ACKSYS ® provides this documentation "as is" without warranty of any kind. In no event shall ACKSYS be held responsible for the profitability and conformity of the hardware compared to the user's requirements.

ACKSYS ® shall not be held responsible for any errors that might be contained in this document, nor for any damages of any amount that the supply, operation or use of this equipment may entail.

ACKSYS ® may revise this document from time to time, or change its contents, without notice.

Société ACKSYS
3 & 5 rue du Stade
BP 4080
78302 POISSY CEDEX

Telephone : +33 1-39-11-62-81
Fax : +33 1-39-11-47-96
Email : ACKSYS@compuserve.com

CONTENTS

I.	INTRODUCTION	I-1
I.1.	MANUAL CONTENTS	I-1
I.2.	CONVENTIONS USED	I-1
I.3.	OVERVIEW OF THE MULTIPROTOCOL PROGRAM.....	I-2
I.4.	INSTALLING THE MULTIPROTOCOL PROGRAM IN THE CARD'S EPROM.....	I-3
I.5.	OVERVIEW OF COMMUNICATION MODES SUPPORTED	I-4
I.5.1.	<i>ASYNCHRONOUS mode</i>	I-4
I.5.2.	<i>SYNCHRONOUS mode</i>	I-5
II.	DIALOG PROCEDURE BETWEEN THE CARD AND THE PC.....	II-17
II.1.	SUBDIVISION OF THE MAILBOX	II-18
II.2.	SETTING UP THE PROGRAM	II-21
II.3.	SENDING COMMANDS TO THE CARD.....	II-22
II.3.1.	<i>Error code management</i>	II-23
II.4.	INTERRUPT MANAGEMENT.....	II-24
II.4.1.	<i>End-of-command interrupts</i>	II-25
II.4.2.	<i>"Event" interrupts</i>	II-28
II.4.3.	<i>Acknowledging interrupts</i>	II-40
II.4.4.	<i>The interrupt queue</i>	II-40
II.5.	USING THE CARD IN "POLLING" MODE	II-41
III.	INITIALIZING CHANNELS IN SOFTWARE	III-43
IV.	INTERPRETING THE GROUP OF 8 LEDS.....	IV-45
V.	THE COMMAND INTERPRETER.....	V-47
V.1.	PROTO (2Fh): INITIALIZE THE PROTOCOL USED BY A CHANNEL	V-47
V.2.	ALLOC (01H): ALLOCATE BUFFER MEMORY FOR AN ASYNCHRONOUS CHANNEL	V-59
V.3.	BDELE (06H): CLEAR TX BUFFER AND STOP	V-61
V.4.	BPARM (18H): READ BUFFER PARAMETERS	V-62
V.5.	BREAK (23H): SEND A BREAK ON AN ASYNCHRONOUS CHANNEL.....	V-66
V.6.	BTRAN (08H): DATA TRANSMISSION	V-67
V.7.	CHDEF (0AH): DEFINE THE CHARACTER STRING FOR AN ASYNCHRONOUS CHANNEL.....	V-70
V.8.	CLRRX (1DH): CLEAR DATA RECEIVED.....	V-72
V.9.	DALOC (15H): DEALLOCATE ASYNCHRONOUS BUFFER MEMORY	V-73
V.10.	GOADR (12H): EXECUTE A PROGRAM IN MEMORY.....	V-74
V.11.	HNGUP (16H): HANG UP MODEM	V-75
V.12.	MBOOT (05H): LOAD A PROGRAM IN MEMORY.....	V-76
V.13.	MINTR (0Ch): "EVENT" INTERRUPT CONDITIONS.....	V-78
V.14.	NOPER (2BH): NOP COMMAND (NO OPERATION).....	V-82
V.15.	PRCTL (30H): CONTROL PROCEDURES.....	V-83
V.16.	RDBUF (09H): READ A RECEIVE BUFFER.....	V-89
V.17.	RELRP (13H): READ CODES, REVISIONS, IDENTIFICATIONS	V-93
V.18.	RINIT (19H): REINITIALIZE THE CARD.....	V-96
V.19.	RMEMO (11H): DUMP A BLOCK OF MEMORY.....	V-97
V.20.	RSMDE (2DH): INITIALIZE ELECTRICAL MODE.....	V-99
V.21.	RSTAT (0DH): READ THE STATUS OF THE COMMUNICATION CHANNELS.....	V-100
V.22.	RXCNT (1BH): READ THE NUMBER OF CHARACTERS OR FRAMES RECEIVED.....	V-102
V.23.	RXENB (04H): ENABLE OR DISABLE RECEPTION	V-104
V.24.	STCNT (0BH): DEFINE THE SIZE OF RECEIVE BLOCKS FOR AN ASYNCHRONOUS CHANNEL.....	V-105
V.25.	STSIG (24H): MANUALLY CONTROL DTR AND RTS.....	V-107
V.26.	STTMO (1Fh): INITIALIZE THE RECEIVE TIMEOUT	V-109
V.27.	TFREE (1Ah): READ THE SPACE AVAILABLE IN TRANSMISSION	V-111
V.28.	VINIT (00H): INITIALIZE COMMUNICATION PARAMETERS	V-113
V.29.	VMODE (03H): INITIALIZE FLOW CONTROL PARAMETERS	V-124

VI.	THE COMMAND INTERPRETER IN X25 MODE	VI-133
VI.1.	LOCATION OF THE X25 COUNTERS IN THE DATA ZONE	VI-133
VI.2.	X25 COMMANDS	VI-135
VI.2.1.	<i>MINTR (0Ch): X25 "event" interrupt conditions.....</i>	<i>VI-136</i>
VI.2.2.	<i>PRCTL (30h):X25 Control Procedure.....</i>	<i>VI-138</i>
VI.2.3.	<i>BTRAN (08h): Send a data packet or an interrupt packet.....</i>	<i>VI-176</i>
VI.2.4.	<i>RDBUF (09h): Read data in a data packet or an interrupt packet.....</i>	<i>VI-179</i>
VII.	APPENDICES.....	VII-183
VII.1.	COMMAND SUMMARY SORTED BY OPCODE	VII-183
VII.2.	ERROR CODE SUMMARY	VII-185
VIII.	NOTES.....	VIII-187
IX.	EVALUATION SHEET	IX-191

I. Introduction

I.1. Manual contents

This manual is designed for programmers developing drivers for the MCX, MCX-*Lite/S* and MCX-*Lite/570* cards, as well as application developers who, for performance reasons, wish to use the cards without using the drivers supplied.

The six chapters in the manual provide all the details you need to obtain optimal performance from the card and its multiprotocol software. Here is a brief description of the manual layout:

Description of supported protocols.

Dialog procedure between the PC and the card using commands and interrupts.

Procedure for starting the program and initializing the channels.

Meaning of the various LEDs¹ on the card in relation to the control, transmission or reception operations.

The command interpreter program.

The command interpreter program in X25 mode.

We recommend you read this document from start to finish.

I.2. Conventions used

Several conventions are used in this manual:

The first concerns how numbers are written using various systems.

Numbers in the decimal system are written with no prefix nor suffix (for example: 15).

Hexadecimal numbers are followed by an "h" (example: 0Fh).

Binary numbers are followed by a "b" (example: 00001111b).

The second convention was adopted to simplify the wording. Since this manual is designed for users of the MCX, MCX-*Lite/S* and MCX-*Lite/570* cards, we have decided to not specify the type of card when information concerns all three cards.

The final convention concerns the typesetting used for describing commands: parameters in italics are reserved, but are not implemented or partially implemented.

¹ LED: Light Emitting Diode

I.3. Overview of the multiprotocol program

The multiprotocol program, developed exclusively for the MCX family of cards with a ZILOG 85C30 or SCA HITACHI HD64570-16 communication controller, allows you to program each of the card's channels using one of the following protocols:

In bit-oriented synchronous mode:

- HDLC at the frame envelope level
- HDLC in ABM mode at the data link level
- X25 at the packet level

In character-oriented synchronous mode:

- BISYNC at the frame envelope level

In asynchronous mode:

- No protocols are supported. Several data flow control methods are supported (Xon/XOff, RTS/CTS, DTR/CTS ...)

Revision history

The first distributed version of the program was V1.2. Previous versions require specific updates.

The HDLC protocol in ABM mode (or LAPB) was implemented in V1.5.

The X25 protocol was implemented in V1.8.

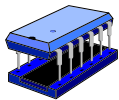
I.4. Installing the multiprotocol program in the card's EPROM

■ If you purchased the software with the card, the program is already loaded in the EPROM (or the Flash EPROM). In this case, no special installation procedure is required.

■ If you purchased the software after the card, however, you will receive:



Either a diskette entitled “**Installation for the multiprotocol software**” allowing you to add the multiprotocol software in your card's Flash EPROM,



or an EPROM or Flash EPROM containing the multiprotocol software. In this case, pay careful attention when you replace the EPROM (reference U21).

The “Installation of the multiprotocol software” diskette is a DOS diskette containing the following files:

- COMMAND.COM, CONFIG.SYS, AUTOEXEC.BAT: System files
- MCX.DAT: BIOS for MCX, MCX-Lite/S or MCX-Lite/570 cards
- MCXFLASH.EXE: FLASH EPROM update program
- MMCX.BIN: Multiprotocol software for the 8 to 64-channel MCX card
- MCX.BIN: Basic software for the 8 to 64-channel MCX card
- MLITESER.BIN: Multiprotocol software for the MCX-Lite/S card
- LITESER.BIN: Basic software for the MCX-Lite/S card
- MLITE570.BIN: Multiprotocol software for the MCX-Lite/570 card
- LITE570.BIN: Basic software for the MCX-Lite/570 card

To load the multiprotocol software in the card's Flash EPROM, simply:

- make sure that jumpers ST2 and ST3 are set to 1-2
- know the base address and the interrupt level used by the card
- and restart your PC with the diskette entitled “Installation of the multiprotocol software” in drive **A:**, then follow the instructions (Selecting the card and its address). After the software has been loaded (and you return to the DOS command interpreter), remove the diskette from the drive and restart the PC. The group of 8 LEDs on the card should count to bit 8 after the self-test phase. If not, check that the Flash EPROM is inserted in the proper direction, and that none of its pins are bent. If no error appears after making all these checks, contact ACKSYS technical support.

I.5. Overview of communication modes supported

I.5.1. ASYNCHRONOUS mode

In asynchronous transmission mode, data is transferred a single data bit at a time, from one device to another. Within the data flow, each character is carried in a binary "frame". This frame includes a start bit indicating the start of the "frame" (or character), from 5 to 8 data bits with the character's binary code, an optional parity bit for error detection, and finally one or more stop bits to indicate the end of the "frame".

Start (1 bit)	Character (5 to 8 bits)	Parity (0 or 1 bit)	Stop (1, 1.5 or 2 bits)
---------------	-------------------------	---------------------	-------------------------

Bit-level characteristics:

- NRZ-encoded characters on 5, 6, 7 or 8 bits
- Programmable parity (none, even or odd)
- Programmable standard speeds from 50 bits/s to 250,000 bits/s for all the extensions.
- "Synchronized" asynchronous mode supported that generates programmable transmit and receive clocks

I.5.2. SYNCHRONOUS mode

Contrary to asynchronous mode, the basic principle of synchronous mode is the use of synchronization characters to synchronize the transmitter and receiver devices. Data is always sent or received at a throughput defined by an internal or external clock. Synchronous communications include 2 groups of protocols:

Character-oriented protocols (COPs): MONOSYNC and BISYNC
Bit-oriented protocols (BOPs): HDLC and SDLC

I.5.2.1. The BISYNC protocol

In this programming mode, the card does not provide transmit or receive error correction procedures (only reception error checking is performed). All frames received on the link may be read and processed by the PC.

BISYNC uses the following frame structure:

SYN 1	SYN 2	information block	BCC
-------	-------	-------------------	-----

The SYN1 SYN2 synchronization sequence (see the PROTO command) is a 16 bit word added at the start of each frame. The information block contains user data, and the card adds no characters or sequence of characters as a header.

The BCC sequence (Block Checking Calculation) is not generated in the current version.

The incoming control signal drops to indicate the end of the information block.

The block size is limited to 2048 bytes.

Bit-level characteristics:

- Data encoded on 8 bits in NRZ, NRZi, FM0 or FM1 format
- No parity
- Internal clocks are programmable from 120 bits/s to 1Mbits/s for the MCX-BP and LITESERIAL extensions
- Internal clocks programmable from 122 bits/s to 4 Mbits/s for the LITE570 extension
- Programmable generation of transmit and receive clocks
- Binary throughput limited to 1Mbits/s on channels 1, 2 and 3 for the MCX-BP extension
- Binary throughput limited to 125 Kbits/s on channels 4 and higher for the MCX-BP extension
- Binary throughput limited to 1Mbits/s on all channels for the LITESERIAL extension
- Binary throughput limited to 4 Mbits/s on all channels for the LITE570 extension
- DTR signal reserved for internal use on channels 1 and 2 for the LITESERIAL extension and channels 1, 2 and 3 for the MCX-BP extension
- RS422 clock signals TxCLOCK and RxCLOCK are not available on channels 4 and higher for an MCX-BP extension

For example, an application can divide the information block for the BSC or BISYNC protocol, developed by IBM, using the following structure:

SOH	Header	STX	Data	ETB or ETX
-----	--------	-----	------	------------

This protocol uses the 128 characters of the ASCII character set. Some of these characters are used as commands:

- SOH = Start Of Header
- STX = Start of TeXt
- ETB = End of Transmission Block
- ETX = End of TeXt
- EOT = End Of Transmission
- ENQ = ENQuiry
- ACK = ACKnowledge
- DLE = Data Link Escape
- NAK = Negative Acknowledge

BSC handles two message types: service or supervision messages, and information messages.

In an information message, the ETB character indicates "temporary or intermediate end of an information block". It is used to delimit information blocks that belong to a single set of data. In this case, the last information block includes the ETX character, which indicates "end of information block".

I.5.2.2. The HDLC/SDLC synchronous frame envelope-level protocols

In this programming mode, the card does not provide transmit or receive error correction procedures (only reception error checking is performed). All frames received on the link may be read and processed by the PC.

HDLC uses the following frame structure:

Flag (7Eh)	User Data Information Block	FCS (16 bits)	Flag (7Eh)
------------	-----------------------------	---------------	------------

Flag: frame-delimiting sequence

FCS: frame checking sequence

The flag

As shown in the preceding illustration, all frames sent and received by the card must start and end with a frame-delimiting sequence, called the flag.

A single flag can be used both as an end flag for one frame and as a start flag for the next frame.

The information block

The information block contains the user data. It contains at least one byte, and cannot exceed 2048 bytes.

The FCS ("Frame Checking Sequence")

The FCS is a series of 16 bits that let the receiver check the validity of the transmitted frame.

It is calculated using the CRC-CCITT polynomial generator: $x^{16}+x^{12}+x^5+1$.

Inserting "0"s

To make sure that flags are not simulated inside a frame, the transmitter adds a "0" after any sequences of five consecutive "1"s. The receiver detects these "0"s and removes them automatically.

Bit-level characteristics:

- Data encoded on 8 bits in NRZ, NRZi, FM0 or FM1 format
- No parity
- Internal clocks are programmable from 120 bits/s to 1Mbits/s for the MCX-BP and LITESERIAL extensions
- Internal clocks programmable from 122 bits/s to 4 Mbits/s for the LITE570 extension
- Programmable generation of transmit and receive clocks
- Binary throughput limited to 1 Mbits/s on channels 1, 2 and 3 for the MCX-BP extension
- Binary throughput limited to 125 Kbits/s on channels 4 and higher for the MCX-BP extension
- Binary throughput limited to 1Mbits/s on all channels for the LITESERIAL extension
- Binary throughput limited to 4 Mbits/s on all channels for the LITE570 extension
- DTR signal reserved for internal use on channels 1 and 2 for the LITESERIAL extension and channels 1, 2 and 3 for the MCX-BP extension
- RS422 clock signals TxCLOCK and RxLOCK are not available on channels 4 and higher for an MCX-BP extension

I.5.2.3. The HDLC ABM or LAPB data link-level protocol

In this programming mode, the card manages the HDLC protocol layer at the data link level, following the X25 protocol's LAPB recommendations (Link Access Procedure, Balanced Mode). Only the basic operations of this protocol are implemented (modulo 8 numbering for information frames, look-ahead window K whose width is less than or equal to 7). It is similar to the HDLC protocol in ABM mode (as named by the ISO).

HDLC uses the following frame structure:

Flag (7Eh)	Address (8 bits)	Command (8 bits)	Data	FCS (16 bits)	Flag (7Eh)
------------	------------------	------------------	------	---------------	------------

The *flag* and *FCS* fields are documented in the section describing the HDLC frame envelope.

Here follows a complete list of the frames recognized in the LAPB protocol (HDLC ABM):

Information (Command)
Supervision (Reply/Command)
RR-Ready to receive
REJ-Reject
RNR-Not ready to receive
Unnumbered
DISC-Disconnect (Command)
UA-Acknowledge (Reply)
FRMR-Reject frame (Reply)
SABM-Connect/Reinitialize (Command)
DM-disconnected mode indicator (Reply)

Only the data fields in the information frames are exchanged between the PC and the card.

The HDLC protocol offers a transport procedure with no data loss and maintains the order of data manipulated by the upper layer using the following mechanisms:

- numbers data frames (modulo 8)
- acknowledges information frames
- repeats information frames when T1 timer times out
- receiver can request retransmission
- connect/disconnect (SABM/DISC)
- look-ahead window of width K (at most K information frames unacknowledged by the receiver at time t)

The *Address* field differentiates the command frames from reply frames on point-to-point lines. The table below describes the use of this address. This field can be configured using the PROTO command.

	DCE (network)	DTE
Command	A (03)	B (01)
Reply	B (01)	A (03)

The *command* field identifies the type of frame.

The *data* field is located in the information frames (see the BTRAN and RDBUF commands) and contains user data (this field is also present in the FRMR frames, but in that case it contains other types of data).

Definition of the K parameter

The K parameter defines the width of the look-ahead window. It lets the MCX card ask the equipment connected at the other end to never send more than K look-ahead frames (before having received the acknowledgment of the first frame). The MCX card also follows this rule. K is assigned a value from 1 to 7 (see the PROTO command).

The RDBUF and BTRAN commands can access K receive buffers and K transmit buffers to let the MCX card exchange information frames. These commands only exchange the *data* field with the PC.

When the K receive buffers are full, the card sends an RNR command to refuse the next information frames. As of that time, at least $\frac{K+1}{2}$ frames must be read by the RDBUF command to enable the card to acknowledge the next frames.

When the receiver acknowledges an information frame sent by the card, the associated transmit buffer is freed.

Definition of the T1 parameter

The T1 timer is used when sending an information frame, or an SABM or DISC command.

When the timer times out, it triggers a retransmission of the first unacknowledged information frame from either the SABM or the DISC frame, depending on the circumstances. A speed-independent default value was chosen. This is slightly less efficient than a speed-dependent value, but given the average trigger frequency of the timer, this loss of efficiency is negligible.

N2 is a parameter associated with the T1 timer. N2 defines the maximum number of resends after a T1 timeout.

Miscellaneous remarks

N1, the maximum size of an information frame (excluding the flags and any added transparency bits), is set at 2052 bytes.

The T2 timer, which defines the maximum time available to the receiver to acknowledge reception, is not used. Frames are acknowledged as soon as possible.

The T3 timer, which defines the time after which the reception of consecutive "1"s triggers the RR command mechanism, is not used.

The PRCTL command handles the connection/disconnection of the LAPB (HDLC ABM), using the LAPLNKUP and LAPLNKDN opcodes (see the PRCTL command in section V.15).

1.5.2.4. The X25 protocol

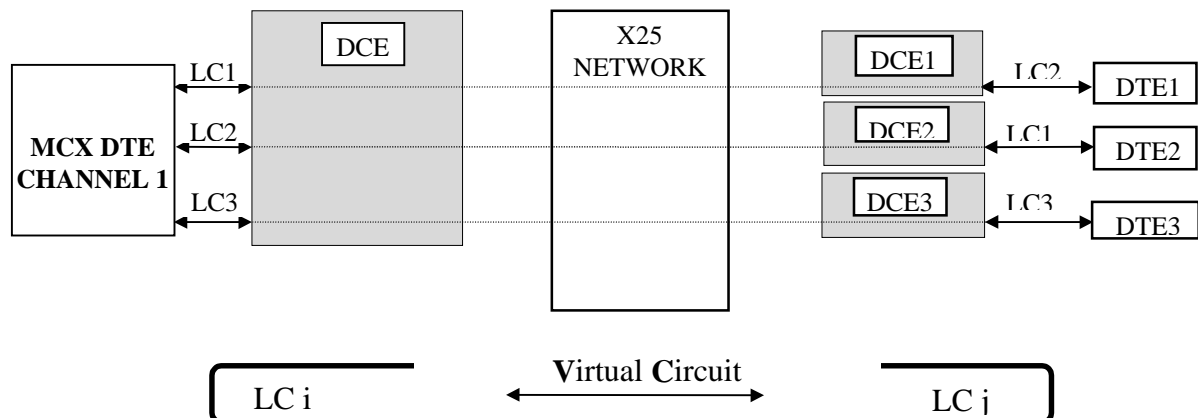
This section provides information under the assumption that the reader understands the basic principles of packet-switching networks, and is familiar with the CCITT's X25 recommendation.

This protocol was developed to comply with the 1988 recommendations of the CCITT's X25 protocol (Section VIII.2). It is based on parts of the LAPB procedure for managing the data link level and the physical layer.

This protocol allows several virtual call or circuits (called VCs) to take place simultaneously, multiplexed over a single link on the MCX card. The MCX card can handle several X25 data links.

As concerns the CCITT's definition of the DTE/DCE interface (Data Terminal Equipment/Data Circuit-terminating Equipment) at the data packet level, an MCX-25 data link behaves like a DTE.

The figure below illustrates a virtual circuit between an MCX data link (channel 1, for example) and 3 DTEs labeled DTE 1, DTE 2, DTE 3:



Logical channels are numbered from 0 to 4,095 (12 bits)

The DTE MCX channel 1 in this example implements 3 virtual circuits allowing it to communicate simultaneously with 3 DTEs:

- virtual circuit Nb.1 (LC 1 on the MCX side and LC 2 on the DTE 1 side)
- virtual circuit Nb.2 (LC 2 on the MCX side and LC 1 on the DTE 2 side)
- virtual circuit Nb.3 (LC 3 on the MCX side and LC 3 on the DTE 3 side)

The logical channels are assigned during the call set-up phase.

Characteristics of an MCX X25 data link

Address format used: NON TOA/NPI (see CCITT recommendation X121)

Modulo 8 numbering for data packets

Packet window set to 2

Programmable size of user data in packets: 128, 256, 512, 1024 or 2048 bytes

128 VCs per data link for 128- or 256-byte packets

64 VCs per data link for 512-byte packets

32 VCs per data link for 1024-byte packets

16 VCs per data link for 2048-byte packets

Types of packets recognized by the card on a switched virtual circuit:

- * Incoming call
- * Communication established
- * Indication of clear
- * Confirmation of clear by the DCE
- * Data from the DCE
- * RR from the DCE
- * RNR from the DCE
- * Indication of reset
- * Confirmation of reset by the DCE
- * Indication of restart
- * Confirmation of restart
- * Interruption
- * Confirmation of interruption
- * Diagnostic

Packets always contain at least 3 bytes; their general format is as follows:

8	7	6	5	4	3	2	1
Q	D	0	1	Logical Channel Group number (LCG)			
GFI							
Logical Channel number (LC)							
Packet type identifier							
Additional information							

General Format Identifier (GFI) :**Q bit: Data Qualifier bit**

Use of the Q bit is optional in the X25 recommendation. Bit 8, called the A bit, in call and disconnect packets is set to 0 to indicate that the address format used is of type NON TOA/NPI.

D bit: resumption confirmation bit

The procedure using the D bit is not implemented. It is always set to 0 in all packets sent by the MCX card.

Bits 6 and 5 are always coded as indicated above. The code used indicates that the transmit and receive sequence numbers are sent in modulo 8.

Logical Channel Group number (0 to 15)

The logical channel group number appears in all packets except for restart packets. This number has local meaning in the DTE/DCE interface.

Logical channel number (0 to 255)

The number of the logical channel group appears in all packets except for restart packets, where it is forced to 0. This number has a local meaning in the DTE/DCE interface.

All the X25 packets are encapsulated in a data block of an LAPB information frame.

The MCX cards uses a standard window size of 2 for transmissions in both directions for each virtual circuit. Two receive buffers and two transmit buffers, accessible by the RDBUF and BTRAN commands, allow the MCX card to exchange data packets with the PC for each initialized logical channel. These buffers are in addition to the K buffers used in LAPB, which are shared by all logical channels.

THIS PAGE INTENTIONALLY LEFT BLANK

II. Dialog Procedure between the Card and the PC

Communications between the card and the host computer in which it is installed use a 32Kbyte dual-ported memory area. The address of this area is programmable (see the "Installation and Technical Specifications Manual for the MCX Range").

The table on the next page describes the use of this memory, also called the mailbox.

The first column in the table contains the zone's offset from the letter-box's address in the PC. The second column indicates the type of operation that can be performed at that address:

WO	Write only
RO	Read only
RW	Read / Write

The third column contains the corresponding mnemonic.

II.1. Subdivision of the mailbox

Dec / Hex	Perm	Name	Field description	
000 / 000h	RW	VALID	Write command validation byte	VALIDATION ZONE
001 / 001h	RW	ENDIT	Write, acknowledge end of command	ENDIT ZONE
002 / 002h	WO	OPCODE	Write command number	OPCODE ZONE
003 / 003h	RO	STATUS	End of command status "Event" interrupt indicator	STATUS ZONE
004 / 004h	RW	PAR. 1	Parameter #1	PARAMETER ZONE
005 / 005h	RW	PAR. 2	Parameter #2	
006 / 006h	RW	PAR. 3	Parameter #3	
007 / 007h	RW	PAR. 4	Parameter #4	
079 / 4Fh	RW	PAR. 76	Parameter #76	
080 / 50h	RO	I.CHAN	"Event" interrupt: Channel # End-of-command interrupt: Channel #	INTERRUPT ZONE
081 / 51h	RO	I.COND	"Event" interrupt: Type End-of-command interrupt: FFh	
082 / 52h	RO	I.PAR1	"Event" interrupt: Status #1 End-of-command interrupt: opcode	
083 / 53h	RO	I.PAR2	"Event" interrupt: Status #2 End-of-command interrupt: Status #1	
090 / 5Ah	RO	I.PAR9	"Event" interrupt: Status #9 End-of-command interrupt: Status #8	
091 / 5Bh			Reserved	RESERVED ZONE
099 / 63h			Reserved	ZONE
100 / 64h	RW	DATA 1	Data zone, 1 st byte	DATA ZONE
101 / 65h	RW	DATA 2	Data zone, 2 nd byte	
102 / 66h	RW	DATA 3	Data zone, 3 rd byte	
32510/7EFEh	RW	DATA 32411	Data zone, 32411 th byte	ZONE
32511/7EFFh	RW	DATA 32412	Data zone, 32412 nd byte	
32512/7F00h	RO	RX Cnt 1	Receive counter channel 1	RX COUNTER ZONE
32514/7F02h	RO	RX Cnt 2	Receive counter channel 2	
32638/7F7Eh	RO	RX Cnt 64	Receive counter channel 64	TX COUNTER ZONE
32640/7F80h	RO	TX Cnt 1	Transmit counter channel 1	
32642/7F82h	RO	TX Cnt 2	Transmit counter channel 2	
32766/7FFEh	RO	TX Cnt 64	Transmit counter channel 64	

Meaning of the "RX COUNTERS" and "TX COUNTERS" zones

Each "RX Cnt j" or "TX Cnt j" counter is two bytes long, divided into least significant bytes and most significant bytes. The counters associated to the installed channels are updated in real time according to the transmit or receive operations. The other counters are insignificant and constant.

Special case for an ASYNCHRONOUS channel

The "RX Cnt j" counter directly tells the programmer the number of characters received on channel j at time t. This counter is incremented by 1 for each character received. The character is stored in an internal receive buffer on the card (except for a character received during an interrupt). This counter is decreased by the number of characters requested at each non-destructive read request (see the RDBUF command).

The "TX Cnt j" counter directly tells the programmer the usage rate of the transmit buffer associated with channel j. Each time a write request is made (see the BTRAN command) this counter is decreased by the size of the block of characters to be sent. It is increased by 1 each time a character is sent on the line. Once the entire block of characters has been sent, the counter returns to its maximum value, which is the size of the transmit buffer.

You may use the ALLOC and DALOC commands to reconfigure the size allocated at power-on for the receive buffers (512 bytes) and the transmit buffer (8192 bytes) for each asynchronous channel.

Special case for an HDLC or BISYNC channel

The "RX Cnt j" counter indirectly tells the programmer the number of frames received on channel j at time t. This counter is incremented by 1024 for each frame received. The received frames are stored in an internal receive buffer on the card. They fill up the entire buffer, whatever its size. The counter is decreased by 1024 at each destructive read request (see the RDBUF command).

The "TX Cnt j" counter indirectly tells the programmer the usage rate of the transmit buffers associated with channel j. Each time a write request is made (see the BTRAN command), a 1024-byte transmit buffer is occupied, whatever the size of the frame to be sent and the counter is decreased by 1024. Once the entire frame has been sent, the counter is incremented by 1024 and the associated buffer is freed. This counter returns to its maximum value (the number of allocated receive buffers x 1024) once all the buffers have been freed.

The programmer cannot reconfigure the size or the number of receive and transmit buffers. They are allocated by the card when the PROTO command is executed.

Special case for a LAPB channel

The "RX Cnt j" buffer indirectly tells the programmer the number of information frames received or the number of receive buffers occupied by channel j at time t. This counter is incremented by 1024 for each information frame received. These frames are stored in an internal receive buffer on the card, and each frame occupies the entire buffer, whatever the size of the frame. This counter is decreased by 1024 at each destructive read request (see the RDBUF command).

The "TX Cnt j" counter indirectly tells the programmer the usage rate of the transmit buffers associated with channel j. Each time a write request is made (see the BTRAN command), a 1024-byte transmit buffer is occupied, whatever the size of the information frame to send. The transmit counter is then decreased by 1024. Once the transmitted frame is acknowledged, this counter is incremented by 1024 and the associated buffer is freed. This counter returns to its maximum value (the width of the look-ahead window*1024) once all the buffers have been freed.

The card allocates K transmit and receive buffers when the PROTO command is executed, where K is the width of the look-ahead window.

Special case for an X25 channel

The "Rx Cnt j" and "Tx Cnt j" counters are not meaningful in X25. However, they provide information on the occupation rate of the buffers at the LAPB software layer. Specific counters for the buffer occupation rate for each logical channel are located in the data zone (See the chapter entitled "The Command Interpreter in X25 mode" section VI.1).

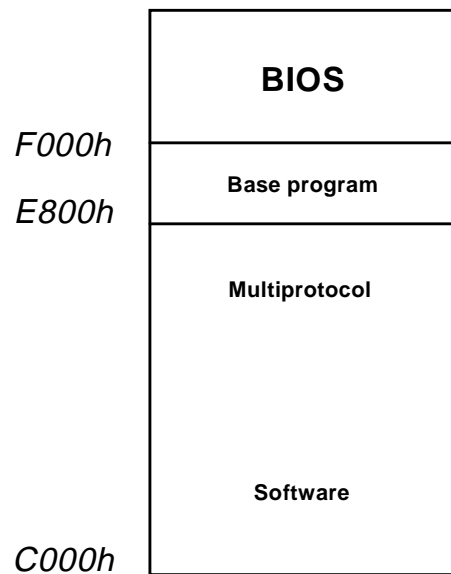
II.2. Setting up the program

Before using the multiprotocol program, make sure the jumpers are set as indicated below:



(Flash) EPROM after installing the multiprotocol program

256 Kbyte (Flash) Eprom



After the card's self-test phase, the LED display counts up to eight bits then starts again; address 0 of the mailbox contains 0Fh, and the next character string is written at address 100 (64h):

MCX IS READY

Now you must send the card the following code to start the multiprotocol software:

RUN 02

This uppercase character string must be written in the data zone of the dual-ported memory (as of address 100 (64h)).

In reply to this startup sequence, the card resets to zero the byte at address 0 in the mailbox (it equaled 0Fh before writing this sequence).

Furthermore, the LEDs stop counting and LED0 starts to blink once a second.

The card is now ready to receive and execute commands.

II.3. Sending commands to the card

Use the following procedure to send a command to the card:

- Write the command code in the OPCODE zone (Address 2)
- Write the required parameters in the PARAMETER zone (Addresses 4 to 4Fh)
- Write the required data in the DATA zone (Addresses 64h to 7EFFh). Warning: if one or more of the card's channels are programmed in X25 mode, the size of the DATA zone is reduced due to the space occupied by the transmit and receive counters associated with each logical channel.
- Write the validation byte (01h) in the VALIDATION zone (Address 0).

An internal mechanism then lets the card execute the command.

Once the command has been executed, the card generates an interrupt on the selected line (see the "Installation Manual and Technical Specifications for the MCX Range") after having updated the following zones:

- The STATUS zone contains an end-of-command interrupt status that can be identified by bit D6 set to 0, an error indicator in bit D7, and the error code sent by the command in bits D5-D4-D3-D2-D1-D0. Bit D7 indicates how the command is executed. If the command returned an error, bit D7 is set to 1 and an error code (different from 0) is written to bits D5-D4-D3-D2-D1-D0. If no error was returned, bit D7 is set to 0 and an error code of 0 is written to bits D5-D4-D3-D2-D1-D0: the status byte is null.
- The DATA zone contains the data sent by the command, if appropriate.
- The VALIDATION byte is reset to 0, indicating that a new command may be sent.
- Bit D7 in the ENDIT zone is set to 1.
- Addresses 50h, 51h... in the INTERRUPT ZONE are documented in section II.4.1.

In any case, you must acknowledge the end-of-command interrupt using the procedure described in section II.4.3.

II.3.1. Error code management

The card returns an error code for each command. This code, also called the return code or the end-of-command interrupt status, is written in the STATUS zone in the mailbox to indicate if the command was correctly executed, if there was an error in one of the parameters, or if an error occurred during command execution.

In most cases, the error codes indicate an error in the command parameters.

Furthermore, some commands return errors related to the channel status at a given moment:

- BTRAN (not enough room to buffer the data, channel disconnected)
- RDBUF (data not available)
- PRCTL (invalid logical state for the channel)
- ...

A complete list of the possible causes for each error is included in the description for each command.

Appendix VII.2 at the end of the manual provides a summary of all the error codes.

Error code 188 (BCh)

Error code 188 is returned by certain commands associated with channels programmed in the LAPB or X25 protocol. It indicates a protocol error that may be identified in the interrupt zone that is updated by the end-of-command interrupt (see section II.4.1).

Error code 189 (BDh)

This code indicates the use of an unauthorized command.

Error code 190 (BEh)

This code indicates the use of an unknown command.

Error code 191 (BFh)

This code indicates the use of a command that is not implemented.

II.4. Interrupt management

As discussed in the previous section, the card generates interrupts on the machine bus.

Two factors can cause an interrupt:

The first concerns the **end-of-command interrupt**.

The second type of interrupt is more complex. The interrupt generated depends on the conditions selected by the MINTR command . These are called "**event interrupts**"

II.4.1. End-of-command interrupts

For these interrupts, bit 6 of the status byte is set to zero.

The card writes additional information in the interrupt zone of the dual-ported memory (respectively, addresses 80 to 90 / 50h to 5Ah).

The PC must absolutely reply to an end-of-command interrupt by sending the ENDIT command (see section II.4.3), in order to enable the card to send another interrupt.

The information that the card writes to the interrupt zone depends on the protocol used by the channel.

ASYNCHRONOUS, HDLC and BISYNC Protocols

Interrupt Zone	Description
Address 80(50h)	Significant if the command has a “channel” parameter. Channel # for the command.
Address 81(51h)	End-of-command interrupt: FFh
Address 82(52h)	Command code (copy of the OPCODE zone)
Addresses 83(53h) to 90(5Ah) are not significant.	

LAPB Protocol

Interrupt Zone	Description
Address 80(50h)	Significant if the command has a “channel” parameter. Channel # for the command.
Address 81(51h)	End-of-command interrupt: FFh
Address 82(52h)	Command code (copy of the OPCODE zone)
Address 83(53h)	Significant if the STATUS zone=188: LAPB protocol error code
Addresses 84(54h) to 90(5Ah) are not significant.	

X25 Protocol

Interrupt Zone	Description
Address 80(50h)	Significant if the command has a “channel” parameter. Channel # for the command.
Address 81(51h)	End-of-command interrupt: FFh
Address 82(52h)	Command code (copy of the OPCODE zone)
Address 83(53h)	Significant if the STATUS zone=188 Error type: 0 if the X25 error is associated with a code from the X25 standard. The error code is provided in byte 87. 255 if the error is internal to the card. Other codes: specific to certain commands.
Address 84(54h)	Significant if the command has an LC² parameter Number of the command's Logical Channel
Address 85(55h)	Significant if the command has an LCG parameter Number of the command's Logical Channel Group
Address 86(56h)	Not significant
Address 87(57h)	Significant if the STATUS zone=188 and if address 83=0 Error code in the X25 standard (see Appendix E in the X25 recommendation)
Address 88(58h) and Address 89(59h)	Not significant
Address 90(5Ah)	Significant if the command has LCG and LC parameters that are not null Logical Channel Index (LCI)

² The PRCTL + X25CALL command updates the LC, LCG and LCI during the end-of-command interrupt.

II.4.2. "Event" interrupts

For these interrupts, bit 6 in the status byte is set to 1.

The card writes additional information in the interrupt zone of the dual-ported memory (respectively, addresses 80 to 90 / 50h to 5Ah).

Each condition may be set individually by the MINTR command. The interrupts may be triggered by data received, by emptying the transmit buffer, or by special events. Their interpretation varies according to the protocol selected with the PROTO command.

Several conditions may be validated for a single channel, and the choice may differ for each communication line. The card sends "event" interrupts in chronological order. An "event" interrupt only indicates one event at a time. If several events occur at the same time, the card generates an interrupt for each event.

Like for an end-of-command interrupt, the PC must absolutely reply to an end-of-command interrupt by sending the ENDIT command (see section II.4.3) in order to enable the card to send another interrupt.

The following tables describe the format of the interrupt zone for each protocol.

ASYNCHRONOUS PROTOCOL

General format for INTERRUPT zones :

Address	Description
80 (50h)	Channel number associated with the event
81 (51h)	Type of interrupt event (see below)
82 (52h)	Least significant byte of the number of characters contained in the receive buffer (except for IT7)
83 (53h)	Most significant byte of the number of characters contained in the receive buffer (except for IT7)
84 (54h)	Detailed diagnostic of the event (see below)

Addresses 85 (55h) to 90 (5Ah) are not significant.

Characteristics of various events:

TYPE 01H: INTERRUPT TRIGGERED FOR EACH CHARACTER RECEIVED

MINTR parameter: **IT0**

Address 84: Character received.

Comments: the character received is not placed in the receive buffer, and therefore the received character counter (RX counters zone) is not incremented.

TYPE 02H: INTERRUPT TRIGGERED WHEN THE NUMBER OF CHARACTERS RECEIVED IS MODULO THE NUMBER OF CHARACTERS PROGRAMMED BY THE STCNT COMMAND, OR INTERRUPT FOR FIRST CHARACTER RECEIVED.

MINTR parameter: **IT1** (see STCNT command)

Address 84: 0 (zero).

Comments:

if STCNT > 1 then this interrupt means: the interrupt was triggered when the number of characters contained in the receive buffer reached an exact multiple of the number of characters programmed by the STCNT command, if STCNT = 1 then this interrupt means: the interrupt was triggered when the first character was received. This interrupt only occurs once, until the MINTR command is executed again.

TYPE 03H: INTERRUPT TRIGGERED WHEN THE RECEIVE BUFFER GOES FROM EMPTY TO NOT EMPTY.

MINTR parameter: **IT2, Mde bit** = 1 (see the MINTR command)

Address 84: 0 (zero).

Comments: the number of characters contained in the receive buffer and written to addresses 82/52h and 83/53h equals 1.

TYPE 04H: INTERRUPT TRIGGERED WHEN THE RECEIVE BUFFER BECOMES FULL.

MINTR parameter: **IT2, Mde bit** = 0 (see the MINTR command)

Address 84: 0 (zero).

Comments: the number of characters contained in the receive buffer and written to addresses 82/52h and 83/53h equals the size allocated for the receive buffer (default equals 512 bytes).

TYPE 08H: INTERRUPT TRIGGERED WHEN A CHARACTER STRING PROGRAMMED BY THE CHDEF COMMAND IS RECOGNIZED.

MINTR parameter: **IT3**

Address 84: Contains the character that was recognized if the comparison is performed on a single character, or the second character of the string if the comparison is performed on two characters³.

Comments: addresses 82 and 83 contain the number of characters contained in the receive buffer, including the string.

TYPE 10H: INTERRUPT TRIGGERED BY A "TIMEOUT" (SEE STTMO COMMAND).

MINTR parameter: **IT4**

Address 84: 0 (zero).

TYPE 20H: INTERRUPT TRIGGERED WHEN A RECEIVE ERROR WAS DETECTED.

MINTR parameter: **IT5**

Address 84: Type of error encountered (0FRP0000)

- If P = 1 Parity error
- If R = 1 Characters lost: the cause of the error is indicated in bit S
 - if R = 1 and S = 0: Characters overwritten in the USART
 - if R = 1 and S = 1: Receive buffer overrun
- If F = 1 frame error.

Comments: If a receive error occurs, the erroneous character is written in the receive buffer after this interrupt is generated (unless the receive buffer is full, or if a Type 01h interrupt was programmed).

³ This function is useful if you are using the CHDEF command to program characters containing an Escape "(1Bh, 27)" in the first position and another character in the second position (for example: keys for moving the cursor). It allows you to read the second character directly, which indicates the cursor key pressed.

TYPE 40H: INTERRUPT TRIGGERED WHEN CTS, CD, RI OR BREAK CHANGES STATE.

MINTR parameter: **IT6**

Address 84: Signal status: (B0CRD000)

- If B = 1 then a break has been detected on the line
- If C = 1 then CTS is high; if C = 0 then CTS is low
- If R = 1 then RI is high; if R = 0 then RI is low
- If D = 1 then CD is high; if D = 0 then CD is low

Comments: a BREAK triggers 2 interrupts. The first is sent when the BREAK is detected: bit (B) is high (1). The second interrupt is sent when the end of BREAK is detected: bit (B) is then low (0).

TYPE 80H: INTERRUPT TRIGGERED WHEN THE TRANSMIT BUFFER IS TOTALLY EMPTY

MINTR parameter: **IT7**

Address 84: 0 (zero)

Comments: exceptionally, addresses 82/52h and 83/53h contain 0 (zero).

HDLC AND BISYNC ENVELOPE LEVEL PROTOCOLS

General format for the INTERRUPT zone :

Address	Description
80 (50h)	Channel number associated with the event
81 (51h)	Type of interrupt (see below)
82 (52h)	Number of frames received (except for IT1 and IT7)
83 (53h)	0 (zero) (except for IT1)
84 (54h)	Detailed diagnostic of the event (see below)

Addresses 85 (55h) to 90 (5Ah) are not significant.

Characteristics of the various events:

TYPE 01H: NOT DEFINED.

MINTR parameter: **IT0**

Comments: if this interrupt is selected, it never occurs.

TYPE 02H: INTERRUPT TRIGGERED FOR EACH FRAME RECEIVED.

MINTR parameter: **IT1**

Address 84: 0 (zero).

Comments: exceptionally,

address 82 contains the least significant byte of the number of characters in the frame

address 83 contains in bits 0, 1, 2 and 3 the four most significant bits of the number of characters in the frame.

address 83 contains in bits 4, 5, 6, 7 a receive frame error code:

bit 7	bit 6	bit 5	bit 4	description
0	0	0	0	No error in the frame
1	0	0	0	Frame length greater than size allocated
0	1	0	0	Bad CRC in HDLC
0	0	1	0	Character overwritten in the USART
0	0	0	0	HDLC frame terminated by an ABORT sequence

TYPE 03H: INTERRUPT TRIGGERED WHEN THE FIRST FRAME IS RECEIVED.

MINTR parameter: **IT2, Mde bit** = 1 (see the MINTR command)
Address 84: 0 (zero).

TYPE 04H: INTERRUPT TRIGGERED WHEN THE 14 RECEIVE BUFFERS BECOME FULL.

MINTR parameter: **IT2, Mde bit** = 0 (see the MINTR command)
Address 84: 0 (zero).

TYPE 10H: INTERRUPT TRIGGERED BY A "TIMEOUT" (SEE THE STTMO COMMAND).

MINTR parameter: **IT4**
Address 84: 0 (zero).

TYPE 20H: INTERRUPT TRIGGERED WHEN A RECEIVE ERROR IS DETECTED.

MINTR parameter: **IT5**
Address 84: Type of error encountered (SCRAB000)

- If S = 1 Frame length greater than size allocated
- If C = 1 CRC error
- If R = 1 Characters overwritten in the USART
- If A = 1 HDLC frame terminated by an ABORT sequence
- If B = 1 Frame lost.

Comments: even when a receive error occurs, the erroneous frame is written in the receive buffer (unless the frame is lost).

TYPE 40H: INTERRUPT TRIGGERED WHEN CTS, CD, RI CHANGES STATE, OR IF AN ABORT SEQUENCE IS DETECTED IN HDLC.

MINTR parameter: **IT6**
Address 84: Signal status: (A0CRD000)

- If A = 1 then an ABORT was detected on the HDLC line
- If C = 1 then CTS is high; if C = 0 then CTS is low
- If R = 1 then RI is high; if R = 0 then RI is low
- If D = 1 then CD is high; if D = 0 then CD is low

Comments: the ABORT sequence triggers 2 interrupts. The first is sent as soon as the ABORT is detected: bit (A) is then high (1). The second interrupt is sent when the end of ABORT is detected: bit (A) is then low (0).

TYPE 80H: INTERRUPT TRIGGERED WHEN THERE ARE NO MORE FRAMES TO TRANSMIT.

MINTR parameter: **IT7**
Address 84: 0 (zero)
Comments: exceptionally, address 83/53h contains 0 (zero).

LAPB OR HDLC-ABM LINK-LEVEL PROTOCOL

General format of the INTERRUPT zone

Address	Description
80 (50h)	Channel number associated with the event
81 (51h)	Type of interrupt (see below)
82 (52h)	Size of the receive window (except for IT1 and IT7)
83 (53h)	0 (zero) (except for IT1)
84 (54h)	Detailed diagnostic of the event (see below)
85 (55h)	Detailed diagnostic of the IT6 event (not significant for other events)

Addresses 86 (56h) to 90 (5Ah) are not significant.

Characteristics of the various events:

TYPE 01H: NOT DEFINED.

MINTR parameter: **IT0**

Comments: if this interrupt is selected, it never occurs.

TYPE 02H: INTERRUPT TRIGGERED FOR EACH INFORMATION FRAME RECEIVED.

MINTR parameter: **IT1**

Address 84: 0 (zero).

Comments: exceptionally,

address 82 contains the least significant byte of the number of characters contained in the information field of the frame's information block
address 83 contains the most significant byte of the number of characters contained in the information field of the frame's information block

TYPE 03H: INTERRUPT TRIGGERED WHEN THE RECEIVE WINDOW CHANGES FROM STATE 0 TO STATE 1

MINTR parameter: **IT2, Mde bit = 1** (see the MINTR command)

Address 84: 0 (zero).

TYPE 04H: INTERRUPT TRIGGERED WHEN THE RECEIVE BUFFERS BECOME FULL

MINTR parameter: **IT2, Mde bit = 0** (see the MINTR command)

Address 84: 0 (zero).

TYPE 10H: INTERRUPT TRIGGERED BY A "TIMEOUT" (SEE THE STTMO COMMAND).

MINTR parameter: **IT4**
Address 84: 0 (zero).

TYPE 20H: INTERRUPT TRIGGERED WHEN A RECEIVE ERROR IS DETECTED.

MINTR parameter: **IT5**
Address 84: Type of error encountered (SCRAB000)

- If S = 1 Frame length greater than N1 bytes (including CRC)
- If C = 1 CRC error
- If R = 1 Characters overwritten in the USART
- If A = 1 HDLC frame terminated by an ABORT sequence
- If B = 1 Frame lost.

Comments: When a receive error occurs, the erroneous frame is not written to the receive buffer.

TYPE 40H: INTERRUPT TRIGGERED WHEN CTS, CD, RI CHANGES STATE, OR IF AN ABORT SEQUENCE IS DETECTED IN HDLC.

MINTR parameter: **IT6**
Address 83: 0
Address 84: Signal status: (A0CRD000)

- If A = 1 then an ABORT was detected on the HDLC line
- If C = 1 then CTS is high; if C = 0 then CTS is low
- If R = 1 then RI is high; if R = 0 then RI is low
- If D = 1 then CD is high; if D = 0 then CD is low

Comments: the ABORT sequence triggers 2 interrupts. The first is sent as soon as the ABORT is detected; bit (A) is then high (1). The second interrupt is sent when the end of ABORT is detected; bit (A) is then low (0).

Address 85: 0

TYPE 40H: INTERRUPT TRIGGERED WHEN THE LOGICAL STATE OF THE LAPB LINK CHANGES

MINTR parameter: **IT6**
Address 83 = 0
Address 84: Signal status: (A0C0D000) (see above)
Address 85: LAPB link status

- 128 (80h): Change to *link disconnected* status
- 129 (81h): Change to *link connected* status
- 132 (84h): Reset the LAPB link

Important remark concerning the reset of the LAPB layer:

In the phase where information is transferred (“*link connected*”), the LAPB layer may be reset if a serious error occurs:

1. Receipt of an FRMR frame
2. Receipt of an invalid frame (invalid NR, unknown control field, invalid frame length)
3. Receipt of an SABM frame
4. Receipt of a DM frame

In all cases, the MCX card informs the host of the incident by sending an IT6 event interrupt (with code 132) if the interrupt is authorized (see the MINTR command).

In cases 1, 3 and 4, the MCX card attempts to make an automatic reconnection by sending an SABM frame:

- If successful, an IT6 interrupt event (with code 129) is sent.
- If not, an IT6 interrupt (with code 128) is sent.

In case 2, the MCX card waits for an SABM frame (waiting for the part of the device connected to the card to be reset).

In all cases, the reset causes all the information in the transmit and receive buffers that has not yet been processed to be lost.

TYPE 80H: INTERRUPT TRIGGERED WHEN ALL THE TRANSMISSION WINDOW IS EMPTY.

MINTR parameter: **IT7**

Address 84: 0 (zero)

Comments: exceptionally, address 83/53h contains 0 (zero).

X25 PROTOCOL

TYPE 03H: SWITCHING THE RECEIVE WINDOW FROM 0 TO 1
MINTR parameter: **IT2, Mde bit = 1** (see the MINTR command)

INTERRUPT ZONE	DESCRIPTION
80 (50h)	Number of the channel associated with the event
81 (51h)	Type of interrupt event: 03h
82 (52h) and 83 (53h)	Reserved
84 (54h)	Logical Channel number
85 (55h)	Logical Channel Group number
86 (56h) to 89 (59h)	Reserved
90 (5Ah)	Logical Channel Index

TYPE 40H: X25 EVENTS
MINTR Parameter: IT6

INTERRUPT ZONE	DESCRIPTION	
80 (50h)	Number of the channel associated with the event	
81 (51h)	Type of interrupt event: 40h	
82 (52h)	Reserved	
83 (53h)	<p>X25 event code:</p> <p>3: outgoing call confirmed by the DCE 4: unconfirmed time-out on outgoing call</p> <p>9: clear requested by the DCE 5: outgoing clear confirmed by the DCE 6: unconfirmed time-out on outgoing clear</p> <p>7: diagnostic packet received 8: protocol error⁴</p> <p>10: reset requested by the DCE 11: outgoing reset confirmed by the DCE 12: unconfirmed timeout on outgoing reset</p> <p>13: follow-on requested by the DCE 14: outgoing follow-on confirmed by the DCE 15: unconfirmed time-out on the outgoing follow-on</p> <p>16: interruption by the DCE 17: interruption confirmed</p> <p>128: data link out of service 129: data link in service 132: data link is reinitializing</p>	<p>X25 event code:</p> <p>1: received incoming call 2: <i>time-out listen</i>⁵</p>
84 (54h)	Logical Channel number ⁶	
85 (55h)	Logical Channel Group number	
86 (56h)	X25 cause code ⁷	first byte of the wildcard's identification
87 (57h)	X25 diagnostic code ⁸	second byte of the wildcard's identification
88 (58h)	reserved	third byte of the wildcard's identification
89 (59h)	reserved	fourth byte of the wildcard's identification
90 (5Ah)	Logical Channel Index ⁹	

⁴ The "Protocol error" event code can only occur if the DCE malfunctions.

⁵ The "timeout listen" event has not been implemented to date.

⁶ The VL and GVL data is only significant for the following X25 event codes:
 2,7,8,13,14,15,128,129

⁷ The X25 cause code is significant for events 8, 9 and 13

⁸ The X25 diagnostic code is significant for events 8, 9 and 13

⁹ The Logical Channel Index is not significant if it is associated with logical channel 0.

TYPE 80H: THE TRANSMISSION WINDOW IS EMPTY
MINTR Parameter: **IT7**

ADDRESS	DESCRIPTION
80 (50h)	Channel number associated with the event
81 (51h)	Interrupt event type: 80h
82 (52h) and 83 (53h)	Reserved
84 (54h)	Logical Channel number
85 (55h)	Logical Channel Group number
86 (56h) à 89 (59h)	Reserved
90 (5Ah)	Logical Channel Index ¹⁰

¹⁰ The Logical Channel Index is not significant if it is associated with logical channel 0.

II.4.3. Acknowledging interrupts

This operation takes place by writing the byte 01h to address 1 in the mailbox (ENDIT Zone). It authorizes the card to modify the interrupt zone and to send another interrupt to the PC.

This operation must be performed both for acknowledging "event" interrupts and end-of-command interrupts.

II.4.4. The interrupt queue

"Event" interrupts as well as end-of-command interrupts are stored in a queue which may contain up to 1120 entries.

If the machine cannot handle these interrupts fast enough, the queue fills up. If an overflow occurs, the card signals the incident by an "event" interrupt with code 0FFh as a parameter at address 80 (50h).

When the queue is totally full, the card waits for as many "ENDIT" sequences as there are events in the queue.

II.5. Using the card in "polling" mode

In this mode, the PC ignores interrupts sent by the card.

To detect end of command and event interrupts, a status bit is read. This is bit 7 in the ENDIT zone, and it is called the STATRDY bit.

In this mode, commands are sent using the same procedure as described in II.3

The interrupt triggered by the card can be detected because the STATRDY bit goes to 1. This bit should be tested as often as possible to avoid filling up the interrupt queue. The type of interrupt may then be identified by reading the STATUS zone, and the associated information (DATA and INTERRUPT zones) should be processed.

The ENDIT command should finally be sent using the procedure described in section II.4.3, in order to enable the card to send the next interrupt.

Important note:

Because of the design of the hardware, the use of the card in "polling" mode is not recommended. Overly systematic "polling" of the dual-ported memory can slow down the card to some extent and consequently reduce performance.

Managing the card with interrupts provides much better results than using "polling" mode.

THIS PAGE INTENTIONALLY LEFT BLANK

III. Initializing Channels in Software

After the start code "RUN 02", **all the card's channels are initialized in asynchronous mode** with the following default parameters:

- RS232D mode (see the RSMDE command)
- Size of the transmit buffer: 8192 bytes (see the ALLOC command)
- Size of the receive buffer: 512 bytes (see the ALLOC command)
- Timeout waiting for a character: 10 seconds (see the STTMO command)
- End-of-character block character: 0Dh (see the CHDEF command)
- Size of a character block: 80 bytes (see the STCNT command)
- No flow control (see the VMODE command)
- No filtering of characters received (see the VMODE command)
- Reception of characters not authorized (see the RXENB command)
- No programmed interrupt events (see the MINTR command)

Warning: at this stage, the channels are not operational, i.e. you cannot send or receive characters.

To make an asynchronous channel operational, simply send the following commands:

RSMDE, VINIT, RXENB

To make an HDLC or BISYNC frame envelope-level channel operational, simply send the following commands:

PROTO, RSDME, VINIT, RXENB

To make an LAPB (HDLC ABM) channel operational, simply send the following commands:

PROTO, RSMDE, VINIT, RXENB, PRCTL (LAPLNKUP)

To make an X25 channel operational, simply send the following commands:

PROTO, RSMDE, VINIT, RXENB, PRCTL (LAPLNKUP), PRCTL (X25LISTN or X25CALL)

In all the above cases, the other commands are optional, and depend on your application.

Here follows a sample initialization sequence:

- PROTO:** Initializes the protocol: (asynchronous, HDLC, BISYNC, LAPB, X25).
- RSMDE:** Selects the electrical interface: RS232D, RS422A, etc.
- MINTR:** Initializes the interrupt mode.
- STSIG:** Sets the DTR and RTS signals.
- VINIT:** Initializes the communication channel.
- VMODE:** Configures flow control in ASYNCHRONOUS mode.
- RXENB:** Enables reception of characters.
- PRCTL:** Protocol-specific commands.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. Interpreting the Group of 8 LEDs

After the power on self test phase described in the "Installation manual and Technical Specifications of the Cards in the MCX Range", the group of 8 LEDs lights up in rapid succession.

If one or more LEDs remain permanently lit, an error has occurred. You should turn off the machine and contact ACKSYS technical support.

As soon as the start code is sent («RUN 02»), the card is operational, and LED 0 turns on and off once per second.

The eight LEDs then have the following meaning:

- "LED" 0 Active (turns on and off once per second)
- "LED" 1 Command being processed
- "LED" 2 Interrupt towards the PC is active
- "LED" 3 Receiving in asynchronous mode
- "LED" 4 Transmitting in asynchronous mode
- "LED" 5 Receiving in synchronous mode
- "LED" 6 Transmitting in asynchronous mode
- "LED" 7 Receive error in asynchronous or synchronous mode (red LED)

THIS PAGE INTENTIONALLY LEFT BLANK

V. The Command Interpreter

V.1. PROTO (2Fh): Initialize the protocol used by a channel

OPCODE = 47 (2Fh)

DESCRIPTION

This command initializes a channel in one of the protocols listed below. After this command, you must send the following commands in order to initialize at the bit level (initialization of the communication controller):

VINIT
RXENB

For an ASYNCHRONOUS channel

The PROTO command creates an 8192-byte transmit buffer and a 512-byte receive buffer.

For a synchronous HDLC or BISYNC frame envelope channel

The PROTO command creates four transmit buffers and fourteen 1024-byte receive buffers. An additional buffer stores the frame currently being received. Its contents will be lost if none of the fourteen buffers is free when the frame has been received.

Warning: frames sent or received fill up an entire buffer, whatever the size of the frame.

For a synchronous LAPB or HDLC ABM channel

The PROTO command creates K transmit buffers and K receive buffers used to send and receive information frames.

The K parameter (Width of the LAPB window) and the maximum size of the data information fields in an information frame are defined in the PROTO command.

Warning: frames sent or received fill up an entire buffer, whatever the size of the frame.

For an X25 data link

The PROTO command initializes an X25 link with a window of 2 packets with at most 128 Virtual Circuits that support 128- or 256-byte packets. Some parameters, indicated in italics, are not used in the current revisions of the MCX MULTIPROTOCOL software; nevertheless they must be initialized as specified in the documentation.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	1	0	1	1	1	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	Cod	Cod	Cod	Cod	Cod	Cod	Cod	Cod
PARAMETER 3	0	0	0	0	Prt	Prt	Prt	Prt
PARAMETER 4	0	0	0	0	0	Plex	Plex	Plex
PARAMETER 5	0	0	0	0	0	0	TXclk	TXclk
PARAMETER 6	0	0	0	0	0	0	RXclk	RXclk
PARAMETER 7	0	0	0	0	0	0	TRxC	TRxC

The following parameters are related to the protocol.
They are described in separate tables for each protocol.

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 specify the channel number. This must be a number between 1 and the maximum number of channels installed.

Cod: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

All the bits in parameter #2 must be set to 1. The value 0 ensures compatibility with previous versions of the PROTO command.

Prt: PARAMETER #3 DB3.DB2.DB1.DB0

These bits in parameter #3 indicate the protocol associated with the channel as per the following table:

DB3	DB2	DB1	DB0	Protocol
0	0	0	0	ASYNCHRONOUS mode
0	0	1	0	BISYNC mode (Frame envelope)
0	1	0	0	HDLC mode (Frame envelope)
0	1	0	1	LAPB Protocol or HDLC ABM
0	1	1	0	X25 protocol

Plex: PARAMETER #4 DB2.DB1.DB0

This bit in parameter #4 defines the protocol's operating mode: Half duplex or Full duplex.

DB2	DB1	DB0	Mode	Remarks
0	0	0	Full Duplex	Supported by all protocols
0	0	1	Half Duplex	Not supported in ASYNCHRONOUS mode

TXclk: PARAMETER #5**DB1.DB0**

These bits in parameter #5 define the source of the transmit clock used by the channel.

		SOURCE OF THE TRANSMIT CLOCK	
DB1	DB0	Clock in RS232D	Clock in differential
0	0	RxCLK pin	\pm RxCLK pins
0	1	TxCLKIn pin	Combination prohibited (MCXBP & LiteSerial) \pm TxCLKIn pins (Lite570)
1	0	Supplied by the USART's baud generator depending on the speed programmed by the VINIT command	
1	1	Extracted from data received	

Combination (1,1) is not supported in ASYNCHRONOUS mode.

RXclk: PARAMETER #6**DB1.DB0**

These bits in parameter #6 define the source of the receive clock used by the channel.

		SOURCE OF THE RECEIVE CLOCK	
DB1	DB0	Clock in RS232D	Clock in differential
0	0	RxCLK Pin	\pm RxCLK Pins (MCXBP & LiteSerial) \pm RxCLK Pins (Lite570)
0	1	TxCLKIn Pin (MCXBP & LiteSerial) Combination prohibited (Lite570)	Combination prohibited
1	0	Supplied by the USART's baud generator depending on the speed programmed by the VINIT command ¹¹	
1	1	Extracted from data received ¹²	

Warning:

If the receive clock is generated from data received:

You cannot use the signal generated internally by the VINIT command for the transmit clock.

The RS422A differential clock signals are only available on the first 3 channels of the MCX-BP extension.

The (1,1) combination is not supported in ASYNCHRONOUS mode.

¹¹ Note for the LITE570 extension : in this case, set the channel's ST jumper to position 2-3

¹² Note for the LITE570 extension: in this case, set the channel's ST jumper to position 2-3.

TRxC: PARAMETER #7

DB1.DB0

These bits in parameter #7 define the type of the outgoing signal on the TXCLKOut pin or on the \pm TXCLKOut pins.

DB1	DB0	OUTGOING SIGNAL ON TXCLKOUT
0	0	Signal is high
0	1	Transmit clock
1	0	Clock generated internally according to the speed programmed in the VINIT command
1	1	Clock extracted from data received

Warning:

These bits are ignored if the transmit clock is provided to the communication controller by the TxCLKIn pin or the \pm TxCLKIn pins (see parameter #5).

The (1,1) combination is not supported in ASYNCHRONOUS mode.

Notes:

For standard asynchronous mode (with no clock), parameters #5, #6 and #7 must be set to 2, 2 and 0, respectively.

Special case: BISYNC frame envelope mode

PARAMETER 8	Syncl	Syncl	Syncl	Syncl	Syncl	Syncl	Syncl	Syncl
PARAMETER 9	Synch	Synch	Synch	Synch	Synch	Synch	Synch	Synch
PARAMETER 10	0	0	0	0	0	0	0	0
PARAMETER 11	0	0	0	0	0	0	0	0
PARAMETER 12	0	0	0	0	0	0	0	0
PARAMETER 13	0	0	0	0	0	0	0	0
PARAMETER 14	0	0	0	0	0	0	0	0
PARAMETER 15	LUI	LUI	LUI	LUI	LUI	LUI	LUI	LUI
PARAMETER 16	LUh	LUh	LUh	LUh	LUh	LUh	LUh	LUh
PARAMETER 17	0	0	0	0	0	0	0	0
PARAMETER 18	0	0	0	0	0	0	0	0

Syncl: PARAMETER #8 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 Synch: PARAMETER #9 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

The bits in parameter #8 define the first synch byte used in BISYNC mode.

The bits in parameter #9 define the second synch byte used in BISYNC mode.

PARAMETER #10 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 PARAMETER #11 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 PARAMETER #12 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 PARAMETER #13 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 PARAMETER #14 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameters #10,11,12,13 and 14 must be set to 0.

LUI: PARAMETER #15 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 LUh: PARAMETER #16 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes for the maximum size of data in a frame. The size must be between 2 and 2048 bytes.

A size of 1024 bytes is allocated if these 2 parameters are null.

PARAMETER #17 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 PARAMETER #18 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameters #17 and 18 must be set to 0.

Special case: HDLC mode frame envelope

PARAMETER 8	0	0	0	0	0	0	0	0
PARAMETER 9	0	0	0	0	0	0	0	0
PARAMETER 10	0	0	0	0	0	0	0	0
PARAMETER 11	0	0	0	0	0	0	0	0
PARAMETER 12	0	0	0	0	0	0	0	0
PARAMETER 13	0	0	0	0	0	0	0	0
PARAMETER 14	0	0	0	0	0	0	0	0
PARAMETER 15	LUI	LUI	LUI	LUI	LUI	LUI	LUI	LUI
PARAMETER 16	LUh	LUh	LUh	LUh	LUh	LUh	LUh	LUh
PARAMETER 17	0	0	0	0	0	0	0	0
PARAMETER 18	0	0	0	0	0	0	0	0

PARAMETER #8	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #9	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #10	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #11	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #12	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #13	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #14	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameters #8,9,10,11,12,13 and 14 must be set to 0.

LUI: PARAMETER #15	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
LUh: PARAMETER #16	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the maximum size of data in an HDLC frame (not including the CRC). The size must be between 2 and 2048 bytes.

A size of 1024 bytes is allocated if these first 2 parameters are null.

PARAMETER #17	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #18	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameters #17 and 18 must be set to 0.

Special case: LAPB and X25 protocols

Warning: for LAPB protocol, parameters #19 to #34 and the data zone are not significant.

PARAMETER 8	Adr	Adr	Adr	Adr	Adr	Adr	Adr	Adr
PARAMETER 9	T1l	T1l	T1l	T1l	T1l	T1l	T1l	T1l
PARAMETER 10	T1h	T1h	T1h	T1h	T1h	T1h	T1h	T1h
<i>PARAMETER 11</i>	<i>T2l</i>	<i>T2l</i>	<i>T2l</i>	<i>T2l</i>	<i>T2l</i>	<i>T2l</i>	<i>T2l</i>	<i>T2l</i>
<i>PARAMETER 12</i>	<i>T2h</i>	<i>T2h</i>	<i>T2h</i>	<i>T2h</i>	<i>T2h</i>	<i>T2h</i>	<i>T2h</i>	<i>T2h</i>
<i>PARAMETER 13</i>	<i>T3l</i>	<i>T3l</i>	<i>T3l</i>	<i>T3l</i>	<i>T3l</i>	<i>T3l</i>	<i>T3l</i>	<i>T3l</i>
<i>PARAMETER 14</i>	<i>T3h</i>	<i>T3h</i>	<i>T3h</i>	<i>T3h</i>	<i>T3h</i>	<i>T3h</i>	<i>T3h</i>	<i>T3h</i>
PARAMETER 15	LUI	LUI	LUI	LUI	LUI	LUI	LUI	LUI
PARAMETER 16	LUh	LUh	LUh	LUh	LUh	LUh	LUh	LUh
PARAMETER 17	0	0	N2	N2	N2	N2	N2	N2
PARAMETER 18	K	K	K	K	K	K	K	K
PARAMETER 19	LICl	LICl	LICl	LICl	LICl	LICl	LICl	LICl
PARAMETER 20	LICH	LICH	LICH	LICH	LICH	LICH	LICH	LICH
<i>PARAMETER 21</i>	<i>HICl</i>	<i>HICl</i>	<i>HICl</i>	<i>HICl</i>	<i>HICl</i>	<i>HICl</i>	<i>HICl</i>	<i>HICl</i>
<i>PARAMETER 22</i>	<i>HICH</i>	<i>HICH</i>	<i>HICH</i>	<i>HICH</i>	<i>HICH</i>	<i>HICH</i>	<i>HICH</i>	<i>HICH</i>
PARAMETER 23	LBCl	LBCl	LBCl	LBCl	LBCl	LBCl	LBCl	LBCl
PARAMETER 24	LBCh	LBCh	LBCh	LBCh	LBCh	LBCh	LBCh	LBCh
PARAMETER 25	HBCl	HBCl	HBCl	HBCl	HBCl	HBCl	HBCl	HBCl
PARAMETER 26	HBCh	HBCh	HBCh	HBCh	HBCh	HBCh	HBCh	HBCh
<i>PARAMETER 27</i>	<i>LOCl</i>	<i>LOCl</i>	<i>LOCl</i>	<i>LOCl</i>	<i>LOCl</i>	<i>LOCl</i>	<i>LOCl</i>	<i>LOCl</i>
<i>PARAMETER 28</i>	<i>LOCh</i>	<i>LOCh</i>	<i>LOCh</i>	<i>LOCh</i>	<i>LOCh</i>	<i>LOCh</i>	<i>LOCh</i>	<i>LOCh</i>
<i>PARAMETER 29</i>	<i>HOCl</i>	<i>HOCl</i>	<i>HOCl</i>	<i>HOCl</i>	<i>HOCl</i>	<i>HOCl</i>	<i>HOCl</i>	<i>HOCl</i>
<i>PARAMETER 30</i>	<i>HOCh</i>	<i>HOCh</i>	<i>HOCh</i>	<i>HOCh</i>	<i>HOCh</i>	<i>HOCh</i>	<i>HOCh</i>	<i>HOCh</i>
PARAMETER 31	CNTl	CNTl	CNTl	CNTl	CNTl	CNTl	CNTl	CNTl
PARAMETER 32	CNTTh	CNTTh	CNTTh	CNTTh	CNTTh	CNTTh	CNTTh	CNTTh
PARAMETER 33	0	0	0	0	Adlen	Adlen	Adlen	Adlen
PARAMETER 34	0	0	0	0	0	0	0	Adfmt
DATA ZONE	X25 ADDRESS STRING length defined by Adlen, format defined by Adfmt (ASCII or BCD)							

Adr: PARAMETER #8 **DB7.DB6.DB5.DB4.DB2.DB1.DB0**

Parameter #8 defines the address used in the LAPB link layer. The only values accepted are 1 and 3.

Address	Behavior
1 (B)	ETTD/DTE/Client
3 (A)	ETCD/DCE/Network

T1l: PARAMETER #9 **DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

T1h: PARAMETER #10 **DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

The bits in parameters #9 and #10 define the least and most significant bytes of the T1 retransmission timeout value for the LAPB link layer, in milliseconds. A value of 0 selects the default value of 2550 ms, whatever the line speed.

The table below indicates the minimum values for the T1 parameter for different line speeds.

Line speed	Minimum value for T1
128 Kbps	100 ms
115.2 Kbps	100 ms
64 Kbps	200 ms
57.6 Kbps	200 ms
38400 bps	300 ms
28800 bps	400 ms
19200 bps	800 ms
14400 bps	800 ms
9600 bps	1600 ms
7200 bps	2000 ms
4800 bps	2550 ms

In general, the T1 timeout should be greater than the time required to send the largest possible frame (263 bytes + bits inserted for transparency), and to receive two frames that are also of maximum length. You should add to this the maximum time required for the device receiving the frame to send an acknowledge, the signal propagation time over the transmission lines and through the modems, and the time required to process a frame received and to de-activate the T1 timer.

T2l: PARAMETER #11 **DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

T2h: PARAMETER #12 **DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

These parameters define the least and most significant bytes of the maximum amount of time (called T2) available to the MCX card to acknowledge receipt of an LAPB information frame. In the current version of the software, the MCX card does not delay acknowledgement for more than the transmission time for a frame of maximum length (this is the most unfavorable case!)

These 2 parameters must therefore be set to 0.

T3l: PARAMETER #13 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
T3h: PARAMETER #14 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the surveillance timeout T3 in the LAPB layer. In the current version of the software, the card does not manage this timeout. These 2 parameters must therefore be set to 0.

LUI: PARAMETER #15 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
LUh: PARAMETER #16 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the maximum size of data in an LAPB information frame, or in an X25 data packet.

For the LAPB protocol, this size must be between 1 and 2048 bytes. The standard size of 259 bytes is allocated if these 2 parameters are null.

For the X25 protocol, only the following values are accepted:

- 128 bytes (this is the standard length)
- 256 bytes
- 512 bytes
- 1024 bytes
- and 2048 bytes

The standard size of 128 bytes is allocated if these 2 parameters are null.

N2: PARAMETER #17 DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #17 define the maximum number of retransmissions for an LAPB information frame, or for a switch to LAPB set mode frame (the N2 parameter in the LAPB protocol).

All values between 0 and 63 are authorized.

The value 0 sets the number of N2 retransmissions to 10.

K: PARAMETER #18 DB7.DB6.DB5.DB4.DB2.DB1.DB0

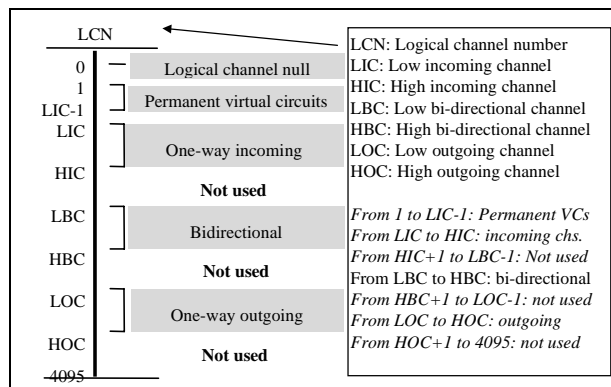
Parameter #18 defines the size of the LAPB window (K parameter).

All values between 0 and 7 are authorized.

The value 0 sets the K window to 7.

7 is the suggested value.

The following parameters are specific to the X25 protocol:



Breakdown and numbering of logical channels

LICl: PARAMETER #19 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 LICh: PARAMETER #20 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the first logical channel number (called LIC) available to the logical switched channels. As permanent VCs have not been implemented yet, the LIC parameter must be set to the value 1 (LICl=1, LICh=0).

HICl: PARAMETER #21 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 HICH: PARAMETER #22 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the number of the highest incoming one-way logical channel (called HIC). As the range of logical channels [LIC,HIC] is not supported, this parameter must be set to the value 4096 (HICl=0, HICH=16).

LBCl: PARAMETER #23 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 LBCh: PARAMETER #24 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the number of the lowest two-way logical channel. Acceptable values for this parameter are between 1 (LBCl=1, LBCh=0) and 4095 (LBCl=255, LBCh=15).

HBCl: PARAMETER #25 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 HBCh: PARAMETER #26 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the number of the highest two-way logical channel. The HBC parameter must be greater than or equal to the LBC parameter, and less than or equal to 4095.

LOCl: PARAMETER #27 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 LOCh: PARAMETER #28 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the lowest outgoing logical channel (called LOC). As the range of logical channels [LOC,HOC] is not supported, this parameter must be set to the value 0 (HOCl=0, HOCh=0).

HOCl: PARAMETER #29 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
HOCh: PARAMETER #30 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the number of the highest outgoing one-way logical channel (called HOC). As the range of logical channels [LOC,HOC] is not supported, this parameter must be set to the value 4096 (HOCl=0, HOCh=16).

CNTl: PARAMETER #31 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
CNTh: PARAMETER #32 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the least and most significant bytes of the address where the receive and transmit counters are located in the mailbox. If CNTl and CNTh are null, the address of the counters is calculated by the card, and returned in these parameters. (See section VI.1)

Adlen: PARAMETER #33 DB3.DB2.DB1.DB0

The 4 least significant bits of this parameter define the length of the X25 address in digits written in the data zone in ASCII or BCD format.

Adfmt: PARAMETER #34 DB0

Bit DB0 in parameter #34 defines the format of the X25 address written in the data zone.

DB0	X25 address format
0	ASCII
1	BCD

DATA ZONE

The data zone contains the X25 NON TOA/NPI address string written in ASCII or BCD format.

ERROR CODE

If the command executes normally, error code 0 is returned. If not, code 170 (AAh) is returned indicating an error in the command.

The error may be caused by:

- Channel number not between 1 and the number of installed channels (max=64)
- Error in parameter #2: undefined protocol
- ASYNC: Error parameters #5 or #6
- HDLC: Parameters #8 and #9 not null
- LAPB: Error parameter #8: Invalid address
- HDLC, BISYNC: Parameters #10, #11, #12, #13, #14 not null
- LAPB: Error parameters #11 and #12: T2 invalid
- LAPB: Error parameters #13 and #14: T3 invalid
- HDLC, BISYNC, LAPB: Error parameters #15 and #16: Data size greater than 2048
- HDLC, BISYNC: Parameters #17 or #18 not null
- LAPB: Parameter #17 greater than 63
- LAPB: Parameter #18 greater than 7
- X25: Parameters #19 to 20: Size of LIC invalid
- X25: Parameters #21 to 22: Size of HIC invalid
- X25: Parameters #23 to 26: Size of LBC or HBC invalid
- X25: Parameters #27 to 28: Size of LOC invalid
- X25: Parameters #29 to 30: Size of HOC invalid
- X25: Parameter #33: Most significant nibble not null
- X25: Parameter #34: Bits DB1 to DB7 not null
- X25: Number of logical channels greater than 128 for 128-or 256-byte packets
- X25: Number of logical channels greater than 64 for 512-byte packets
- X25: Number of logical channels greater than 32 for 1024-byte packets
- X25: Number of logical channels greater than 16 for 2048-byte packets
- X25: Number of logical channels null
- X25: Data zone: Invalid X25 address (includes non-numerical characters)

V.2. ALLOC (01h): Allocate buffer memory for an asynchronous channel

OPCODE = 1 (01h)

DESCRIPTION

This command allocates memory for an asynchronous channel's receive or transmit buffer. Buffer size is limited to 31 Kbytes.

When executed on a synchronous channel, this command simply returns an error and has no effect. The transmit and receive buffers for synchronous channels are static, and are initialized when the protocol is established (see the PROTO command).

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	0	0	0	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 3	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 4	0	TPB	0	0	0	0	0	0

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 specify the channel number. This number must be between 1 and the number of channels installed.

Ln1: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter is the least significant byte of the size (expressed in bytes) to allocate for the selected buffer.

Ln1: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter is the most significant byte of the size (expressed in bytes) to allocate for the selected buffer. Warning: the buffer size must never exceed 31 Kbytes.

BTy: PARAMETER #4 DB6

This bit identifies the type of buffer for which memory should be allocated as per the following table:

DB6	Buffer type
0	Transmit
1	Receive

ERROR CODE

If the command executes normally on an asynchronous channel, the error code returned equals 0. If not, code 130 (82h) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Illegal combination on parameter #4
- Size of a TX or RX buffer greater than 31 Kbytes or equal to 0
- Size of an RX buffer less than or equal to 148 bytes
- Not enough memory

If the command is executed on a synchronous channel, error code 189 (BFh) is returned (command not authorized).

NOTE CONCERNING BUFFERS ASSOCIATED WITH ASYNCHRONOUS CHANNELS

When the card is powered on, all channels are initialized in asynchronous mode, and the associated buffers are programmed with the following default values:

- 8192 bytes for the transmit buffers
- 512 bytes for the receive buffers

These values are used until a DALOC command, a new ALLOC command, or the PROTO command is issued.

V.3. BDELE (06h): Clear TX buffer and stop

OPCODE = 6 (06h)

DESCRIPTION

For an asynchronous channel

This command stops the ongoing transmission of a buffer, and totally erases the memory. This feature is especially useful for VIDEOTEXT servers, where you may sometimes wish to quickly stop pages from executing in fast mode with the "MORE" key.

For an HDLC channel

Sending a BDELE command to an HDLC channel stops the transmission of the current frame (if appropriate) then clears all the buffers containing frames that have not yet been sent. Stopping the transmission of an HDLC frame triggers an ABORT sequence.

For a BISYNC channel

Sending a BDELE command to a BISYNC channel stops the transmission of the current frame (if appropriate) then clears all the buffers containing frames that have not yet been sent.

For an LAPB channel

Sending a BDELE command to a LAPB channel clears all the buffers containing information frames that have not yet been sent

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	0	1	1	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#

C#: PARAMETER #1

DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 specify the channel number. This number must be between 1 and the number of channels installed.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 135 (87h) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not between 1 and the number of channels installed (64 max.)
- Corresponding buffer not allocated by the ALLOC command if the channel is in asynchronous mode.

IMPORTANT NOTE

The BDELE command does not trigger an IT7 interrupt, even if this condition was programmed by the MINTR command.

V.4. BPARM (18h): Read buffer parameters

OPCODE = 24 (18h)

This command reads the characteristics of the card's internal buffers used for sending and receiving data packets.

For channels programmed in asynchronous mode, this command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	1	0	0	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	BTy	BTy	0	0	0	0	0	0
DATA ITEM 1	Cnl	Cnl	Cnl	Cnl	Cnl	Cnl	Cnl	Cnl
DATA ITEM 2	Cnh	Cnh	Cnh	Cnh	Cnh	Cnh	Cnh	Cnh
DATA ITEM 3	SGl	SGl	SGl	SGl	SGl	SGl	SGl	SGl
DATA ITEM 4	SGh	SGh	SGh	SGh	SGh	SGh	SGh	SGh
DATA ITEM 5	ADl	ADl	ADl	ADl	ADl	ADl	ADl	ADl
DATA ITEM 6	ADh	ADh	ADh	ADh	ADh	ADh	ADh	ADh
DATA ITEM 7	LnL	LnL	LnL	LnL	LnL	LnL	LnL	LnL
DATA ITEM 8	LnH	LnH	LnH	LnH	LnH	LnH	LnH	LnH
DATA ITEM 9	IDl	IDl	IDl	IDl	IDl	IDl	IDl	IDl
DATA ITEM 10	IDh	IDh	IDh	IDh	IDh	IDh	IDh	IDh
DATA ITEM 11	Ril	Ril	Ril	Ril	Ril	Ril	Ril	Ril
DATA ITEM 12	Rih	Rih	Rih	Rih	Rih	Rih	Rih	Rih

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 define the channel number. This must be a number between 1 and the number of channels installed.

BTy: PARAMETER #2 DB7.DB6

These two bits identify the buffer type:

DB7	DB6	Buffer type
0	0	Transmit buffer
0	1	Receive buffer
1	0	Forbidden combination
1	1	Forbidden combination

Cnl: DATA ITEM # 1 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This data item is the image of the least significant byte of the FINBUF pointer used for buffer management.

Cnh: DATA ITEM # 2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This data item is the image of the most significant byte of the FINBUF pointer used for buffer management.

SGl: DATA ITEM # 3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 3 are the least significant bits of the segment containing the requested buffer.

SGh: DATA ITEM # 4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 4 are the most significant bits of the segment containing the requested buffer.

ADl: DATA ITEM # 5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 5 are the least significant bits of the address where the requested buffer is located.

ADh: DATA ITEM # 6 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 6 are the most significant bits of the address where the requested buffer is located.

LnL: DATA ITEM # 7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 7 are the least significant bits of the length initially allocated by the ALLOC command.

LnH: DATA ITEM # 8 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 8 are the most significant bits of the length initially allocated by the ALLOC command.

IDl: DATA ITEM # 9 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 9 are the least significant bits of the buffer index.

IDh: DATA ITEM # 10 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 10 are the most significant bits of the buffer index.

Ril: DATA ITEM # 11 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 11 are the least significant bits of the rotating index used in buffer management.

Rih: DATA ITEM # 12 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in data item # 12 are the most significant bits of the rotating index used in buffer management.

ERROR CODE ON AN ASYNCHRONOUS CHANNEL

If the command executes normally, an error code of 0 is returned. If not, code 148 (94h) is returned indicating an error in the command.

The cause of the error may be:

- Channel # not between 1 and the number of channels installed (max = 64)
- Buffer not allocated for the ALLOC command
- Forbidden combination for BTy (11) parameter 2

IMPORTANT NOTE

The buffer segment and address returned by the BPARM command specify its actual address in memory. The RMEMO command simply reads the buffer contents.

For channels programmed in synchronous mode, this command uses the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	1	0	0	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	BTy	BTy	0	0	0	0	0	0
DATA ITEM 1	BSizel	BSizel	BSizel	BSizel	BSizel	BSizel	BSizel	BSizel
DATA ITEM 2	BSizeh	BSizeh	BSizeh	BSizeh	BSizeh	BSizeh	BSizeh	BSizeh
DATA ITEM 3	W	W	W	W	W	W	W	W

#B: PARAMETER # 1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 define the channel number. This must be a number between 1 and the number of channels installed.

BTy: PARAMETER # 2 DB7.DB6

These two bits identify the buffer type:

DB7	DB6	Buffer type
0	0	Transmit buffer
0	1	Receive buffer
1	0	Forbidden combination
1	1	Forbidden combination

BSizel: DATA ITEM # 1 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This data item specifies the least significant byte of the maximum user data size in a data packet.

BSizeh: DATA ITEM # 2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This data item specifies the most significant byte of the maximum user data size in a data packet.

W: DATA ITEM # 3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This data item specifies the number of buffers used to send/receive data packets.

ERROR CODE ON A SYNCHRONOUS CHANNEL

If the command executes normally, an error code of 0 is returned. If not, the code 148 (94h) is returned indicating an error in the command.

The cause of the error may be:

- Channel # not between 1 and the number of channels installed (max = 64)
- Forbidden combination for BTy (11) parameter 2

V.5. BREAK (23h): Send a break on an asynchronous channel

OPCODE = 35 (23h)

This command sends a “Break” on an asynchronous transmission channel. The “Break” is a series of zeroes during a given time period.

There are two types of “Breaks”:

Type of break	Duration
short	1 second
long	4 seconds

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	1	0	0	0	1	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	0	0	0	0	0	0	0	Tbk

C#: PARAMETER # 1

DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 define the channel number. This must be a number between 1 and the number of channels installed.

Tbk: PARAMETER # 2

DB0

This bit in parameter number 2 indicates the type of “Break” to send:

Tbk	Type of break
0	short
1	long

IMPORTANT NOTE

This command can only be used correctly if the channel in question has already been initialized by the VINIT command.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, the code 158 (9Eh) is returned to indicate an error in the command parameter.

The cause of the error may be:

- Channel # not between 1 and the number of channels installed (max = 64)
- Error parameter number 2

On a synchronous channel, the command returns error code 189 (unauthorized command).

V.6. BTRAN (08h): Data transmission

OPCODE = 8 (08h)

DESCRIPTION

This command loads a transmit buffer and starts sending characters.

The data block size must be provided in parameters #2 and #3.

For an asynchronous channel

The data block size must never exceed the amount of memory allocated by the ALLOC command (8192 bytes is the default at power-on).

This command can only be used correctly after performing the following operations:

Allocate the buffer with the ALLOC command.

Initialize the communication parameters with the VINIT command.

Once the command has been sent, the card automatically starts sending characters on the requested line, and gradually frees space in the buffer. After a certain amount of time, a new transmission may be started before the previous one has finished, although you must make sure that there is enough free space in the buffer to receive the new block of information. The transmit buffers are rotary.

Use the TFREE command at any time to discover the space available in the transmit buffer.

For an HDLC or BISYNC channel

The maximum data block size is the maximum size allocated by the PROTO command.

The BTRAN command can only be used correctly after performing the following operations:

The channel must be configured in synchronous HDLC or BISYNC with the PROTO command.

Then the channel must be initialized with the VINIT command.

The BTRAN command simply stacks up the data from the mailbox into a free transmit buffer. The stacked buffer(s) will be transmitted over the line if the bit level is operational. The buffer is freed once it has been completely transmitted. The TFREE command provides the number of available transmit buffers at any given time.

For an LAPB channel

The maximum data block size is limited to the maximum size allocated by the PROTO command.

The BTRAN command can only be used correctly after performing the following operations:

Configure the channel in LAPB with the PROTO command.

Initialize the channel with the VINIT command.

Establish the LAPB connection (the PRCTL+LAPLNKUP command followed by a type 40h interrupt event, with a "link connected" status indicator).

Once the BTRAN command has been sent, the card stores the information frame in one of the K free transmit buffers. Once the preceding transmissions have completed, the card transmits the information frame on the line if the line's logical state allows it. The occupied buffer is only freed once its acknowledge has been received.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	1	0	0	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 3	Lnh	Lnh	Lnh	Lnh	Lnh	Lnh	Lnh	Lnh
DATA ZONE	DATA TO BE TRANSMITTED							

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 define channel number. This number must be between 1 and the number of channels installed.

Ln1: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the least significant byte of the size of the data block to transmit.

Lnh: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the most significant byte of the size of the data block to transmit.

DATA ZONE

The data zone of the dual-ported memory contains the characters to be copied to the card. The data zone is located at a positive offset of 100 (64h) from the base address in the dual-ported memory.

ERROR CODE ON AN ASYNCHRONOUS CHANNEL

If the command executes normally, an error code of 0 is returned. If not, code 137 (89h) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Size equal to 0 or greater than 31 Kbytes
- Size requested greater than the size reserved
- Size available less than the size requested
- Transmit buffer not reserved by the ALLOC command

ERROR CODE ON A HDLC OR BISYNC CHANNEL

If the command executes normally, an error code of 0 is returned. If not, code 137 (89h) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Size equal to 0 or greater than the allocated size
- No more buffers available

ERROR CODE ON A LAPB CHANNEL

If the command executes normally, an error code of 0 is returned. If not, code 137 (89h) is returned to indicate an error in the command. If a protocol error is detected, the command returns an error code of 188 (BCh).

The cause of error 137 may be:

- Channel # not included between 1 and the number of channels installed
- No more buffers available
- Size equal to 0

The cause of error 188 may be:

- Size of the information block greater than the maximum data size defined in the PROTO command: Error code 2 at address 83 (53h) of the INTERRUPT ZONE.
- Logical state of the link incorrect for the type of operation requested: error code 1 at address 83 (53h) in the INTERRUPT ZONE

IMPORTANT NOTE

A connection may be broken at the request of the remote device or following a serious error, at the same time as the BTRAN command is issued. In this case, although the initial status was "link connected", the BTRAN command returns error 188 preceded by a type 40h interrupt event with an indication of "link disconnected" or "Reinitializing".

V.7. CHDEF (0Ah): Define the character string for an asynchronous channel

OPCODE = 10 (0Ah)

This command informs the card of the end-of-string characters that, when detected, trigger an IT3 interrupt event on the PC bus to indicate that a complete phrase has been received. This condition may be selected or masked using the MINTR command.

Note that the end-of-string may be detected over 1 or 2 characters. For example, the Minitel's "RETURN" key sends two characters.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	1	0	1	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	0	0	0	NbS	NbS	NbS	NbS	NbS
PARAMETER 3	0	0	0	0	0	0	StL	StL
PARAMETER 4	St0	St0	St0	St0	St0	St0	St0	St0
PARAMETER 5	St1	St1	St1	St1	St1	St1	St1	St1
PARAMETER 6	0	0	0	0	0	0	StL	StL
PARAMETER 7	St0	St0	St0	St0	St0	St0	St0	St0
PARAMETER 8	St1	St1	St1	St1	St1	St1	St1	St1
PARAMETER 48	0	0	0	0	0	0	StL	StL
PARAMETER 49	St0	St0	St0	St0	St0	St0	St0	St0
PARAMETER 50	St1	St1	St1	St1	St1	St1	St1	St1

C#: PARAMETER # 1

DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 define the channel number. This must be a number between 1 and the number of channels installed.

NbS: PARAMETER # 2

DB4.DB3.DB2.DB1.DB0

This parameter indicates the number of character strings that will be used with the MINTR command to warn that CPU that a data block has been received. 16 strings with 1 or 2 characters may be programmed. The characters in a string may be, for example, a CR; a CR,LF sequence; or 2 Minitel ASCII codes.

The NbS parameter must be between 1 and 16.

StL: PARAMETER # 3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48

DB1.DB0

This parameter indicates the number of characters in the string defined below. It may equal 1 or 2; it must be repeated before each new string definition.

STn: PARAMETER # 4+n, ...,49+n DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

The STn parameters specify the ASCII characters that will define the end-of-string characters. For example, if in parameter #3, the coded length for StL is 1, then parameter #4 contains the character while parameter #5 is not significant.

ERROR CODE

If the command executes normally, the error code returned equals 0. If not, code returned equals 140 (8Ch).

The cause of the error may be:

- Channel number not between 1 and the number of lines installed (max = 64)
- Number of strings greater than 16 or equal to 0
- Number of characters in a string greater than 2

On a synchronous channel, the command returns error code 189 (unauthorized command).

IMPORTANT NOTE

This function is ignored if the corresponding interrupt (IT3) is not selected by the MINTR command.

If the IT3 interrupt event is programmed but the CHDEF command has not been sent, an IT3 interrupt is triggered for each carriage return (ASCII code 13).

V.8. CLRRX (1Dh): Clear data received**OPCODE = 29 (1Dh)****DESCRIPTION****For an asynchronous channel**

This command completely erases the contents of the receive buffer. All the characters received are lost.

For an HDLC channel

This command completely erases the contents of the 14 receive buffers. All frames received are lost.

For an LAPB channel

This command completely erases the contents of the K receive buffers. All information frames received are lost.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	1	1	0	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#

C#: PARAMETER #1**DB6.DB5.DB4.DB3.DB2.DB1.DB0**

The <C#> bits in parameter #1 define the channel number. This number must be between 1 and the number of channels installed.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 151 (97h) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Corresponding asynchronous buffer not reserved by the ALLOC command

V.9. DALOC (15h): Deallocate asynchronous buffer memory**OPCODE = 21 (15h)****DESCRIPTION**

This command frees all space reserved for transmit and receive buffers for all channels programmed in asynchronous mode.

You will have to reserve memory for the buffers associated with the asynchronous channels using the ALLOC command.

This command may be used to remove all the allocations executed by default when the card is powered on.

This command always returns error code 0, and does not affect synchronous channels.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	0	1	0	1

ERROR CODE

No parameters are required for this command, and the error code returned is always zero.

V.10. GOADR (12H): Execute a program in memory**OPCODE = 18 (12h)****DESCRIPTION**

This command executes a program in memory starting at the segment address indicated by parameters #1, #2, #3 and #4. This program must have been loaded previously with the MBOOT command. Once the program is started, you lose control of the card during the time that the program executes. Character transmission and reception are not suspended, however, unless a CLI (forbid interruptions) command is executed by the program that is launched.

For a program to return control to the interpreter, it must finish with a RETF (0CBh) command, and must not modify the STACK POINTER or the STACK SEGMENT. The system provides 128 bytes on the stack for the launched program. If that is not enough, you should declare a local stack and save the position of the stack in your initial instructions.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	0	0	1	0
PARAMETER 1	SGl	SGl	SGl	SGl	SGl	SGl	SGl	SGl
PARAMETER 2	SGh	SGh	SGh	SGh	SGh	SGh	SGh	SGh
PARAMETER 3	ADl	ADl	ADl	ADl	ADl	ADl	ADl	ADl
PARAMETER 4	ADh	ADh	ADh	ADh	ADh	ADh	ADh	ADh

SGl: PARAMETER #1 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #1 contains the least significant byte of the segment that contains the program to be started.

SGh: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #2 contains the most significant byte of the segment that contains the program to be started.

ADl: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #3 are the least significant byte of the program start address.

ADh: PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #4 are the most significant byte of the program start address.

ERROR CODE

The GOADR command does not return an error code. Nevertheless, a zero is written in the status of the mailbox when the launched program returns control to the card's command interpreter.

V.11. HNGUP (16h): Hang up Modem

OPCODE = 22 (16h)

This command hangs up the modem for the channel in question. This operation temporarily resets the DTR signal.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	0	1	1	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	0	0	0	0	0	Tmp	Tmp	Tmp

C#: PARAMETER #1 **DB6.DB5.DB4.DB3.DB2.DB1.DB0**

The <C#> bits in parameter #1 define the channel number. This must be a number between 1 and the number of channels installed.

Tmp: PARAMETER # 2 **DB2.DB1.DB0**

These three bits specify the time (in seconds) during which the DTR signal must be low so that the modem's electronics hang up the line. This value must be between 1 and 7 seconds.

It is programmed in order to meet the requirements of different types of modems.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, the code 159 (9Fh) is returned indicating an error in the command.

The cause of the error may be:

- Channel # not between 1 and the number of channels installed (max = 64).
- Time requested out of bounds

V.12. MBOOT (05h): Load a program in memory**OPCODE = 5 (05h)****DESCRIPTION**

This command loads a program into the card's memory by blocks.

Put the address of the code block (SEG:ADR) in parameters #1, #2, #3 and #4, and the size of the code in parameters #5 and #6. Warning: if the sum of the program size and the start address is greater than 64 Kbytes (0FFFFH), any code with an address greater than 10000H will be written to addresses zero and above in the same segment.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	0	1	0	1
PARAMETER 1	SGl	SGl	SGl	SGl	SGl	SGl	SGl	SGl
PARAMETER 2	SGh	SGh	SGh	SGh	SGh	SGh	SGh	SGh
PARAMETER 3	ADl	ADl	ADl	ADl	ADl	ADl	ADl	ADl
PARAMETER 4	ADh	ADh	ADh	ADh	ADh	ADh	ADh	ADh
PARAMETER 5	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 6	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
DATA ZONE	BLOCK OF CODE TO LOAD INTO MEMORY							

SGl: PARAMETER #1 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #1 contains the least significant byte of the segment where the block of code should be loaded.

SGh: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #2 contains the most significant byte of the segment where the block of code should be loaded.

ADl: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #3 contains the least significant byte of the address where the code block should be loaded in the card's memory.

ADh: PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #4 contains the most significant byte of the address where the code block should be loaded in the card's memory.

Ln1: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #5 contains the least significant byte of the size of the block to be loaded into memory.

Ln: PARAMETER #6 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #6 contains the most significant byte of the size of the block to be loaded into memory.

Warning: the block size must not exceed the size of the mailbox minus 1 Kbyte. If not, an error will be indicated.

DATA ZONE

The data zone of the dual-ported memory must contain the code block to be copied to the card.

Reminder: the data zone is located at a positive offset of 100 (64h) from the base address of the dual-ported memory.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 134 (86h) is returned to indicate an error in the command.

The cause of the error is:

- Size of the code block greater than 31 Kbytes or equal to 0.

V.13. MINTR (0Ch): "Event" interrupt conditions**OPCODE = 12 (0Ch)****DESCRIPTION**

This command enables or masks the card's "event" interrupt conditions.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	1	1	0	0
PARAMETER 1	Mde	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	IT7	IT6	IT5	IT4	IT3	IT2	IT1	IT0

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter indicates the number of the channel to be initialized. It should be a number between 1 and the number of channels installed.

Mde: PARAMETER #1 DB7

This parameter modifies the meaning of IT2, depending if it is set to 1 or 0.

ITn: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #2 specify the events that will generate an interrupt on the PC bus. The following table describes how these bits correspond to different events:

For an asynchronous channel

IT#	Event that triggers interrupt	Comments
IT0	Character received.	Except handshake characters. The character is not added to the buffer but is given to the interrupt. The character counter is not modified.
IT1 (STCNT=1)	First character received.	See STCNT command.
IT1 (STCNT>1)	Sequence of STCNT characters received.	See STCNT command.
IT2 (Mde=0)	Receive buffer full.	
IT2 (Mde=1)	Receive buffer has gone from empty to not-empty.	
IT3	End of string	See CHDEF command
IT4	Time out	See STTMO command.
IT5	Receive error detected.	Parity error, frame error or character lost.
IT6	CTS, CD, or RI signals changed state, or BREAK sequence detected.	
IT7	Transmit buffer completely empty (including stop bit).	

For an HDLC or BISYNC channel

IT#	Event that triggers interrupt	Comments
IT0	Not defined.	
IT1	Frame received.	
IT2 (Mde=0)	14 frames received.	
IT2 (Mde=1)	First frame received.	
IT3	Not defined.	
IT4	Time-out	See STTMO command.
IT5	Receive error detected.	Frame or CRC error (in HDLC mode), characters in controller's FIFO buffer overwritten.
IT6	CTS or DCD signals changed state, or an ABORT signal detected in HDLC mode.	
IT7	No more frames to send	

For an LAPB channel

IT#	Event that triggers interrupt	Comments
IT0	Not defined.	
IT1	Information frame received.	
IT2 (Mde=0)	Receive window full	
IT2 (Mde=1)	Switched receive window from 0 to 1	
IT3	Not defined.	
IT4	Time-out	See STTMO command.
IT5	Receive error detected.	Frame or CRC error, characters in the controller's FIFO buffer overwritten, frame lost.
IT6	CTS or DCD signals changed state, or an ABORT signal detected, or the link's logical state has changed.	
IT7	Transmit window empty.	

NOTE

Each of the card's channels can have a different operating mode for interrupts. You can also select several conditions for a given channel.

The new set of conditions simply replace the previous conditions, however.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 142 (8Eh) is returned to indicate an error in the parameters.

The cause of the error is:

- Channel # not between 1 and the number of channels installed (64 max.)

V.14. NOPER (2Bh): NOP command (no operation)**OPCODE = 43 (2Bh)****DESCRIPTION**

This command starts an empty command on the card which generates an end of command interrupt.

This command may be used to test the dialog mechanism between the PC and the interpreter.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	1	0	1	0	1	1

ERROR CODE

This command always returns an error code equal to 0.

V.15. PRCTL (30h): Control procedures**OPCODE = 48 (30h)****DESCRIPTION**

This command only affects LAPB or X25 channels.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	1	1	0	0	0	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	0	0	0	0	0	0	0	0
PARAMETER 3	0	0	0	0	0	0	0	0
PARAMETER 4	0	0	0	0	0	0	0	0
PARAMETER 5	0	0	0	0	0	0	0	0
PARAMETER 6	0	0	0	0	0	0	0	0
PARAMETER 7	OP	OP	OP	OP	OP	OP	OP	OP

The following parameters are related to the opcode (parameter #7). They are described in separate tables.

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 specify the channel number. This must be a number between 1 and the maximum number of channels installed.

PARAMETER #2, #3, #4, #5, #6 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

All these parameters must be set to 0. They are reserved for future extensions.

OP: PARAMETER #7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #7 indicate the opcode associated with the PRCTL command.

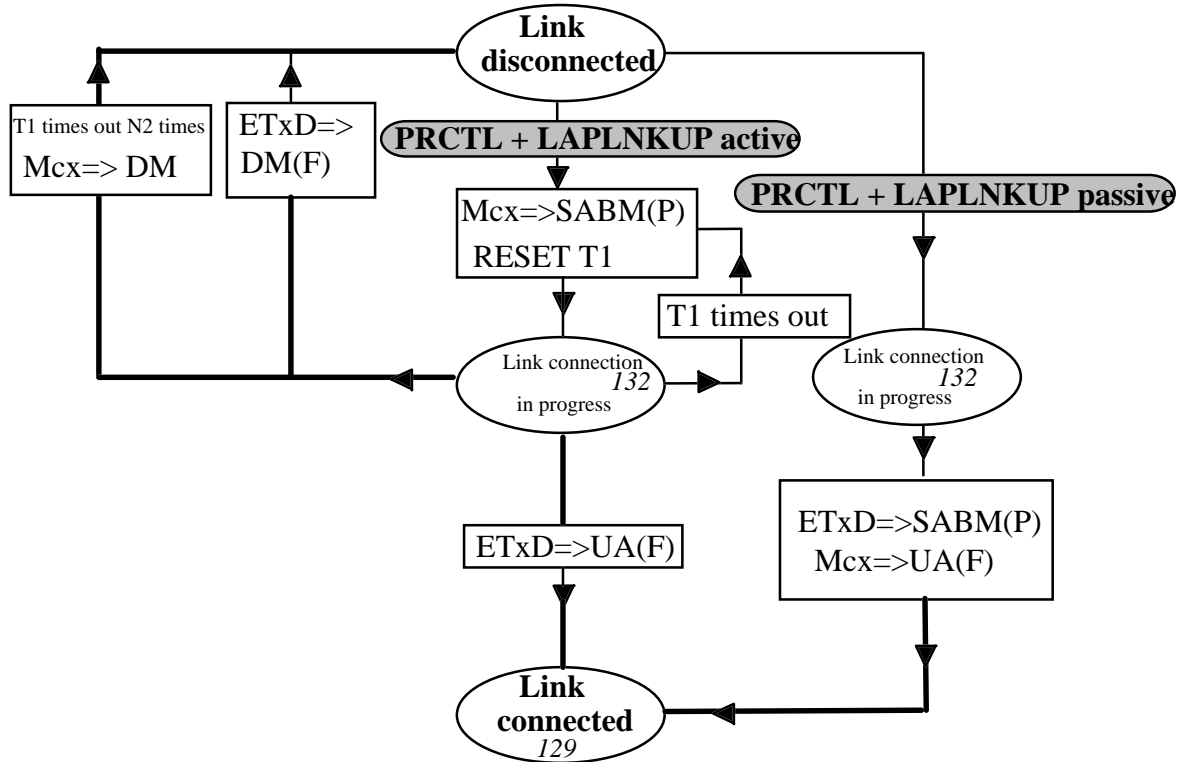
The following parameters are specific to each opcode, and are documented for each one.

0	0	0	0	0	0	0	0	LAPLNKUP (request an LAPB connection)
0	0	0	0	0	0	0	1	LAPLNKDN (request to disconnect an LAPB)
0	0	0	0	1	0	1	0	LAPSTATE (Logical state of the LAPB link)

LAPLNKUP opcode

PARAMETER 9	0	0	0	0	0	0	0	A
-------------	---	---	---	---	---	---	---	---

This opcode may only be used by the PRCTL command if the channel was programmed as per the LAPB protocol and if the LAPB link is in the "link disconnected" state. The opcode establishes an LAPB connection as illustrated by the following state diagram.



ETxD is the device communicating with the MCX card

Links with thick lines indicate that an IT6 interrupt event

(Change of logical state in the LAPB link) may be sent to the PC by the MCX card.

A: PARAMETER #9 **DB0**

This bit in parameter #9 associated with the LAPLNKUP opcode selects one of the two types of connection: either an "active" connection (initiated by the MCX card) or a "passive" connection (initialized by the device connected to the MCX card).

A = 0	Active connection
A = 1	Passive connection

A request for an active connection causes one or several unnumbered SABM frames with the P bit to be sent while waiting for a UA reply with the F bit. If a UA(F) reply is received before N2 SABM(P) frames are sent, the DTE and the DCE are set to an information transfer phase ("link connected" state). This active connection request may be cancelled by sending a disconnection request (see PRCTL+LAPLNKDN below). No reply or a DM reply with the F bit sets the MCX card in the disconnected state.

A request for a passive connection lets the MCX card acknowledge an SABM command frame (with or without the P bit) coming from the device connected to the card with a UA reply (with the corresponding F bit). The card then enters the information transfer phase (connected state). This connection request may be canceled by sending a disconnect request (see PRCTL+LAPLNKDN below).

The changes of the logical state of the LAPB link following a PRCTL + LAPLNKUP command trigger an IT6 interrupt to the PC (see the MINTR command) if that interrupt is enabled.

PRCTL + LAPLNKUP ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 171 (ABh) or 188 (BCh) is returned, indicating an error in the parameters or a protocol error.

If the command is executed on an asynchronous channel or on a synchronous HDLC envelope-level channel, BISYNC, then error code 189 (BDh) is returned (unauthorized command).

The cause of error 171 is:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Parameter #7 (opcode) invalid
- Parameter #7 (opcode) not authorized for the protocol

The cause of error 188 is:

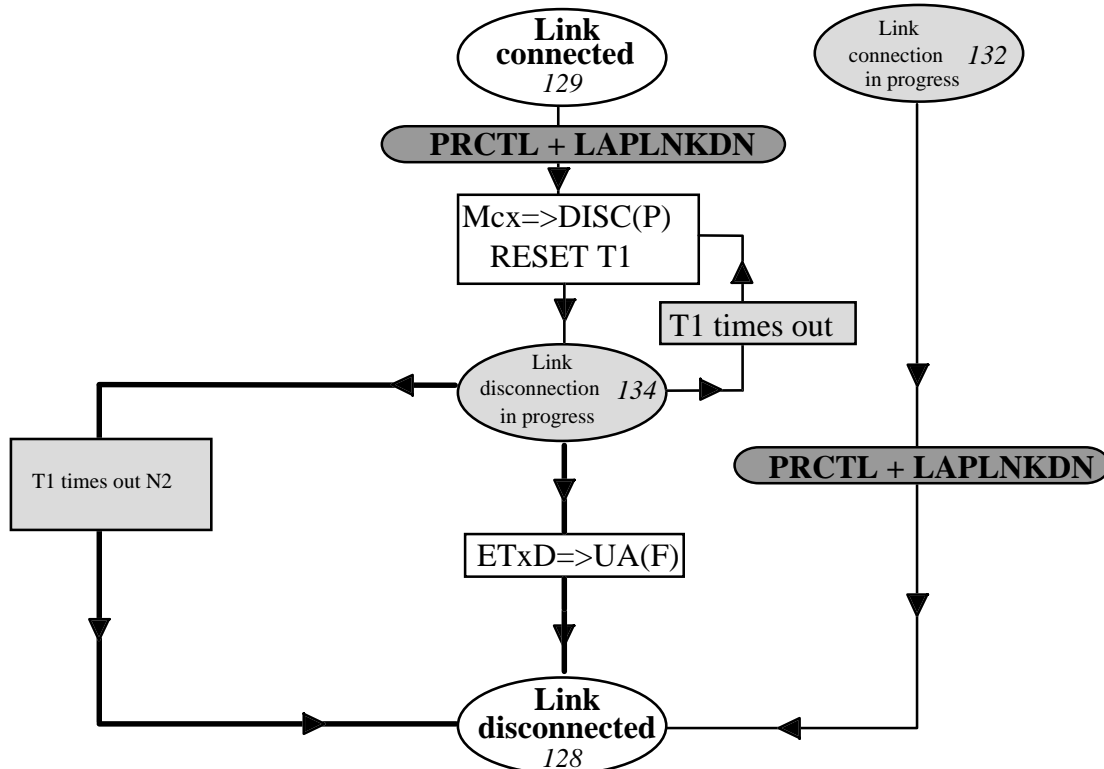
- Logical state of the link is incorrect for the type of operation requested: error code 1 at address 83 (53h) in the INTERRUPT ZONE

LAPLNKDN Opcode

The **LAPLNKDN** code requires no parameters.

This opcode can only be used by the PRCTL command if the channel was programmed for the LAPB protocol and if the LAPB logical link is in the information transfer state. It can also be used to cancel a passive connection request.

- “Link connected” state
- “Link connection in progress” state



EtxD is the device communicating with the MCX card

Links drawn with thick lines indicate that an IT16 interrupt event

(Change of logical state in the LAPB link) may be sent to the PC by the MCX card.

A request to disconnect causes one or several unnumbered DISC frames with the P bit to be sent, waiting for a UA reply with the F bit. If a UA(F) reply is received before sending N2 DISC(P) frames, the DTE and the DCE are placed in the “link disconnected” state. If the device connected to the card does not reply to the disconnect request after N2 attempts, the card signals the event by sending a DM frame, and enters the “link disconnected” state.

Changes in the LAPB link's logical state following the PRCTL + LAPLNKDN command trigger an IT6 interrupt to the PC (see the MINTR command) if this interrupt is enabled.

Caution: “link connected” and “link connection in progress” transitions are generated by the PC and do not generate IT6 interrupts.

PRCTL + LAPLNKDN ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 171 (ABh) or 188 (BCh) is returned to indicate an error in the parameters or a protocol error.

If the command is executed on an asynchronous channel or on a synchronous HDLC envelope-level channel, BISYNC, then error code 189 (BDh) is returned (unauthorized command).

The cause of error 171 is:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Parameter #7 (opcode) invalid
- Parameter #7 (opcode) not authorized for the protocol

The cause of error 188 is:

- Logical state of the link is incorrect for the type of operation requested: error code 1 at address 83 (53h) in the INTERRUPT ZONE

The LAPSTATE opcode

No parameters are required for the LAPSTATE opcode.

This opcode may only be used by the PRCTL command if the channel was programmed using the LAPB protocol. It provides the logical state of the LAPB link. This information is sent in data item #1 as indicated below:

DATA ITEM 1	S7	S6	S5	S4	S3	S2	S1	S0
-------------	----	----	----	----	----	----	----	----

Data item 1 = 128: *Link disconnected*

Data item 1 = 129: *Link connected*

Data item 1 = 132: *Link connection in progress*

Data item 1 = 134: *Link disconnection in progress*

ERROR CODE PRCTL + LAPSTATE

The command always returns error code 0 on an LAPB channel.

If the command is executed on an asynchronous channel or on a synchronous HDLC envelope-level channel, BISYNC, then error code 189 (BDh) is returned (unauthorized command).

V.16. RDBUF (09h): Read a receive buffer**OPCODE = 9 (09h)****DESCRIPTION**

This command reads the contents of a receive buffer.

The command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	1	0	0	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 3	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 4	0	0	0	0	0	Typ	Typ	Typ
DATA ZONE	DATA READ							

C#: PARAMETER #1**DB6.DB5.DB4.DB3.DB2.DB1.DB0**

The <C#> bits in parameter #1 specify the channel number associated with the buffer to be read. This number must be between 1 and the maximum number of channels installed.

For an asynchronous channel

Ln1: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 This parameter is the least significant byte of the number of characters to read.

Ln2: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 This parameter is the most significant byte of the number of characters to read. Warning: the number of characters to read must never be greater than the size allocated in memory by the ALLOC command.

Typ: PARAMETER #4 DB2.DB1.DB0
 These bits in parameter #4 specify the type of read to perform as per the following table:

DB2	DB1	DB0	Type of read
0	0	0	Read (n) characters
0	0	1	No-destructive read of (n) characters
0	1	0	Read the entire buffer
1	0	0	Read (n) characters and copy them to the mailbox, even if the buffer does not contain enough characters

Other combinations of DB2, DB1, and DB0 are illegal and return an error.

Type 0: Read (n) characters.

In this case, parameters #2 and #3 must contain the number of characters to be read. This command modifies the card's internal pointers: it is therefore a destructive read. If the number of characters requested is greater than the number of characters in the buffer when the command is sent, the card returns error code 139 and parameters #2 and #3 are automatically loaded with the number of characters available in the buffer. In this case, the characters are not copied into the mailbox.

Type 1: Read (n) characters (non destructive).

This operation is identical to the previous one, except the card's internal pointers are not modified. This read is therefore non-destructive: the characters are not removed from the buffer.

Type 2: Read the entire buffer.

In this case, all characters in the buffer are read and removed. Parameters #2 and #3 are automatically loaded with the total number of characters in the buffer.

Type 4: Read (n) characters and copy them in the buffer, even if the number of characters requested is not currently present in the buffer.

In this case, parameters #2 and #3 contain the number of characters to be read. This command modifies the card's internal pointers: it is therefore a destructive read. If the number of characters requested is greater than the number of characters in the buffer when the command is sent, the card returns error code 139 and parameters #2 and #3 are automatically loaded with the number of characters available in the buffer. In this case, the data zone of the mailbox contains these characters.

DATA ZONE

The data zone in the dual-ported memory contains the characters read by this command. The data zone is located at a positive offset 100 (64h) from the dual-ported memory base address.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 138 (8Ah) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Number of characters to read equal to 0, or greater than 31 Kbytes
- Number of characters to read greater than the size allocated
- Receive buffer not allocated by the ALLOC command
- Illegal combination for this type of read

If the error code equals 139 (8Bh), the number of characters requested is not currently in the buffer, and parameters #2 and #3 indicate the number of characters available. For a type 0 read, the characters are not copied to the mailbox. They are copied to the mailbox for a type 4 read, however, and deleted from the buffer.

For a synchronous channel

Lnl: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 This is an output parameter. After executing the RDBUF command, it contains the least significant byte of the number of characters in the frame's "data field".

Lnh: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 This is an output parameter. After executing the RDBUF command, it contains the most significant byte of the number of characters in the frame's "data field".

The 4 most significant bits have a special meaning for HDLC and BISYNC channels.

- 0 0 0 0: No error in the frame
- 1 0 0 0: Frame length greater than the allocated buffer size
- 0 1 0 0: Bad CRC in HDLC
- 0 0 1 0: Character overwritten in the SCC
- 0 0 0 1: ABORT sequence received in HDLC

Typ: PARAMETER #4 DB2.DB1.DB0
 These bits in parameter #4 indicate the kind of read to perform as per the table below:

DB2	DB1	DB0	Type of read
0	0	0	Read a frame
0	0	1	Read a frame without freeing the buffer
1	0	0	Read a frame

Type 0: Read a frame

This command frees the card's buffer: it is a destructive read. Parameters #2 and #3 contain the number of characters in the frame.

Type 1: Read a frame without freeing the buffer

This is a non-destructive read: the buffer is not freed. Parameters #2 and #3 contain the number of characters in the frame.

Type 4: Read a frame

This type of read, supported to provide a certain level of compatibility with asynchronous channels, operates exactly like type 0.

DATA ZONE

The data zone in the dual-ported memory contains the frame read by the command. The data zone is located at a positive offset 100 (64h) from the dual-ported memory base address.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 138 (8Ah) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Illegal combination for this type of read

V.17. RELRP (13h): Read codes, revisions, identifications**OPCODE = 19 (13h)****DESCRIPTION**

This command returns the following information:

- Revision of the multiprotocol software
- Revision of the basic software
- Confidential code
- Card ID code
- Number of lines installed
- Total memory size
- Microprocessor type
- Microprocessor clock frequency
- Math co-processor installed or not
- Type of back panel
- Auxiliary MCX-PWS power supply installed or not
- Type of serial extension

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	0	0	1	1
DATA ITEM 1	VFWM	VFWM	VFWM	VFWM	VFWM	VFWM	VFWM	VFWM
DATA ITEM 2	Cf1	Cf1	Cf1	Cf1	Cf1	Cf1	Cf1	Cf1
DATA ITEM 3	Cf2	Cf2	Cf2	Cf2	Cf2	Cf2	Cf2	Cf2
DATA ITEM 4	Cf3	Cf3	Cf3	Cf3	Cf3	Cf3	Cf3	Cf3
DATA ITEM 5	Cf4	Cf4	Cf4	Cf4	Cf4	Cf4	Cf4	Cf4
DATA ITEM 6	0	1	0	0	1	1	0	1
DATA ITEM 7	0	1	1	0	0	0	1	1
DATA ITEM 8	0	1	1	1	1	0	0	0
DATA ITEM 9	0	Nlg	Nlg	Nlg	Nlg	Nlg	Nlg	Nlg
DATA ITEM 10	Fpu	Cpu	Pws	Bpt	Mem	Mem	Mem	Mem
DATA ITEM 11	Clk	Clk	Clk	Clk	Clk	Clk	Clk	Clk
DATA ITEM 12	VMD	VMD	VMD	VMD	VMD	VMD	VMD	VMD
DATA ITEM 13	VDG	VDG	VDG	VDG	VDG	VDG	VDG	VDG
DATA ITEM 14	VBI	VBI	VBI	VBI	VBI	VBI	VBI	VBI
DATA ITEM 15	BPE	BPE	BPE	BPE	BPE	BPE	BPE	BPE
DATA ITEM 16	VFW	VFW	VFW	VFW	VFW	VFW	VFW	VFW
DATA ITEM 17	X	X	X	X	X	X	X	BAT
DATA ITEM 18	Ext	Ext	Ext	Ext	Ext	Ext	Ext	Ext

VFWM: DATA ITEM #1**DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

The bits in data item #1 encode the version number of the card's multiprotocol software. For example, 12H indicates revision 1.2.

Cf1: DATA ITEM #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This data item contains the card's confidential code #1. This value is masked and stored in the MCX card.

Cf2: DATA ITEM #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This data item contains the card's confidential code #2. This value is masked and stored in the MCX card.

Cf3: DATA ITEM #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This data item contains the card's confidential code #3. This value is masked and stored in the MCX card.

Cf4: DATA ITEM #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This data item contains the card's confidential code #4. This value is masked and stored in the MCX card. The four bytes Cf1Cf2Cf3Cf4 are written in the card during the manufacturing process. They contain a confidential code that lets you protect your program, by refusing to run if another confidential code is read.

When the purchaser did not request a confidential code, the command returns Cf1.Cf2.Cf3.Cf4 = 87.65.43.21h.

DATA ITEMS #6,7,8 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Data items 6,7,8 contain the values of the following ASCII characters:

Mcx

They provide a quick way to detect if the card is in the machine. They also let the programmer automatically determine the card's address and segment.

Nlg: DATA ITEM #9 DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits indicate the number of channels installed on the card. This number ranges from 2 to 64.

Mem: DATA ITEM #10 DB3.DB2.DB1.DB0

These bits indicate the card's total memory capacity, in Megabytes.

Bpt: DATA ITEM #10 DB4

This bit indicates the type of MCX-BP back panels connected:

Bpt = 0 Single oscillator units

Bpt = 1 Double oscillator units

Pws: DATA ITEM #10 DB5

This bit indicates whether or not the auxiliary MCX-PWS power supply is connected to the card.

Pws = 0 Power supply not installed

Pws = 1 Power supply installed

Cpu: DATA ITEM #10 DB6

This bit indicates the type of microprocessor installed on the card:

Cpu = 0 386SX
 Cpu = 1 486 SLC or 486 SXLC

Fpu: DATA ITEM #10 DB7

This bit indicates whether or not an 80387SX co-processor is installed on the card.

Fpu = 0 Co-processor not installed
 Fpu = 1 Co-processor installed

Clk: DATA ITEM #11 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

The bits in data item #11 indicate the clock frequency used by the card's processor. The value returned is in MHz.

VMD: DATA ITEM #12 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

The bits in data item #12 encode the version number of the MCX-DOS extension.

VDG: DATA ITEM #13 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

The bits in data item #13 encode the version number of the resident part of MCX-DEBUG.

VBI: DATA ITEM #14 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

The bits in data item #14 encode the MCX card's BIOS version number.

BPE: DATA ITEM #15 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits indicate the revision of the card's MCX-BP back panels.

DBi = 0 Back panel #i+1 revision less than E
 DBi = 1 Back panel #i+1 revision >= E

Units that are REV.E and higher include an additional 8 bit register that let you read the state of the RING INDICATOR signal on each of the unit's 8 lines.

VFW: DATA ITEM #16 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

The bits in data item #16 encode the version number of the card's basic software. For example, code 11H indicates revision 1.1.

BAT : DATA ITEM # 17 DB0

The bit BAT in data item # 17 indicates the state of the battery

BAT = 0.....OK
 BAT = 1.....not OK

EXT : DATA ITEM # 18 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits indicate the type of serial extension connected to the MCX-LITE/0 or MCX-00 Board.

EXT = 0.....MCX-BP Extension
 EXT = 1.....Lite Serial Extension
 EXT = 2.....Lite570 Extension

ERROR CODE

The RELRP command always returns error code 0.

V.18. RINIT (19h): Reinitialize the card**OPCODE = 25 (19h)****DESCRIPTION**

This command resets the card's software.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	1	0	0	1
PARAMETER 1	0	0	0	0	0	0	0	Typ

Typ: PARAMETER #1**DB0**

If Typ = 1, all the card's variables and communication lines are reset to zero. The card is then waiting for the "RUN 02" start code.

If Typ = 0, the card returns an error.

WARNING

Since the card does not return an end of command interrupt if Typ = 1, the programmer can determine if the card is ready to receive the RUN 02 start code by reading address 0 in the mailbox. Once the card is ready, it writes byte 0Fh at that address and the character string "MCX IS READY" at address 100 (64h).

ERROR CODE

If the command does not execute normally, code 154 (9Ah) is returned to indicate an error in the command.

The cause of the error is:

- Error in parameter #1

V.19. RMEMO (11h): Dump a block of memory**OPCODE = 17 (11h)****DESCRIPTION**

This command lists a memory block.

The block's address is supplied in parameters #1 and #2, while parameters #3 and #4 contain the block's segment.

Parameters #5 and #6 contain the size of the block to read.

The information is transferred into the card's data zone.

Warning: if the sum of the requested size and the start address is greater than 64 Kbytes (0FFFFH), the contents of addresses greater than 10000H will be replaced by the contents of addresses zero and above in the same segment.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	0	0	0	1
PARAMETER 1	SGl	SGl	SGl	SGl	SGl	SGl	SGl	SGl
PARAMETER 2	SGh	SGh	SGh	SGh	SGh	SGh	SGh	SGh
PARAMETER 3	ADl	ADl	ADl	ADl	ADl	ADl	ADl	ADl
PARAMETER 4	ADh	ADh	ADh	ADh	ADh	ADh	ADh	ADh
PARAMETER 5	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 6	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
DATA ZONE	LISTED DATA BLOCK							

SGl: PARAMETER #1 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #1 contains the least significant byte of the segment of the block to be listed.

SGh: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #2 contains the most significant byte of the block's segment.

ADl: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #3 contains the least significant byte of the address of the block to be listed.

ADh: PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #4 contains the most significant byte of the block's address.

Ln1: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #5 contains the least significant byte of the size of the block to be read.

Ln1: PARAMETER #6 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #6 contains the most significant byte of the block's size.

DATA ZONE

The data zone in the dual-ported memory contains the data block read. The data zone is located at a positive offset 100 (64h) from the dual-ported memory base address.

ERROR CODE

A zero error code is returned if the command executes normally. If not, the system returns error code 146 (92h).

The cause of the error may be:

- Data block size greater than 31 Kbytes
- Data block size equal to zero

V.20. RSMDE (2Dh): Initialize Electrical mode**OPCODE = 45 (2Dh)****DESCRIPTION**

This command selects the communication channel's operating mode.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	1	0	1	1	0	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	0	0	0	0	Mde	Mde	Mde	Mde

C#: PARAMETER #1**DB6.DB5.DB4.DB3.DB2.DB1.DB0**

These bits in parameter #1 contain the channel number. This number must be between and the number of lines installed.

Mde: PARAMETER #2**DB3.DB2.DB1.DB0**

These bits select the electrical interface for the channel.

DB3	DB2	DB1	DB0	Mode	MCX-LITE/570	MCX-LITE/S ET MCX-XX
0	0	0	0	RS232	Supported	Supported (Yellow LED off)
0	0	0	1	RS422	Supported	Supported (Yellow LED on)
0	0	1	0	RS485	Supported	Non supported
0	0	1	1	RS449	Supported	Non supported
0	1	0	0	V35	Supported	Non supported
0	1	0	1	V36	Supported	Non supported
0	1	1	0	EIA530	Supported	Non supported
0	1	1	1	EIA530-A	Supported	Non supported
1	0	0	0	Tri-State	Supported	Non supported

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 168 (A8h) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Parameter #2 incorrect

V.21. RSTAT (0Dh): Read the status of the communication channels**OPCODE = 13 (0Dh)****DESCRIPTION**

This command dynamically reads the status of all the communication channels.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	1	1	0	1
DATA ITEM 1	TXe	RXr	CDt	CTS	RIg	Par	Ovr	Brk
DATA ITEM 2	TXe	RXr	CDt	CTS	RIg	Par	Ovr	Brk
DATA ITEM 3	TXe	RXr	CDt	CTS	RIg	Par	Ovr	Brk
DATA ITEM 4	TXe	RXr	CDt	CTS	RIg	Par	Ovr	Brk
DATA ITEM 5	TXe	RXr	CDt	CTS	RIg	Par	Ovr	Brk
.....								
DATA ITEM 61	TXe	RXr	CDt	CTS	RIg	Par	Ovr	Brk
DATA ITEM 62	TXe	RXr	CDt	CTS	RIg	Par	Ovr	Brk
DATA ITEM 63	TXe	RXr	CDt	CTS	RIg	Par	Ovr	Brk
DATA ITEM 64	TXe	RXr	CDt	CTS	RIg	Par	Ovr	Brk

The various states are written consecutively for each channel in the dual-ported memory's data zone. The data zone is located at a positive offset 100 (64h) from the dual-ported memory base address.

TXe: DATA ITEMS 1 to the number of lines installed **DB7**

If this bit equals 1, the transmitter for the corresponding channel is ready to receive a new character. Note that internally, the communication processors can store several transmit characters.

RXr: DATA ITEMS 1 to the number of lines installed **DB6**

If this bit equals 1, a character was received by the receiver, and may be read by the card's CPU. Note that internally the communication processors can store several receive characters.

CDt: DATA ITEMS 1 to the number of lines installed **DB5**

If this bit equals to 1, the "CARRIER DETECT" signal is present on the channel's connector.

CTS: DATA ITEMS 1 to the number of lines installed **DB4**
If this bit equals 1, the "CLEAR TO SEND" signal is present on the channel's connector.

RIg: DATA ITEMS 1 to the number of lines installed **DB3**
If this bit equals 1, the "RING INDICATOR" signal is present on the channel's connector.

PAR: DATA ITEMS 1 to the number of lines installed **DB2**
If this bit equals 1, a parity or a frame error has been detected on the corresponding channel's receiver.

Ovr: DATA ITEMS 1 to the number of lines installed **DB1**
If this bit equals 1, characters have been overwritten in the receiver. This may occur when the card's CPU could not read the incoming characters quickly enough, and other characters were received that erased the preceding ones.

Brk: DATA ITEMS 1 to the number of lines installed **DB0**
For an asynchronous channel
 If this bit is set to 1, a "BREAK" signal was detected on the channel's receiver. The "BREAK" signal is low for approximately 250 ms for a short "BREAK" or 3.5 seconds for a long "BREAK" (DEC VT100 standards).
For an HDLC or LAPB channel
 If this bit is set to 1, an ABORT signal was detected on the channel's receiver.

ERROR CODE

The RSTAT command always returns an error code of 0.

IMPORTANT NOTE

Information returned for uninstalled channels is not significant. The events corresponding to bits DB0, DB1 and DB2 are memorized as soon as they are detected, and only reset to zero by the RSTAT command.

V.22. RXCNT (1Bh): Read the number of characters or frames received**OPCODE = 27 (1Bh)****DESCRIPTION****For an asynchronous channel**

This command returns the number of characters in the receive buffer. The information returned is written in the dual-ported memory's data zone.

For a synchronous channel

This command returns the number of frames in the receive buffer. The information returned is written in the dual-ported memory's data zone.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	1	0	1	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
DATA ITEM 1	Nbl	Nbl	Nbl	Nbl	Nbl	Nbl	Nbl	Nbl
DATA ITEM 2	Nbh	Nbh	Nbh	Nbh	Nbh	Nbh	Nbh	Nbh

C#: PARAMETER #1**DB6.DB5.DB4.DB3.DB2.DB1.DB0**

The <C#> bits in parameter #1 indicate the channel number. This number must be between 1 and the number of channels installed.

Nbl: DATA ITEM #1**DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

For an asynchronous channel

This data item contains the least significant bits of the number of characters in the buffer.

For a synchronous channel

This data item contains the number of frames in the buffers.

Nbh: DATA ITEM #2**DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

For an asynchronous channel

This data item contains the most significant bits of the number of characters in the buffer.

For a synchronous channel

This item equals 0.

IMPORTANT NOTE

You may also read the number of characters or frames received by directly reading the mailbox. The counters are updated in real time in the dual-ported memory. This is a faster method for reading the information, and disturbs the card less.

Each counter is a 16 bit WORD organized in least significant byte (8 bits) then most significant byte (8 bits).

For an asynchronous channel: The counter directly supplies the number of characters received.

For a synchronous channel: The counter supplies the number of frames received multiplied by 1024.

The address of the first counter is 32512 (7F00H in hexadecimal) relative to the card's segment. The second counter is at address 32514, the third counter at address 32516, and so forth.

Counter values for uninstalled lines are invariable, and always equal to zero.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 150 (96h) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Associated asynchronous buffer not allocated with the ALLOC command

V.23. RXENB (04h): Enable or disable reception**OPCODE = 4 (04h)****DESCRIPTION**

This command enables or disables reception for a channel.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	0	1	0	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	0	0	0	0	0	0	0	A/D

C#: PARAMETER #1**DB6.DB5.DB4.DB3.DB2.DB1.DB0**

The <C#> bits in parameter #1 specify the channel number. This number must be between 1 and the number of channels installed.

A/D: PARAMETER #2**DB0**

This bit will enable or disable reception on the channel in question.

DB0	Receive
1	Enabled
0	Disabled

IMPORTANT NOTE

This command is only valid if the line to be enabled has already been initialized by the VINIT or PROTO+VINIT command. Furthermore, when you enable a receive channel, the value of the "timeout", if it has been programmed, is automatically reloaded into the counter to avoid resending the STTMO command after an RXENB = OFF, RXENB = ON sequence.

Reception on an LAPB channel is automatically enabled by the VINIT command.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 133 (85h) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Corresponding asynchronous buffer not allocated by the ALLOC command

V.24. STCNT (0Bh): Define the size of receive blocks for an asynchronous channel

OPCODE = 11 (0Bh)

DESCRIPTION

This command lets the MCX card generate an interrupt every "n" characters received, or when the 1st character is received (n is defined by parameters #2 and #3), if the interrupt is authorized by the MINTR command.

Sending this command to a synchronous channel does not return an error, but has no effect.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	1	0	1	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	Cnl	Cnl	Cnl	Cnl	Cnl	Cnl	Cnl	Cnl
PARAMETER 3	Cnh	Cnh	Cnh	Cnh	Cnh	Cnh	Cnh	Cnh

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 specify the receive channel number. This number is between 1 and the number of channels installed.

Cnl: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #2 contain the least significant byte of the number of characters that must be in the buffer to generate an IT1 interrupt.

Cnh: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #3 contain the most significant byte of the number of characters that must be in the buffer to generate an IT1 interrupt.

IMPORTANT NOTE

This function is only operational if the corresponding IT1 interrupt has been selected with the MINTR command.

An interrupt is generated each time the number of characters in the receive buffer equals "modulo" the number of characters programmed by this command.

The number contained in parameters #2 and #3 must not exceed the size allocated for the associated buffer.

If the IT1 interrupt event is programmed using the MINTR command without having sent the STCNT command, an IT1 interrupt is triggered every 80 characters.

WARNING

If the programmed value is 1, the interrupt generated will not operate as usual.

In that case the interrupt means: Interrupt following the receipt of the first character. The condition will be automatically canceled when the first character is received. To re-enable it, simply send the MINTR command again with condition IT1.

This feature is useful for determining the start of a session on a communication channel.

ERROR CODE

The system returns an error code of 0 if the command executes normally. If not, error code 141 (8Dh) is returned. This command does not generate an error if it is sent to a synchronous channel, but it has no effect.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Number of characters greater than the allocated buffer size

V.25. STSIG (24h): Manually control DTR and RTS**OPCODE = 36 (24h)****DESCRIPTION**

This command lets you manually control the DTR and RTS signals.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	1	0	0	1	0	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	0	0	0	0	0	0	DTR	RTS

C#: PARAMETER #1**DB6.DB5.DB4.DB3.DB2.DB1.DB0**

The <C#> bits in parameter #1 indicates the channel number. This number must be between 1 and the number of channels installed.

RTS: PARAMETER #2**DB0**

This bit enables or disables the RTS signal (Request to Send).

RTS	State
0	Disabled
1	Enabled

Warning: the RTS bit has no effect in the following configuration:

RTS signal control mode other than “manual control by STSIG”. (See the VMODE command).

DTR: PARAMETER #2**DB1**

This bit enables or disables the DTR signal (Data Terminal Ready).

DTR	State
0	Disabled
1	Enabled

Warning: the DTR bit has no effect in the following configurations:

DTR signal control mode other than “manual control by STSIG”. (See the VMODE command).

Synchronous channel 1, 2 or 3 on an MCX-Lite/S or MCX-xx card. The DTR signal is reserved for other purposes in that configuration.

IMPORTANT NOTE

The STSIG command may be sent before the VINIT command in order to define the state of the signals following VINIT.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 160 (A0h) is returned to indicate an error in the command parameters.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Bits DB2 to DB7 in parameter #2 not null

V.26. STTMO (1Fh): Initialize the receive timeout**OPCODE = 31 (1Fh)****DESCRIPTION**

The MCX card may trigger an IT4 event interrupt (see the MINTR command) if no characters or frames are received over a given amount of time (10 seconds by default).

The STTMO command allows you to program this time between 1 and 255 seconds.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	1	1	1	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	Tmo	Tmo	Tmo	Tmo	Tmo	Tmo	Tmo	Tmo

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 specify the receive channel number. This must be a number between 1 and the number of channels installed.

Tmo: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #2 contains the value of the "timeout" for this channel, in seconds. Authorized values are between 1 and 255 seconds.

IMPORTANT NOTE

The countdown is started in the following conditions:

- An IT4 interrupt was confirmed by the MINTR command
- Reception was authorized by the RXENB command

The countdown starts over at the programmed value in the following cases:

- An RXENB ON command or an STTMO command was executed
- An IT4 interrupt was sent by the MCX card (the countdown reached 0)
- At each character or frame received.

The countdown stops in the following cases:

- The IT4 interrupt is disabled by the MINTR command
- The reception was disabled by the RXENB command.

If the countdown reaches 0, an IT4 interrupt is triggered.

Sample initialization sequence

- Optionally send the STTMO command: timeout value
- Send the MINTR command: enable IT4
- Send the RXENB: ON command; the countdown starts.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 153 (99h) is returned.

The cause of the error may be:

- Channel # not between 1 and the number of channels installed (64 max.)
- Parameter #2 equals zero.

V.27. TFREE (1Ah): Read the space available in transmission**OPCODE = 26 (1Ah)****DESCRIPTION**

This command returns the space available for the transmit channel whose number is indicated in parameter #1. The information returned is written in the dual-ported memory's data zone. If the transmit buffers are rotary, the space available is automatically updated as characters or frames are sent.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	1	1	0	1	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
DATA ITEM 1	Lnl	Lnl	Lnl	Lnl	Lnl	Lnl	Lnl	Lnl
DATA ITEM 2	Lnh	Lnh	Lnh	Lnh	Lnh	Lnh	Lnh	Lnh

C#: PARAMETER #1**DB6.DB5.DB4.DB3.DB2.DB1.DB0**

The <C#> bits in parameter #1 specify the channel number associated with the command. This number must be between and the number of channels installed.

Lnl: DATA ITEM #1**DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

For an asynchronous channel, this data item contains the least significant bits of the free space in the transmit buffer.

For a synchronous channel, this data item contains the number of transmit buffers available.

Lnh: DATA ITEM #2**DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

For an asynchronous channel, this data item contains the most significant bits of the free space in the transmit buffer.

For a synchronous channel, this data item equals 0.

IMPORTANT NOTE

You can also read the space available in a transmit buffer by directly reading the mailbox. Counters are updated in real time in the dual-ported memory. This is a faster method for reading the information, and disturbs the card less. Each counter is a WORD (16 bits) organized with the least significant byte (8 bits) followed by the most significant byte (8 bits). It indicates the space available in the TX buffer.

For an asynchronous channel: The counter directly provides the number of characters available in the transmit buffer.

For a synchronous channel: The counter contains the number of transmit buffers available, multiplied by 1024.

The address of the first counter is 32640 (7F80H in hexadecimal) relative to the card's segment. The second counter is located at address 32642, the third at address 32644, and so forth.

The state of counters for uninstalled lines is invariable.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 149 (95h) is returned to indicate an error in the command.

The cause of the error may be:

- Channel # not included between 1 and the number of channels installed (64 max.)
- Associated asynchronous buffer not allocated by the ALLOC command

V.28. VINIT (00h): Initialize communication parameters

OPCODE = 0 (00h)

DESCRIPTION

This command initializes a channel's command parameters, and makes the channel operational. After executing VINIT, you can transmit on this channel (the BTRAN command).

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	0	0	0	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	E/O	Par	Stp	Stp	BRx	BRx	BTx	BTx
PARAMETER 3	Clk	Cod	Cod	Spd	Spd	Spd	Spd	Spd
PARAMETER 4	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 5	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 6	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 7	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits indicate the channel number. This number must be between 1 and the number of channels installed.

BTx: PARAMETER #2 DB1.DB0

These 2 bits define the number of bits per character sent. The number of bits per character for a synchronous channel must equal 8.

DB1	DB0	Number of bits per character sent
0	0	5
1	0	6
0	1	7
1	1	8

BRx: PARAMETER #2 DB3.DB2

These 2 bits define the number of bits per character received. The number of bits per character for a synchronous channel must equal 8.

DB3	DB2	Number of bits per character received
0	0	5
1	0	6
0	1	7
1	1	8

If 5, 6 or 7 bits are selected, the MCX card automatically forces non-significant bits to 0.

Stp: PARAMETER #2

DB5.DB4

These 2 bits define the number of stop bits per character sent or received for an asynchronous channel. These bits are ignored for a synchronous channel.

DB5	DB4	Number of stop bits
0	0	Illegal combination
0	1	1
1	0	1.5
1	1	2

Par: PARAMETER #2

DB6

This bit enables or disables the calculation and verification of the parity bit. It must be set to 0 for a synchronous channel.

DB6	Parity
1	Enabled
0	Disabled

E/O: PARAMETER #2

DB7

This bit selects the type of parity calculation (odd or even) for an asynchronous channel. This choice is only read if parity is enabled (Bit D6 set to 1 in parameter 2). This bit is ignored for synchronous channels.

DB7	Parity
1	even
0	odd

Spd: PARAMETER #3

DB4.DB3.DB2.DB1.DB0

These bits program the speed for the channel's baud generator.

The baud generator may be used to generate transmit and receive clocks for a channel (see the PROTO command).

"SPD" FIELD	ASYNCHRONOUS CHANNEL: 250KB/S MAX			
	TX/RX SPEED IN BITS / SEC	ERROR RATE IN %		ERROR RATE IN % LITE570 16MHZ
		LITESERIAL & MCXBP 16MHZ	14MHZ	
0 (00h)	50	0	0	0.16
1 (01h)	75	-0.005	0	0.16
2 (02h)	110	0.010	0.002	0.03
3 (03h)	134.5	0.013	0.001	0.15
4 (04h)	150	0.010	0	0.16
5 (05h)	300	-0.020	0	0.16
6 (06h)	600	0.040	0	0.16
7 (07h)	1200	-0.080	0	0.16
8 (08h)	1800	-0.080	0	-0.08
9 (09h)	2000	0	0.174	0
10 (0Ah)	2400	0.160	0	-0.32
11 (0Bh)	3600	-0.080	0	-0.08
12 (0Ch)	4800	0.160	0	0.16
13 (0Dh)	7200	0.644	0	-0.08
14 (0Eh)	9600	0.160	0	0.16
15 (0Fh)	19200	0.160	0	0.16
16 (10h)	38400	0.160	0	0.16
Speed 17 is not supported by the LITE570 extension				
17 (11h)	Special speed for MCX and LITE/S			
18 (12h)	14400	-0.794	0	0.64
19 (13h)	28800	2.124	0	-0.79
20 (14h)	57600	-3.549	0	2.12
21 (15h)	64000	-2.344	2.857	-2.34
22 (16h)	115200	8.507	0	3.55
23 (17h)	125000	0	-7.8	0
24 (18h)	250000	0	-7.8	0
Speed 30 is only supported by the LITE570 extension				
30 (1Eh)				Special speed LITE570
31 (1Fh)	User speed			

"SPD" FIELD	SYNCHRONOUS CHANNEL			
	SPEED OF THE BRG IN BITS/SEC	ERROR RATE IN % LITESERIAL & MCXBP		ERROR RATE IN % LITE570
		16MHz	14MHz	16MHz
0 (00h)	134.5	-0.001	0.001	0.15
1 (01h)	134.5	-0.001	0.001	0.15
2 (02h)	134.5	-0.001	0.001	0.15
3 (03h)	134.5	-0.001	0.001	0.15
4 (04h)	150	0.001	0	0.16
5 (05h)	300	-0.001	0	0.16
6 (06h)	600	0.003	0	0.16
7 (07h)	1200	-0.005	0	0.16
8 (08h)	1800	0.010	0	-0.08
9 (09h)	2000	0	0.011	0
10 (0Ah)	2400	0.010	0	0.16
11 (0Bh)	3600	0.010	0	-0.08
12 (0Ch)	4800	-0.020	0	0.16
13 (0Dh)	7200	0.010	0	-0.08
14 (0Eh)	9600	0.040	0	0.16
15 (0Fh)	19200	-0.080	0	0.16
16 (10h)	38400	0.160	0	-0.32
Speed 17 is not supported by the LITE570 extension				
17 (11h)	Special speed MCX and LITE/S			
18 (12h)	14400	-0.080	0	-0.08
19 (13h)	28800	-0.080	0	-0.08
20 (14h)	57600	-0.080	0	-0.08
21 (15h)	64000	0	0.174	0
22 (16h)	115200	0.644	0	0.6
23 (17h)	125000	0	-0.03	0
Speeds 24, 25, 26 are not supported on channels greater than 3 on an MCXBP extension				
24 (18h)	250000	0	1.694	0
25 (19h)	500000	0	-1.696	0
26 (1Ah)	1000000	0	5.326	0
Speeds 27, 28, 29 and 30 are only supported by the LITE570 extension				
27 (1Bh)	2000000			0
28 (1Ch)	4000000			0
29 (1Dh)	Reserved			
30 (1Eh)				Special speed LITE570
31 (1Fh)				User speed

SPEED 17, RECOGNIZED BY THE MCXBP AND LITESERIAL EXTENSIONS, allows you to program a special speed by directly initializing the WR12 and WR13 registers¹³ in the SCC 85C30 baud generator by computing a counting factor called TC.

The TC counting factor is linked to the speed as follows:

Transmission	Encoding	Clock	TC Formula	Max. Theoretical speed at F = 16 MHz
Asynchronous	NRZ	None	$TC = \frac{F}{32 \times V} - 2$	250 Kbps
Synchronized asynchronous	NRZ	Internal	$TC = \frac{F}{2 \times V} - 2$	4 Mbps
Synchronous	All	Internal	$TC = \frac{F}{2 \times V} - 2$	4 Mbps
Synchronous	NRZi	Extracted from received data	$TC = \frac{F}{64 \times V} - 2$	125 Kbps
Synchronous	FM0/FM1	Extracted from received data	$TC = \frac{F}{32 \times V} - 2$	250 Kbps

The communication speed S is expressed in bps.

The oscillator frequency F is expressed in Hertz (16×10^6 or 14.7456×10^6)

If the operation returns a value that is not an integer, you should round off to the nearest number. If the difference between the integer and the decimal part of the number is too large, the transmission may contain a non-negligible percentage of errors.

WR12: PARAMETER #4 DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #4 contains the least significant byte of the TC counting factor (image of the WR12 register).

WR13: PARAMETER #5 DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #5 contains the most significant byte of the TC counting factor (image of the WR13 register).

Example: compute the TMC in asynchronous mode with no clocks for V= 38400 bits/s with F = 16MHZ

$$TMC = \frac{16000000}{32 \cdot 38400} = 11.02$$

Rounding off to the nearest integer gives TMC=11

$$\text{The actual speed is } \frac{16000000}{32 \cdot (11 + 2)} = 38461 \text{ bits/s, the error is } \frac{38461,51 - 38400}{38400} = 0.16\%$$

¹³ These registers are described in the "Am8530/Am85C30 Technical Manual", available on request from companies that distribute AMD communication products.

SPEED 30, RECOGNIZED BY THE LITE570 EXTENSION, allows you to program a special speed by directly initializing the TMC, RXS, TXS, MD1 and MD2 registers¹⁴ in the SCA HITACHI HD64570 baud generator by computing 4 values: TMC, CM, BR and DPLL.

These 4 values are related to the speed as follows:

Transmission	Encoding	Clock	V Formula	Max. Theoretical speed at F = 16 MHz
Asynchronous	NRZ	None	$V = \frac{F}{CM \cdot TMC \cdot 2^{BR}}$ with CM = 16, 32, 64	500 Kbps
Synchronized asynchronous	NRZ	Internal	$V = \frac{F}{CM \cdot TMC \cdot 2^{BR}}$ with CM = 1	5.3 Mbps
Synchronous	All	Internal	$V = \frac{F}{TMC \cdot 2^{BR}}$	8 Mbps
Synchronous	All	Extracted from received data	$V = \frac{F}{DPLL \cdot TMC \cdot 2^{BR}}$	2 Mbps

V is the communication speed expressed in bits/s

F is the oscillator frequency, expressed in Hertz ($16 \cdot 10^6$)

TMC varies between 1 and 256, Br varies between 0 and 9, CM equals 1, 16, 32 or 64, DPLL equals 8, 16 or 32.

Important note:

If BR equals 0, and TMC is greater than 2, then the cyclical ratio of the outgoing clocks is not equal to 50%.

TMC: PARAMETER #4

DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #4 sets the value of TMC (image of the TMC register).

Authorized values are between 1 and 255.

A value of 0 sets TMC to 256.

¹⁴ These registers are described in the "HD64570 SCA Serial Communication Adapter User's Manual", available on request from companies that distribute HITACHI communication products.

PARAMETER #5 CM1 CM0 DPLL1 DPLL0 BR3 BR2 BR1 BR0
 Parameter #5 sets the values for CM, DPLL and BR associated with registers MD1, MD2, RXS and TXS.

BR3	BR2	BR1	BR0	BR
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
Other combinations not valid				

CM1	CM0	CM
0	0	1
0	1	16
1	0	32
1	1	64
Other combinations not valid		

DPLL1	DPLL0	DPLL
0	0	8
0	1	16
1	0	32
Other combinations not valid		

CM1 and CM0 are the images of the BRATE1 and BRATE0 bits respectively in the MD1 register.

DPLL1 and DPLL0 are the images of the DRATE1 and DRATE0 bits respectively in the MD2 register.

BR3, BR2, BR1 and BR0 are the images of bits TXBR3/RXBR3, TXBR2/RXBR2, TXBR1/RXBR1 and TXBR0/RXBR0 in the TXS/RXS registers, respectively.

Example: compute TMC, CM and BR in asynchronous mode with no clock for V= 9600 bits/s and F = 16MHZ

In order to obtain maximum precision, you should try to obtain the highest possible value of TMC, with the product of $2^{Br} \cdot CM$ as small as possible.

This gives Br = 0 and CM = 16

$$TMC = \frac{16000000}{16 \cdot 9600 \cdot 2^0} = 104.16$$

Rounding off to the nearest integer gives TMC= 104

The actual speed is $\frac{16000000}{16 \cdot 104} = 9615.38$ bits/s, the error is $\frac{9615,38 - 9600}{9600} = 0.16\%$

SPEED 31, RECOGNIZED BY ALL EXTENSIONS, allows you to define a given speed by directly providing the desired speed encoded in an unsigned double word (32 bits) in parameters #4, #5, #6 and #7:

SPD: PARAMETER #4 SPD7.SPD6.SPD5.SPD4.SPD3.SPD2.SPD1.SPD0
Parameter #4 contains the least significant byte of the least significant word of the speed.

SPD: PARAMETER #5 SPD15.SPD14.SPD13.SPD12.SPD11.SPD10.SPD9.SPD8
Parameter #5 contains the most significant byte of the least significant word of the speed.

SPD: PARAMETER #6 SPD23.SPD22.SPD21.SPD20.SPD19.SPD18.SPD17.SPD16
Parameter #6 contains the least significant byte of the most significant word of the speed.

SPD: PARAMETER #7 SPD31.SPD30.SPD29.SPD28.SPD27.SPD26.SPD25.SPD24
Parameter #7 contains the most significant byte of the most significant word of the speed.

Err: PARAMETER #8 Err7 Err6 Err5 Err4 Err3 Err2 Err1 Err0
Parameter #8 contains the error percentage tolerated between the speed indicated in parameters #4, #5, #6 and #7 and speed actually programmed. This parameter is expressed in ‰.
All values are accepted (from 0‰. to 255‰.).

If the actual speed exceeds the accepted tolerance, the VINIT command simply returns an error with code 129 (decimal).

MAXIMUM BINARY THROUGHPUTS SUPPORTED BY THE MCX MULTIPROTOCOL SOFTWARE

- * **In synchronous mode with hardwired clocks:**
 - 1 Mbps on channels on channels 1, 2 and 3 of an MCXBP extension
 - 125 Kbps on channels 4 and higher of an MCXBP extension
 - 1 Mbps on all channels of a LITESERIAL extension
 - 4 Mbps on all channels of a LITE570 extension

- * **In synchronous mode with no hardwired clocks:**
 - 250 Kbps on all channels of a LITESERIAL extension in NRZ or NRZi mode
 - 125 Kbps on all channels of a LITESERIAL extension in FM0 or FM1 mode
 - 250 Kbps on channels 1, 2 and 3 of an MCXBP extension in NRZ or NRZi mode
 - 125 Kbps on channels 4 and higher of an MCXBP extension in all modes
 - 2 Mbps on all channels of a LITE570 extension

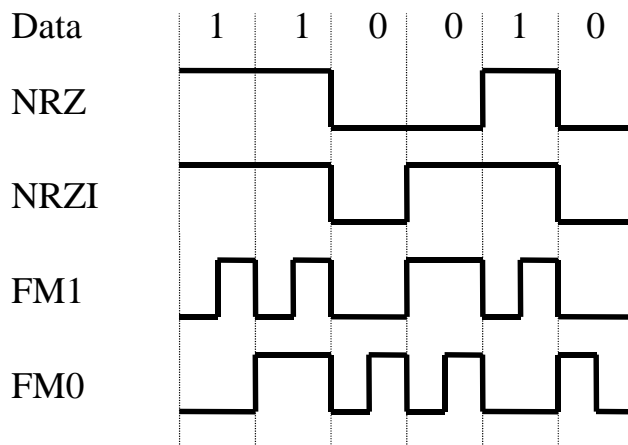
- * **In asynchronous mode or in synchronized asynchronous mode:**
 - 250 Kbps on all extensions.

Cod: PARAMETER #3

DB6.DB5

These bits define the type of coding used to receive and transmit characters, as shown in the table below:

DB6	DB5	Coding
0	0	NRZ (No-Return to Zero)
0	1	NRZ-I (No-Return to Zero Inverted)
1	0	FM1 (Biphase Mark)
1	1	FM0 (Biphase Space)



Clk: PARAMETER #3

DB7

This bit in parameter #3 selects the oscillator used by the 8 channels on the MCX-BP back panel or the 2 channels on the Lite Serial extension.

DB7	Oscillator type
0	16 MHz
1	14.7456 MHz

Bit DB7 must be set to 0 for a LITE570 extension.

NOTE

If all the bits in parameter 2 equal 0, then the default initialization is selected. Parameters 3, 4 and 5 are ignored. The default values selected are:

For an asynchronous channel:

- | | |
|--------------------------------------|-------------|
| • Length of characters sent/received | 8 bits |
| • Number of stop bits per character | 1 bit |
| • Parity | None |
| • Transmission/reception speed | 9600 bits/s |
| • Coding | NRZ |

For a synchronous channel:

- | | |
|--------------------------------------|-------------|
| • Length of characters sent/received | 8 bits |
| • Parity | None |
| • Transmission/reception speed | 9600 bits/s |
| • Coding | NRZ |

IMPORTANT NOTE CONCERNING THE STATE OF THE RTS AND DTR SIGNALS AFTER THE VINIT COMMAND:

If no protocol was defined for handling the RTS and DTR signals (see the VMODE command), the **first** VINIT command sent following a PROTO command enables the RTS and DTR¹⁵ signals unless an STSIG command was sent before VINIT. Otherwise, the VINIT command does not change the state of these signals.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, the code 129 (decimal) is returned to indicate an error in the parameters.

The cause of the error may be:

- Channel # not between 1 and the number of channels installed (max=64)
- Forbidden combination for Stp (00) parameter 2 for asynchronous channels
- Forbidden combination for Cod (xx) parameter 3 for asynchronous channels
- Forbidden combination for BRx (xx) or BTx (xx) parameter 2 for synchronous channels
- Forbidden combination for Par (x) parameter 2 for synchronous channels
- Selected speed forbidden
- 14.7456 MHz Oscillator selected for LITE570 extension
- Error rate too high for speed 31

¹⁵ The DTR signal is not available on synchronous channels 1 and 2 of a LITE-SERIAL extension, nor on the synchronous channels 1, 2 and 3 of an MCX-BP extension.

V.29. VMODE (03h): Initialize flow control parameters

OPCODE = 3 (03h)

DESCRIPTION

This command initializes a channel's flow control parameters.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	0	0	1	1
PARAMETER 1	Typ	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	Cod	Cod	Cod	Cod	Cod	Cod	Cod	Cod
PARAMETER 3	0	0	CtSSens	DcdSens	RiSens	DsrSens	NullStrip	ErrRepl
PARAMETER 4	ErrC	ErrC	ErrC	ErrC	ErrC	ErrC	ErrC	ErrC
PARAMETER 5	0	0	iXon	iXon	iDTR	iDTR	iRTS	iRTS
PARAMETER 6	0	0	oXon	oXon	oRi	oDcd	oDsr	oCts
PARAMETER 7	Xon	Xon	Xon	Xon	Xon	Xon	Xon	Xon
PARAMETER 8	Xoff	Xoff	Xoff	Xoff	Xoff	Xoff	Xoff	Xoff
PARAMETER 9	offL	offL	offL	offL	offL	offL	offL	offL
PARAMETER 10	onLl	onLl	onLl	onLl	onLl	onLl	onLl	onLl
PARAMETER 11	onLh	onLh	onLh	onLh	onLh	onLh	onLh	onLh

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

The <C#> bits in parameter #1 define the channel number. This must be a number between 1 and the number of channels installed.

Typ: PARAMETER #1 DB7

The "Typ" bit in parameter #1 must be set to 1.

Cod: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

The 8 "Cod" bits in parameter #2 must be set to 0.

ErrRepl: PARAMETER #3 DB0

The "ErrRepl" bit in parameter #3 specifies if the characters received with a frame or parity error should be replaced by the character defined in parameter #4.

DB0	Replace bad character
0	No
1	Yes replacement character: parameter #4

The ErrRepl bit must be set to 0 for a synchronous channel.

NullStrip: PARAMETER #3**DB1**

The “NullStrip” bit in parameter #3 specifies if null characters received should be ignored.

DB1	Ignore null character
0	No
1	Yes

The NullStrip bit must be set to 0 for synchronous channels.

DsrSens: PARAMETER #3**DB2**

The “DsrSens” bit in parameter #3 specifies if characters received when the DSR signal is low should be ignored.

DB2	Ignore characters received if DSR low
0	No
1	Yes

Warning: the DSR signal is not available on the MCX-BP and LITE-SERIAL extensions; therefore the “DsrSens” bit must be set to 0.

RiSens: PARAMETER #3**DB3**

The “RiSens” bit in parameter #3 specifies if characters received when the RI signal is low should be ignored.

DB2	Ignore characters received if RI low
0	No
1	Yes

DcdSens: PARAMETER #3**DB4**

The “DcdSens” bit in parameter #3 specifies if characters received when the DCD signal is low should be ignored.

DB4	Ignore characters received if DCD low
0	No
1	Yes

CtsSens: PARAMETER #3**DB5**

The “CtsSens” bit in parameter #3 specifies if characters received when the CTS signal is low should be ignored.

DB5	Ignore characters received if CTS low
0	No
1	Yes

Important note concerning the RiSens, DcdSens, CtsSens and DsrSens bits in synchronous HDLC mode:

If a control signal prevents low reception when receiving an HDLC frame, the frame is considered as invalid when it is read by the RDBUF command. (terminated by an ABORT sequence).

ErrC: PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

The 8 “ErrC” bits in parameter #4 define the substitution character for characters received with a parity or frame error. See the ErrRepl bit in parameter #3.

iRTS: PARAMETER #5 DB1.DB0

The 2 iRTS bits in parameter #5 define the control mode for the RTS signal.

DB1	DB0	Control the RTS signal	In asynchronous mode	In synchronous mode
0	0	Manual control by STSIG ¹⁶	Supported	Supported
0	1	Input flow control	Supported	Not supported
1	0	Toggle during transmission	Not Supported	Supported
1	1	Forbidden combination		

If incoming RTS flow control is enabled in asynchronous mode, the card disables the RTS signal as soon as the number of characters contained in the receive buffer reaches the buffer size defined in the ALLOC command, minus the limit defined in parameter #9. The RTS signal is re-enabled as soon as the number of characters contained in the receive buffer is less than or equal to the limit defined in parameters #10 and #11.

iDTR: PARAMETER #5 DB3.DB2

The 2 iDTR bits in parameter #5 define the control mode for the DTR signal.

DB3	DB2	Control of the DTR signal	In asynchronous mode	In synchronous mode
0	0	Manual control by STSIG ¹⁶	Supported	Supported
0	1	Input flow control	Supported	Not supported
1	0	Forbidden combination		
1	1	Forbidden combination		

If incoming DTR flow control is enabled in asynchronous mode, the card disables the DTR signal as soon as the number of characters contained in the receive buffer reaches the buffer size defined in the ALLOC command, minus the limit defined in parameter #9. The DTR signal is re-enabled as soon as the number of characters contained in the receive buffer is less than or equal to the limit defined in parameters #10 and #11.

¹⁶ See the STSIG command page V-108.

iXon: PARAMETER #5**DB5.DB4**

The 2 iXon bits in parameter #5 define if incoming Xon/Xoff flow control is enabled.

DB5	DB4	Xon/Xoff flow control on input
0	0	No
0	1	Yes without blocking transmission when sending the Xoff character
1	0	Yes and blocks transmission when sending the Xoff character
1	1	Forbidden combination

If incoming Xon/Xoff flow control is enabled on an asynchronous channel, the card sends an Xoff character as soon as the number of characters contained in the receive buffer reaches the buffer size defined in the ALLOC command, minus the limit defined in parameter #8. An Xon character is sent as soon as the number of characters contained in the receive buffer is less than or equal to the limit defined in parameters #9 and #10. You can also block transmission after sending an Xoff character. This allows you to adapt to devices that interpret the first character received after an Xoff as an Xon.

The Xon and Xoff characters must be defined in parameters #7 and #8.

Xon/Xoff incoming flow control is not supported in synchronous mode; therefore the 2 iXon bits (DB5 and DB4) must be set to 0.

oCts: PARAMETER #6**DB0**

This bit in parameter #6 indicates if CTS outgoing flow control is enabled.

DB0	CTS flow control on output
0	No
1	Yes

If outgoing CTS flow control is enabled, character transmission is blocked if the CTS signal is low, and is authorized if it is high.

If outgoing CTS flow control is disabled, character transmission is possible whatever the state of CTS.

oDsr: PARAMETER #6**DB1**

This bit in parameter #6 indicates if outgoing DSR flow control is enabled.

DB1	DSR flow control on output
0	No
1	Yes (not implemented)

If outgoing DSR flow control is enabled, character transmission is blocked if the DSR signal is low, and is authorized if it is high.

If outgoing DSR flow control is disabled, character transmission is possible whatever the state of DSR.

Warning: the DSR signal is not available on the MCX-BP and LITE-SERIAL extensions; therefore the “oDsr” bit must be set to 0.

oDcd: PARAMETER #6**DB2**

This bit in parameter #6 indicates if the outgoing DCD flow control is enabled.

DB2	Flow control DCD on output
0	No
1	Yes

If outgoing DCD flow control is enabled, character transmission is blocked if the DCD signal is low, and is authorized if it is high.

If outgoing DCD flow control is disabled, character transmission is possible whatever the state of DCD.

oRi: PARAMETER #6**DB3**

This bit in parameter #6 indicates if Ring Indicator outgoing flow control is enabled.

DB2	RI flow control on output
0	No
1	Yes

If outgoing RI flow control is enabled, character transmission is blocked if the RI signal is low, and is authorized if it is high.

If outgoing RI flow control is disabled, character transmission is possible whatever the state of RI.

Warning: the Lite570 extension does not support RI flow control; therefore the oRI bit must be set to 0.

Important note concerning flow control on output in synchronous HDLC mode:

If a signal that manages flow control on output goes low when transmitting a frame, the frame is terminated by an ABORT sequence.

oXon: PARAMETER #6**DB5.DB4**

These bits in parameter #6 define if outgoing Xon/Xoff flow control is enabled.

DB5	DB4	Xon/Xoff flow control on output
0	0	No
0	1	Yes, and transmission is unblocked upon receipt of an Xon character
1	0	Yes, and transmission is unblocked upon receipt of any character
1	1	Invalid combination

If Xon/Xoff flow control on output is enabled, the receipt of an Xoff character blocks the transmission of all characters. In “unblock transmission upon receipt of an Xon character” mode, the receipt of an Xon character re-enables transmission. In “unblock transmission upon receipt of any character” mode, any character received after an Xoff re-enables transmission.

The Xon and Xoff characters must be defined in parameters #7 and #8.

Xon/Xoff flow control on output is not supported in synchronous mode; therefore the 2 oXon bits (DB5 and DB4) must be set to 0.

Xon: PARAMETER #7**DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

These bits in parameter #7 define the Xon character used in Xon/Xoff flow control on input and output. This parameter is ignored if Xon/Xoff flow control has not been programmed.

The most commonly-used character is the ASCII DC1 character (11 in hexadecimal).

Xoff: PARAMETER #8**DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

These bits in parameter #8 define the Xoff character used in the Xon/Xoff flow control on input and on output. This parameter is ignored if Xon/Xoff flow control has not been programmed.

The most commonly used character is ASCII character DC3 (13 in hexadecimal).

offL: PARAMETER #9

DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #9 define the limit above which flow control on input blocks the transmitter connected to the card. A limit of n ($0 \leq n \leq 255$) bytes indicates that the transmitter will be blocked when the number of characters contained in the receive buffer reaches the limit $b-n$ (where b is the size of the receive buffer).

This parameter is ignored if flow control has not been programmed.

onLl: PARAMETER #10

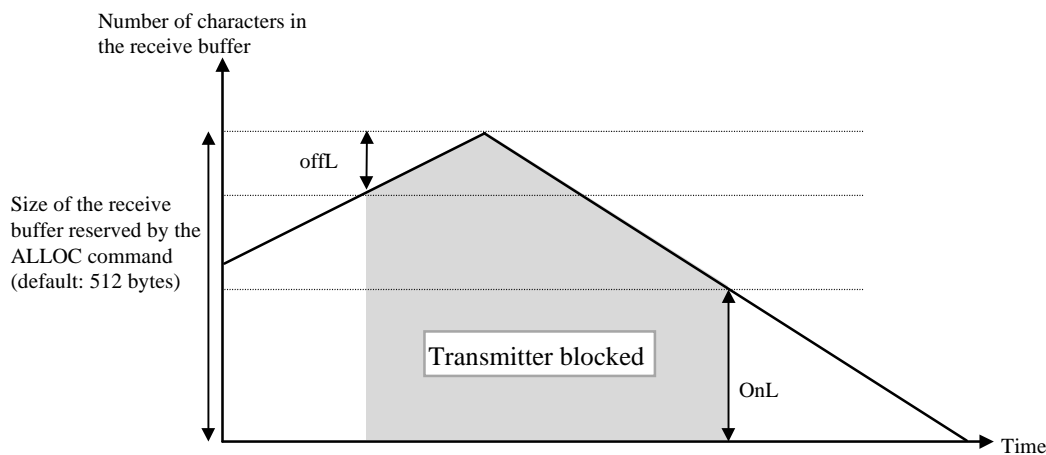
DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

onLh: PARAMETER #11

DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameters #10 and #11 define the least significant byte and the most significant byte of the limit above which flow control on input unblocks the transmitter connected to the card. A limit of m ($0 \leq m \leq 65535$) bytes indicates that the transmitter will be unblocked when the number of characters contained in the receive buffer reaches the limit m .

This parameter is ignored if flow control has not been programmed.

**IMPORTANT NOTE**

- ◆ To ensure compatibility with existing applications, the VMODE command supports the format documented in the basic software User's Manual (the Typ bit in parameter #1 at 0).
- ◆ If no incoming flow control is enabled on an asynchronous channel, an internal mechanism prevents characters from entering the receive buffer if it is full. In this case, received characters are lost. An IT5 interrupt event (receive error) can be generated for each lost character (see the MINTR command).
- ◆ Outgoing Xon/Xoff flow control on an asynchronous channel only works if the receipt of characters has been confirmed by the RXENB command.
- ◆ If at least one type of incoming flow control has been programmed, the offL (parameter #9) and onL (parameters #10 and #11) limits must meet the following condition: $\text{offL} + \text{onL} < \text{Size of the receive buffer}$. Therefore you must define the size of the receive buffer (see the ALLOC command) before configuring flow control. If an ALLOC command is nevertheless sent after a VMODE command, the ALLOC command initializes the offL limit to 20 and onL to 128.
- ◆ The DSR signal is not available on the MCX-BP and LITE-SERIAL extensions.

ERROR CODE

If the command executes normally, an error code of 0 is returned. If not, code 132 (decimal) is returned to indicate an error in the parameters.

On a synchronous channel, the command returns error code 0, but has no effect.

The cause of the error may be:

- Channel # not between 1 and the number of channels installed (max=64)
- Parameter #2 not null
- Parameter #3 invalid
- Parameter #5 invalid
- Parameter #6 invalid
- Receive or transmit buffer not reserved by the ALLOC command
- Xon character (parameter #7) identical to the Xoff character (parameter #8)
- Error in offL and onL limits: $\text{offL} + \text{onL} \geq \text{Size of the receive buffer}$

THIS PAGE INTENTIONALLY LEFT BLANK

VI. The Command Interpreter in X25 mode

VI.1. Location of the X25 counters in the data zone

Address	Perm	Name	Field description	
0064	RW	DATA 1	Data zone, 1 st byte	DATA
0065	RW	DATA 2	Data zone, 2 nd byte	
0066	RW	DATA 3	Data zone, 3 rd byte	
BaseAd-1	RW	DATA	Data zone, last byte	ZONE
BaseAd ¹⁷ at 7EFF	RO	RX Cnt x25 k TX Cnt x25 k	X25 counters	X25 COUNTERS ZONE

You can use the “RX Cnt X25 k” and “TX Cnt X25 k” counters to determine the state of the transmit and receive windows for a logical channel on an X25 data link at a given moment in time.

The RX Cnt X25 k counter is increased by 1024 for each packet received on the kth logical channel and decreased by 1024 as soon as a packet is read (see the RDBUF command) on that logical channel. Therefore this counter equals 0 when the receive window is empty, and “1024xSize of the window” when the window is full.

The TX Cnt X25 k counter is decreased by 1024 each time a packet is sent on the kth logical channel (see the BTRAN command) and increased by 1024 as soon as a packet is acknowledged on that logical channel. Therefore this counter equals 0 when the transmit window is full, and “1024xSize of the window” when the window is empty.

The programmer must define the base address for these counters (called BaseAd) using the PROTO command. The BaseAd address must be part of the data zone in the mailbox, and must be in the upper part of that zone so as to leave as much room as possible for exchanging asynchronous data packets sent or received.

The suggested formula is the following:

$$\text{BaseAd} = 8000\text{h} - 100\text{h} - \text{CxN} \times 4$$

where:

- 8000h Size of the mailbox
- 100h Space reserved by the LAPB counters
- C Physical channel number indexed starting from 1
- N Maximum number of LCs on the channel that supports the most LCs
- 4 Number of bytes per LC

The counters for an X25 channel occupy the memory between BaseAd and BaseAd+4N-1. This formula makes it possible to store the counters for each channel configured in X25 at the end of the mailbox’s data zone and to reserve 4n bytes of memory per channel.

The organization of these counters in the mailbox’s data zone depends on the number of channels configured in X25, and the number of logical channels supported per channel.

¹⁷ The base address of the X25 counters is defined by the programmer in the PROTO command.

For example, we shall examine the case of a card programmed with two X25 channels in order to explain how and where to place these counters.

Channel 1 X25, 64 LC

Channel 2 X25, 128 LC

Channels 3, 4 ... (immaterial)

The channel that supports the most LCs is channel 2, therefore $N=128$.

The base address for channel 1 counters is: $8000h-100h-1 \times 128 \times 4 = 7D00h$

The base address for channel 2 counters is: $8000h-100h-2 \times 128 \times 4 = 7B00h$

These counters reside in 1024 bytes the upper part of the mailbox. Note that this memory space could have been reduced to $128 \times 4 + 64 \times 4 = 768$ bytes.

7B00h	RO	DATA 31233	Receive counter channel 2 LC #1	X25
7B02h	RO	DATA 31235	Receive counter channel 2 LC #2	
7B04h	RO	DATA 31237	Receive counter channel 2 LC #3	
7B7Ch	RO	DATA 31357	Receive counter channel 2 LC #63	
7B7Eh	RO	DATA 31359	Receive counter channel 2 LC #64	
7B80h	RO	DATA 31361	Transmit counter channel 2 LC #1	
7B82h	RO	DATA 31363	Transmit counter channel 2 LC #2	
7B84h	RO	DATA 31365	Transmit counter channel 2 LC #3	
7BFCh	RO	DATA 31485	Transmit counter channel 2 LC #63	
7BFEh	RO	DATA 31487	Transmit counter channel 2 LC #64	
7C00h to 7CFFh	RW		Zone not used	COUNTER
7D00h	RO	DATA 31745	Receive counter channel 1 LC #1	
7D02h	RO	DATA 31747	Receive counter channel 1 LC #2	
7D04h	RO	DATA 31749	Receive counter channel 1 LC #3	
7DFCh	RO	DATA 31997	Receive counter channel 1 LC #127	
7DFEh	RO	DATA 31999	Receive counter channel 1 LC #128	
7E00h	RO	DATA 32001	Transmit counter channel 1 LC #1	
7E02h	RO	DATA 32003	Transmit counter channel 1 LC #2	
7E04h	RO	DATA 32005	Transmit counter channel 1 LC #3	
7EFCh	RO	DATA 32253	Transmit counter channel 1 LC #127	
7EFEh	RO	DATA 32255	Transmit counter channel 1 LC #128	ZONE

Warning: zones allocated to receive the RX and TX counters associated with the channels configured as X25 channels must not be overwritten by data.

VI.2. X25 Commands

Only the following commands documented in the previous chapter may be applied to an X25 channel:

GOADR, HNGUP, MBOOT, NOPER, PRCTL, PROTO, RELRP, RINIT, RMEMO, RSMDE, RSTAT, RXENB, STSIG, VINIT, VMODE.

The ALLOC, BDELE, BPARM, BREAK, CHDEF, CLRRX, RXCNT, STTMO and TFREE commands return error code 189 (unauthorized command).

The STCNT command returns error code 0, but has no effect.

The general card management commands:

GOADR, MBOOT, NOPER, RELRP, RINIT, RMEMO,

and the following commands:

HNGUP, PROTO, RSMDE, RSTAT, RXENB, STSIG, VINIT, VMODE.

are not covered in this chapter. They apply to X25 channels directly, with no additional parameters.

Only the following commands requiring new parameters, or parameters specific to the X25 protocol, are described in this chapter:

- **MINTR**
- **PRCTL**
- **BTRAN**
- **RDBUF**

VI.2.1.MINTR (0Ch): X25 “event” interrupt conditions**OPCODE = 12 (0Ch)****DESCRIPTION**

This command enables or masks the “event” interrupt conditions for an X25 data link. Warning: this command concerns all the X25 data link’s logical channels.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	1	1	0	0
PARAMETER 1	Mde	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	IT7	IT6	0	0	0	IT2	0	0

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #1 indicate the X25 data link’s number. This number must be between 1 and the number of lines installed.

Mde: PARAMETER #1 DB7

This parameter changes the meaning of IT2, depending on whether it is set to 1 or to 0.

ITn: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #2 specify the events that trigger an interrupt on the PC’s bus.

The table below describes how these bits correspond to the various events.

IT #	Event that triggers the interrupt	Remarks
IT0	Undefined	
IT1	Undefined	
IT2 (Mde=0)	Undefined	
IT2 (Mde=1)	The size of the receive window switches from 0 to 1	
IT3	Undefined	
IT4	Undefined	
IT5	Undefined	
IT6	X25 event: * Call packet received * Waiting for call timeout * Outgoing call confirmed * Outgoing call timeout * Clear confirmed * Clear timeout * Diagnostic packet received * Protocol error * Clear requested by the DCE * Reset requested by the DCE * Reset confirmed * Reset timeout * Follow-on requested by the DCE * Follow-on confirmed * Follow-on timeout expired * DCE interrupt * Interrupt confirmed * Data link out of service * Data link in service * Data link reset in progress	
IT7	The transmit window has been emptied	

NOTE

The card's operating mode for interrupts may be totally different for each channel. Furthermore, you can select several conditions for a single channel. Nevertheless, the new group of conditions replaces any previous condition(s).

ERROR CODE

If the command executed normally, error code 0 is returned. If not, code 142 (8Eh) is returned to indicate an error in the parameters.

The cause of the error is:

- Parameter #1: Invalid channel #

VI.2.2.PRCTL (30h):X25 Control Procedure

OPCODE = 48 (30h)

DESCRIPTION

This command is used:

- to send a call packet,
- to filter incoming calls,
- to free a logical channel,
- to reinitialize a CVC,
- and to connect or disconnect the link at the LAPB data link level.

This command contains an “opcode” parameter (PARAMETER #7) that specifies the action to perform. Each opcode requires a certain number of its own specific parameters.

Some X25 parameters written in italics are not used by the card in the current version of the program. Nevertheless, these parameters must be initialized as indicated.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	1	0	1	1	1	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	0	0	0	0	0	0	0	0
PARAMETER 7	OP	OP	OP	OP	OP	OP	OP	OP

C#: PARAMETER #1

DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #1 indicate the number of the X25 data link, whatever the opcode defined in parameter #7. This number must be between 1 and the number of lines installed.

PARAMETER #2

DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #2 must be set to 0 whatever the opcode defined in parameter #7.

Parameter #7 contains the opcode associated with the PRCTL command. Each code has a list of specific parameters (parameters #3 to #6 and #8 and above) described below.

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	CODE OPERATION
0	0	0	0	0	0	0	0	LAPLNKUP (LAPB Connection)
0	0	0	0	0	0	0	1	LAPLNKDN (LAPB disconnection)
0	0	0	0	1	0	1	0	LAPSTATE (State of the LAPB link)
0	0	0	0	0	0	1	0	X25LISTN (Filtering of incoming calls)
0	0	0	0	0	0	1	1	X25CALL (Call request)
0	0	0	0	0	1	0	0	X25LIB (Clear request)
0	0	0	0	0	1	0	1	X25RESET (Reset request)
0	0	0	0	0	1	1	0	X25CFCALL (Call confirmation)
0	0	0	0	0	1	1	1	X25CFLIB (Clear confirmation)
0	0	0	0	1	0	0	0	X25CFINT (Interruption confirmation)
0	0	0	0	1	0	0	1	X25GETPKT (Read X25 packet)

VI.2.2.1. *PRCTL + LAPLNKUP opcode (00h) command*

See previous chapter

VI.2.2.2. *PRCTL + LAPLNKDN opcode (01h) command*

See previous chapter

VI.2.2.3. *PRCTL + LAPSTATE opcode (0Ah) command*

See previous chapter

VI.2.2.4.PRCTL + X25LISTN opcode (02h) command

The MCX DTE cannot accept calls from a remote DTE without the user's preliminary agreement. Opcode X25LISTN allows users to employ "wildcard" to define caller addresses, as well as the addresses called that are authorized on the data link when receive an incoming call packet is received (warning: all the link's switched logical channels are concerned by these "wildcards").

Wildcards are made up of 2 ASCII-encoded numerical character strings which may optionally include the "?" wildcard (to substitute a single numerical character) or the "*" wildcard (to substitute any numerical character), separated by any character, **and a 32-bit identifier**.

General wildcard format:

- * A 32 bit identifier that is not already used by another wildcard
- * An ASCII string "STRING1/STRING2" (the character '/' is used here as the delimiting character).
 - "STRING1" is the wildcard associated with the address of the MCX DTE. It is used to define authorized addresses on the data link.
 - "STRING2" is the wildcard associated with the calling address. It is used to define authorized initiator addresses on the data link.

As soon as an incoming call packet is received on a logical channel, the MCX card looks for a wildcard that matches the calling address and the called address contained in this call packet. If nothing is found, the MCX card refuses the call, and starts a procedure to free the call. Otherwise, the call may be confirmed by the MCX card or by the PC.

Virtual Circuit Set-up Diagram: Receiving an incoming call packet that matches a wildcard

The connection is set up as follows:

The network transmits an “incoming call” packet to the responding MCX DTE on the free logical channel with the lowest number.

The MCX card sends an IT6 interrupt to the PC with the “*Call received*” event code.

If automatic call confirmation was programmed, the MCX card indicates that it accepts the call, and sends a “call accepted” packet to the network.

If not, the PC can read the call data, facilities (see the PRCTL+X25GETPKT command) and reply

by sending a call confirmation (PRCTL+X25CFCALL procedure)
or by initiating a clear procedure (PRCTL+X25LIB procedure).

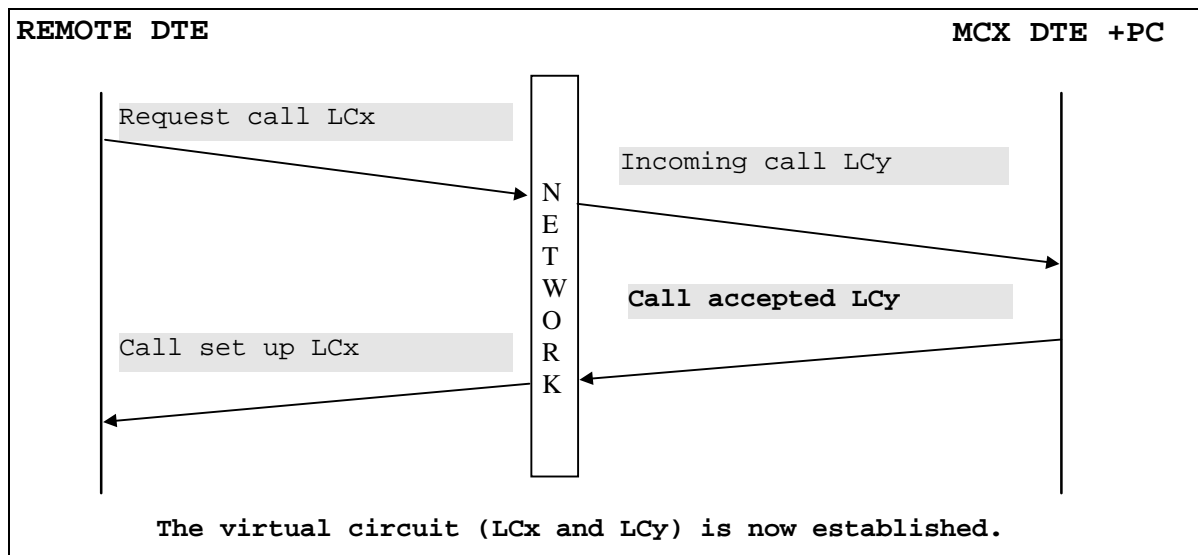


Figure VI-1

Virtual Circuit Set-up Diagram: Receiving an incoming call packet that does not match a wildcard

The connection is set up as follows:

If a call packet is received and it does not match any defined wildcards, the MCX card automatically initiates a clear procedure.

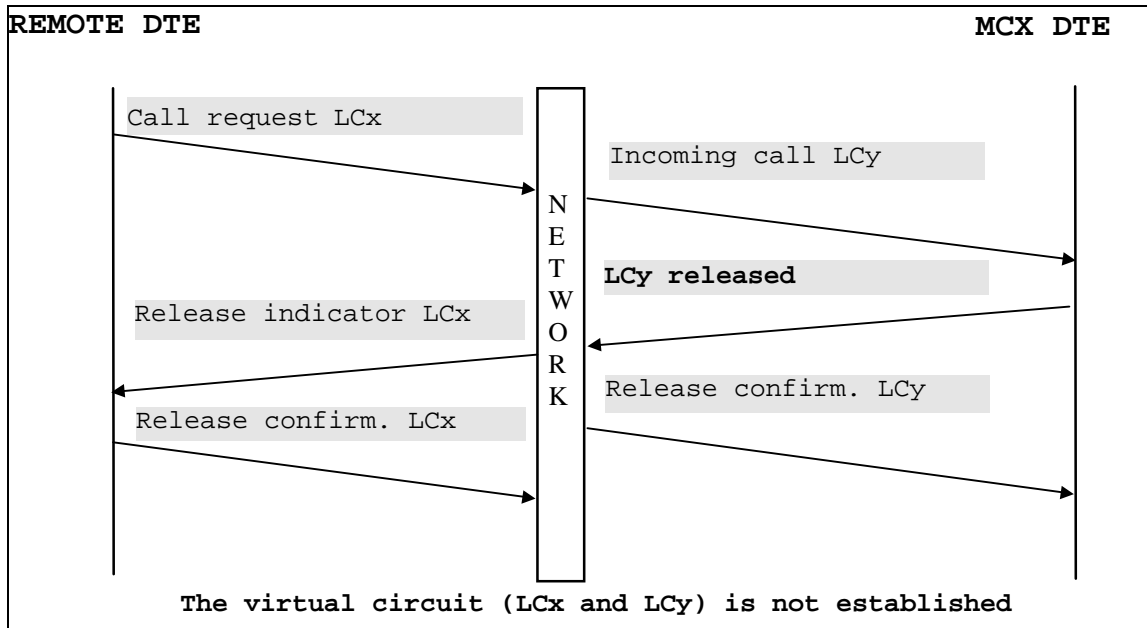


Figure VI-2

The **PRCTL+X25LISTN** command has the following parameters:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 3	X	X	X	X	X	X	X	X
PARAMETER 4	X	X	X	X	X	X	X	X
PARAMETER 5	X	X	X	X	X	X	X	X
PARAMETER 6	X	X	X	X	X	X	X	X
PARAMETER 7	0	0	0	0	0	0	1	0
PARAMETER 8	0	0	0	0	0	Autolib	Gw	Autocal
PARAMETER 9	LIdwl	LIdwl	LIdwl	LIdwl	LIdwl	LIdwl	LIdwl	LIdwl
PARAMETER 10	LIdwh	LIdwh	LIdwh	LIdwh	LIdwh	LIdwh	LIdwh	LIdwh
PARAMETER 11	HIdwl	HIdwl	HIdwl	HIdwl	HIdwl	HIdwl	HIdwl	HIdwl
PARAMETER 12	HIdwh	HIdwh	HIdwh	HIdwh	HIdwh	HIdwh	HIdwh	HIdwh
PARAMETER 13	<i>Tal</i>	<i>Tal</i>	<i>Tal</i>	<i>Tal</i>	<i>Tal</i>	<i>Tal</i>	<i>Tal</i>	<i>Tal</i>
PARAMETER 14	<i>Tah</i>	<i>Tah</i>	<i>Tah</i>	<i>Tah</i>	<i>Tah</i>	<i>Tah</i>	<i>Tah</i>	<i>Tah</i>
PARAMETER 15	0	0	0	0	0	0	0	0
PARAMETER 16	lloc	lloc	lloc	lloc	lloc	lloc	lloc	lloc
PARAMETER 17	ldis	ldis	ldis	ldis	ldis	ldis	ldis	ldis
DATA ZONE	WILDCARD							

X: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 X: PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 X: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 X: PARAMETER #6 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters are not significant.

OP: PARAMETER #7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

In this case, parameter #7 contains opcode X25LISTN (02h).

Autocal: PARAMETER #8 DB0

If this bit equals 1, it indicates that the incoming call packet is processed in the card. In this case, the MCX card is responsible for confirming the incoming call.

If this bit equals 0, this bit indicates that the incoming call packet is returned to the application with the user data and facilities. In this case, the application is responsible for confirming the incoming call.

Gw: PARAMETER #8 DB1

Gw = 1..... Create wildcard (the wildcard is in the data zone)

Gw = 0..... Delete wildcard. Only the Idw identifier is used, the other parameters and data are ignored

Autolib: PARAMETER #8 DB2

If this bit equals 1, it indicates that the indication of clear packet is processed in the card.

If this bit equals 0, it indicates that the indication of clear packet is returned by the application with the user data and facilities. In this case, the application is responsible for confirming the clear indication.

LIdwl: PARAMETER #9 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
Parameter #9 contains bits 0 to 7 of the of the identifier associated with the wildcard.

LIdwh: PARAMETER #10 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
Parameter #10 contains bits 8 to 15 of the identifier associated with the wildcard.

HIIdwl: PARAMETER #11 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
Parameter #11 contains bits 16 to 23 of the identifier associated with the wildcard.

HIIdwh: PARAMETER #12 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
Parameter #12 contains bits 24 to 31 of the identifier associated with the wildcard.

Tal: PARAMETER #13 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
Tah: PARAMETER #14 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
These 2 parameters define the least significant byte and most significant byte of a call packet's timeout. This timeout is not implemented in the current version of the program. These 2 parameters must be set to 0.

PARAMETER #15 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
Parameter #15 must be set to 0.

lloc: PARAMETER #16 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
This parameter contains the length of the wildcard associated with the called MCX DTE's principal and complementary address.

ldis: PARAMETER #17 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
This parameter contains the length of the wildcard associated with the calling DTE's principal and complementary address.

DATA ZONE

The data zone contains the ASCII-coded wildcard. It is not required in requests to delete the wildcard, because only its identifier is used.

The wildcard is made up of 2 ASCII character strings:

The first string contains the wildcard associated with the MCX DTE's principal and complementary addresses.

The second string contains the wildcard associated with the initiating DTE's principal and complementary addresses. These 2 strings must be separated by any character (/, for example). Characters recognized in the wildcard are numerical characters (0 to 9), the "?" character that represents any character, and the "*" character that represents any series of characters. Wildcards can only contain a single "*" character.

Examples of wildcards:

?1/*12: the call is confirmed only if the called address includes 2 digits and ends with 1, and if the calling address ends with the digits 12.

***/456:** The call is confirmed whatever the called address, and if the calling address equals 456.

ERROR CODE PRCTL+X25LISTN

If the command executed normally, error code 0 is returned. If not, code 171 (ABh) is returned to indicate an error in the parameters.

The cause of error 171 may be:

- Parameter #1: Invalid channel #
- Parameter #8 invalid
- Parameters #13, 14 and 15 not null
- Parameter #16: Length of the wildcard equal to 0 or greater than 15
- Parameter #17: Length of the wildcard equal to 0 or greater than 15
- Number of wildcards greater than a specified limit (currently 15)
- Wildcard identifier already used (when creating a wildcard)
- Invalid wildcard format
- Wildcard identifier missing (when deleting a wildcard)

VI.2.2.5. *PRCTL + opcode X25CALL (03h) command*

The X25CALL opcode sets up an X25 connection initiated by the MCX card on a logical channel with the highest possible number among the free bi-directional or outgoing logical channels. This minimizes the risk of call collision (but the risk remains nevertheless for bi-directional calls!).

Warning: the logical channel on which the call packet is sent is determined by the MCX card. Users learn of this logical channel when the end-of-command interrupt occurs by reading addresses 84 (54h) and 85 (55h) in the mailbox's interrupt zone.

Warning: the network must not delay the response to a call longer than necessary, because in that case the MCX card will certainly clear the call after a timeout of 200s. In this case, the MCX card sends an IT6 interrupt event to the PC with the *"timeout during outgoing call"* event code. In this case, the card immediately initiates a clear procedure.

After the X25CALL operation, the PC receives an IT6 interrupt event with one of the following codes:

- **Timeout during outgoing call:** The network did not reply. In this case, the card initiates a clear procedure.
- **Outgoing call confirmed:** The VC is established with the remote DTE.
(See the diagram for a normal VC set-up initiated by the MCX card)
- **Clear received:** The call is refused by the remote DTE or the network
(See the diagram for a VC set-up initiated by the MCX card but refused by the network or the DTE)

Set-up Diagram for a normal VC initiated by the MCX card

A normal connection is set up as follows:

The MCX DTE, in this call the initiating DTE, sends a “call request” packet, stipulating the address of the responding DTE on a free logical channel with the highest possible number among the bi-directional or unidirectional outgoing channels, and starts a timeout counter set to 200s.

The network then sends an “incoming call” packet to the DTE on the free logical channel with the lowest number.

The responding DTE indicates that it accepts the call by transmitting a “call accepted” packet to the network, or else it initiates the clear procedure described below.

The network indicates to the MCX DTE that the call is established by sending a call established packet on the same logical channel as the “call request” packet. The MCX card sends an IT6 interrupt event to the PC with the “*outgoing call confirmed*” event code.

The logical channels thus assigned make up a virtual circuit that will always be used for transmitting data between the MCX DTE and the remote DTE. The card is then ready to send and receive data packets on the VC thus established (see the BTRAN and RDBUF commands).

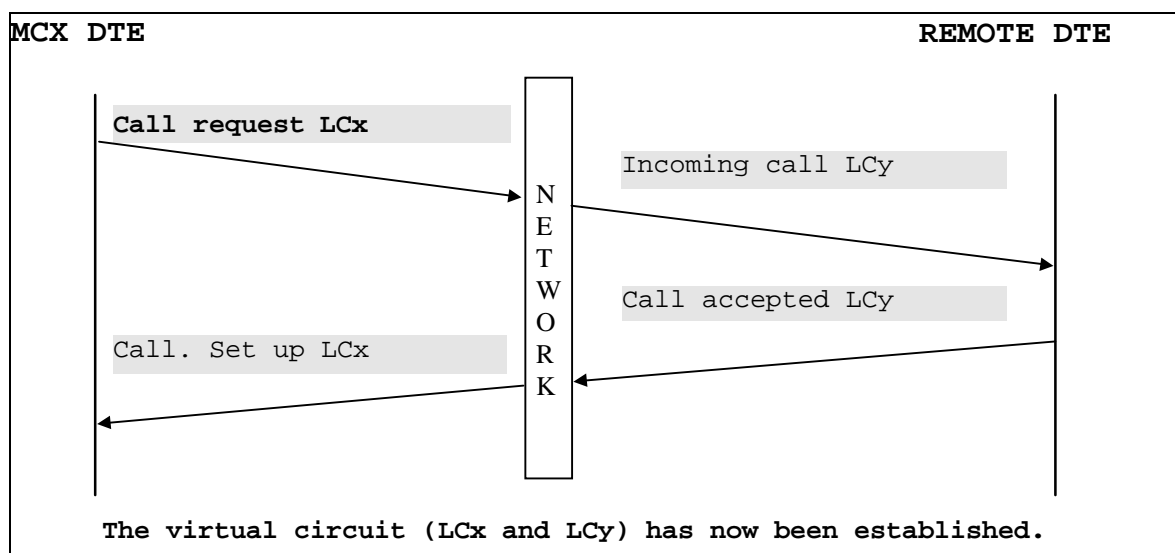


Figure VI-3

Set-up Diagram for a VC initiated by the MCX card and refused by the remote DTE

The failed communication occurs as follows:

The MCX DTE, in this case the initiating DTE, sends a “call request” packet, stipulating the address of the responding DTE on the free logical channel with the highest possible number among the bi-directional or unidirectional outgoing logical channels and starts a timeout counter set to 200s.

The network then sends an “incoming call” packet to the responding DTE on the free logical channel with the lowest number.

The responding DTE indicates that it refuses the communication by transmitting a “clear request” packet to the network.

The network informs the MCX DTE that the call is refused by sending it a “indication of clear” packet on the same logical channel as the “call request” packet. The MCX card sends an IT6 interrupt event to the PC with the “*clear received*” event code and confirms the clear request. More information that may be used by the PC is written in this zone (X25 code for the reason for the clear and diagnostic code).

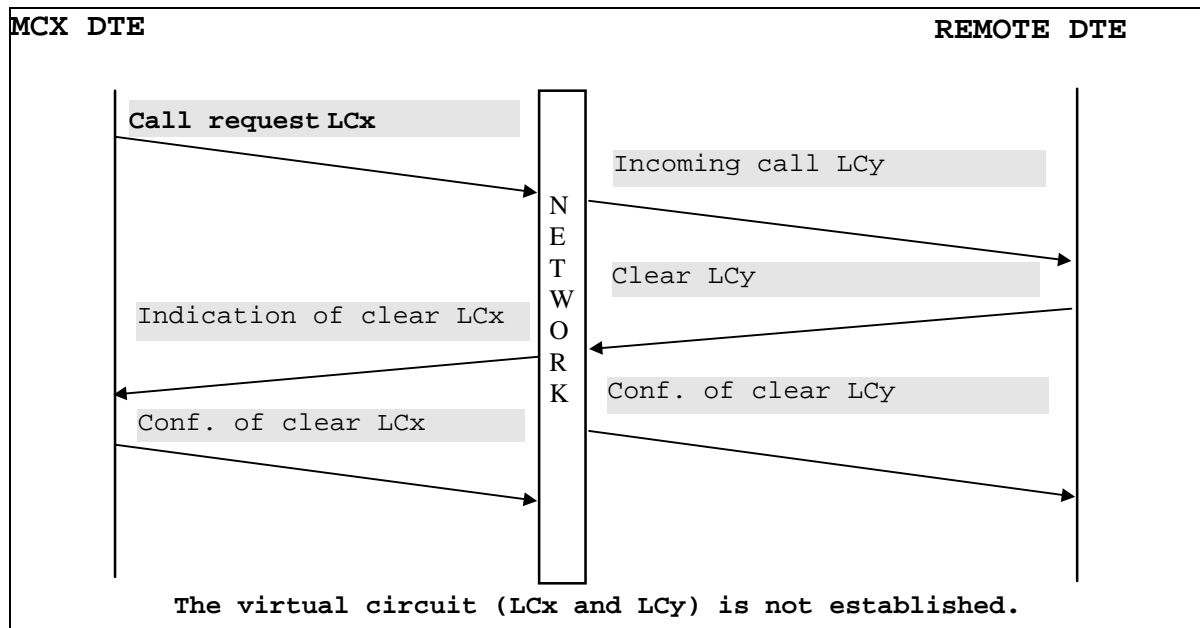


Figure VI-4

Set-up Diagram for a VC initiated by the MCX card and refused by the network

The failed communication occurs as follows:

The MCX DTE, in this case the initiating DTE, sends a “call request” packet, stipulating the address of the responding DTE on a free logical channel with the highest possible number among the bi-directional or unidirectional outgoing logical channels and starts a timeout counter set to 200s.

The network refuses the call request by sending a “indication of clear” packet to the MCX DTE. The MCX card sends an IT6 interrupt event to the PC with the “*clear received*” event code and confirms the clear request. More information that may be used by the PC is written in this zone (X25 code for the reason for the clear and diagnostic code).

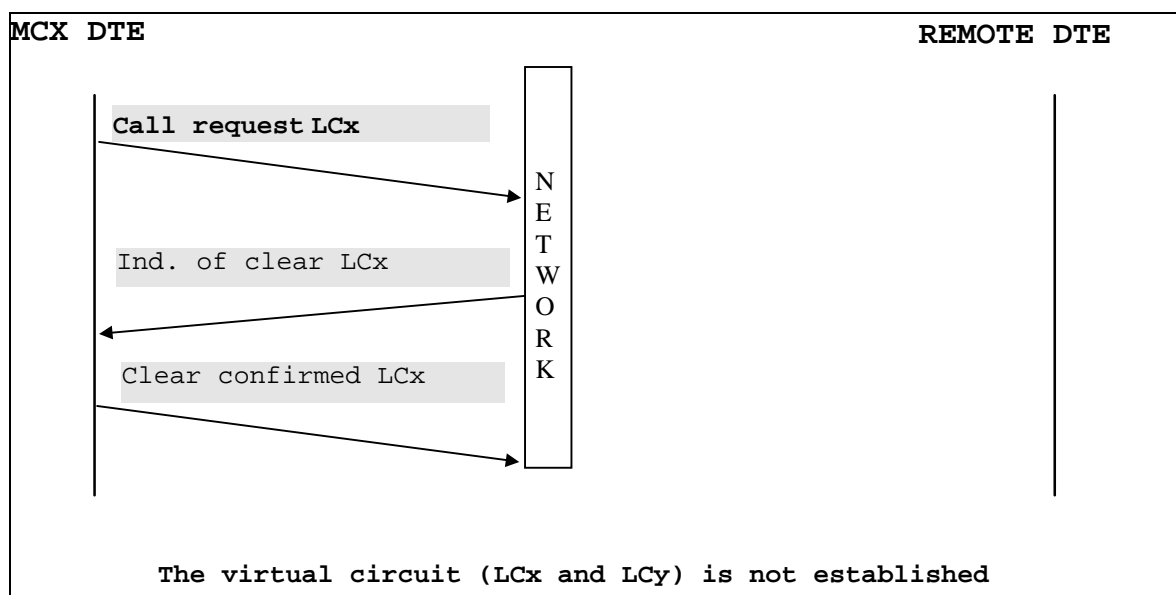


Figure VI-5

The **PRCTL+X25CALL** command has the following parameters:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 3	0	0	0	0	0	0	0	0
PARAMETER 4	0	<i>D</i>	0	0	0	0	0	0
PARAMETER 5	0	0	0	0	0	0	0	0
PARAMETER 6	0	0	0	0	0	0	0	0
PARAMETER 7	0	0	0	0	0	0	1	1
PARAMETER 8	GaVL	GaVL	0	0	0	Autolib	0	<i>Autoconf</i>
PARAMETER 9	0	0	0	0	0	0	0	0
PARAMETER 10	0	0	0	0	0	0	0	0
PARAMETER 11	0	0	0	0	0	0	0	0
PARAMETER 12	0	0	0	0	0	0	0	0
<i>PARAMETER 13</i>	<i>T21l</i>	<i>T21l</i>	<i>T21l</i>	<i>T21l</i>	<i>T21l</i>	<i>T21l</i>	<i>T21l</i>	<i>T21l</i>
<i>PARAMETER 14</i>	<i>T21h</i>	<i>T21h</i>	<i>T21h</i>	<i>T21h</i>	<i>T21h</i>	<i>T21h</i>	<i>T21h</i>	<i>T21h</i>
PARAMETER 15	0	0	0	0	0	0	0	0
PARAMETER 16	Res	Res	Res	Res	Res	Res	Res	Res
PARAMETER 17	lloc	lloc	lloc	lloc	ldis	ldis	ldis	ldis
PARAMETER 18	lsc	lsc	lsc	lsc	lsc	lsc	lsc	lsc
PARAMETER 19	ld	ld	ld	ld	ld	ld	ld	ld
DATA ZONE	Responder's X25 address coded in ASCII Facilities Call data							

PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #3 must be set to 0.

D: PARAMETER #4 DB6

State of D bit in the call packet's header.

D = 0: D bit equals 0

D = 1: D bit equals 1

This bit must be set to 0.

PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

PARAMETER #6 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameters #5 and #6 must be set to 0.

OP: PARAMETER #7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

In this case, parameter #7 contains opcode X25CALL (03h).

Autoconf: PARAMETER #8 DB0

If this bit equals 1, it indicates that the call confirmation packet is processed in the card.

If this bit equals 0, it indicates that the call confirmation packet is returned to the application with the user data and facilities.

This bit must be set to 1.

Autolib: PARAMETER #8**DB2**

If this bit equals 1, it indicates that the indication of clear packet is processed in the card.
 If this bit equals 0, it indicates that the indication of clear packet is returned by the application with the user data and facilities. In this case, the application is responsible for confirming the indication of clear.

GaVL: PARAMETER #8**DB7.DB6**

DB7	DB6	Range of logical channels
0	0	Reserved
0	1	Reserved
1	0	Bi-directional
1	1	Reserved

PARAMETER #9	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #10	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #11	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #12	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

All these parameters must be set to 0.

T21l: PARAMETER #13 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

T21h: PARAMETER #14 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the most significant byte and the least significant byte of the T21 call confirmation timeout. In the current version of the program, the timeout is set to 200s and cannot be changed. These 2 parameters must therefore be set to 0.

PARAMETER #15 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter must be set to 0.

Res: PARAMETER #16 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter must be set to 0.

ldis: PARAMETER #17 DB3.DB2.DB1.DB0

This parameter contains the length in digits of the responding (remote) DTE's X25 address.

lloc: PARAMETER #17 DB7.DB6.DB5.DB4

This parameter contains the length in digits of the local initiator's X25 sub-address. The principal local address is provided in the PROTO command. The total length of the principal address and sub-address must not exceed 15 bytes.

lsc: PARAMETER #18 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the length (in bytes) of the facilities field. The length must not exceed 109 bytes. If no facilities are required, this parameter must be null. The MCX card does not check the contents of the facilities field.

ld: PARAMETER #19 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the length (in bytes) of the user's call data. The length must not exceed 128 bytes if the "fast selection" additional service is selected and 16 bytes if it is not selected. In any case, the MCX card simply checks that the length does not exceed 128 bytes. If no call data is required, this parameter must be null.

DATA ZONE

The data zone contains the following information (in this order):

- The local initiator's X25 sub-address

- The X25 address of the responding DTE in ASCII format (15 digits max.).

- The optional facilities field

- The optional user call data

ERROR CODE PRCTL+X25CALL

If the command executed normally, error code 0 is returned. If not, code 171 (ABh) or 188 (BCh) is returned to indicate an error in the parameters or a protocol error.

The cause of error 171 may be:

- Parameter #1: Invalid channel #
- Parameter #8 invalid
- Parameters #9 to #16 not null
- Parameter #17: Length of the principal address+sub-address greater than 15
- Parameter #17: Length of the responder's X25 address null

The cause of error 188 may be:

- Length of the facilities greater than 109
- Length of the user's call data greater than 128
- Invalid data link status
- No more logical channels available
- Invalid X25 address format

VI.2.2.6. PRCTL + opcode X25LIB (04h) command

Opcode X25LIB liberates an established virtual circuit; this is initiated by the MCX card, which starts a timeout counter set to 180s.

Clear Diagram for a virtual circuit initiated by the MCX card

The call is cleared as follows:

The MCX DTE, in this case the initiating DTE, sends a “clear request” packet, stipulating the logical channel to be cleared.

The network then sends a “Clear confirmation” packet to the MCX DTE. The MCX card sends an IT6 interrupt event with the “*clear confirmed*” event code.

From that point onwards, the MCX DTE can no longer receive data from the remote DTE over this virtual circuit; it can set up a new virtual circuit on this logical channel. The network informs the remote DTE of the clear request by sending it a “clear indication” packet. The remote DTE must respond with a “Clear confirmation” packet.

If the call times out, one last clear attempt is made. After both attempts have timed out, the MCX card sends an IT6 interrupt event to the PC with the “*clear timeout*” event code. The logical channel is considered to be cleared and may be used again for new calls.

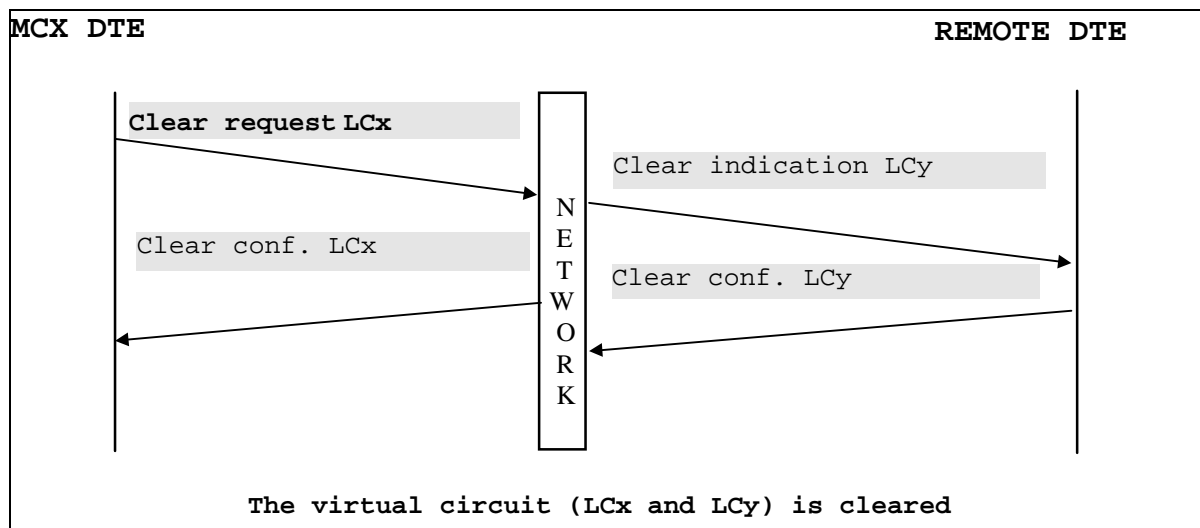


Figure VI-6

The **PRCTL+X25LIB** command has the following parameters:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 3	0	0	0	0	0	0	0	0
PARAMETER 4	0	0	0	0	0	0	0	0
PARAMETER 5	LC	LC	LC	LC	LC	LC	LC	LC
PARAMETER 6	0	0	0	0	LCG	LCG	LCG	LCG
PARAMETER 7	0	0	0	0	0	1	0	0
PARAMETER 8	0	0	0	0	0	<i>Evlib</i>	0	<i>Autoconf</i>
PARAMETER 9	Clib	Clib	Clib	Clib	Clib	Clib	Clib	Clib
PARAMETER 10	Dlib	Dlib	Dlib	Dlib	Dlib	Dlib	Dlib	Dlib
PARAMETER 11	0	0	0	0	0	0	0	0
PARAMETER 12	0	0	0	0	0	0	0	0
<i>PARAMETER 13</i>	<i>T23l</i>	<i>T23l</i>	<i>T23l</i>	<i>T23l</i>	<i>T23l</i>	<i>T23l</i>	<i>T23l</i>	<i>T23l</i>
<i>PARAMETER 14</i>	<i>T23h</i>	<i>T23h</i>	<i>T23h</i>	<i>T23h</i>	<i>T23h</i>	<i>T23h</i>	<i>T23h</i>	<i>T23h</i>
PARAMETER 15	lsc	lsc	lsc	lsc	lsc	lsc	lsc	lsc
PARAMETER 16	ld	ld	ld	ld	ld	ld	ld	ld
DATA ZONE	Facilities User clear data							

PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameters #3 and #4 must be set to 0.

LC: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the logical channel number.

LCG: PARAMETER #6 DB3.DB2.DB1.DB0

This parameter contains the logical channel group number.

OP: PARAMETER #7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

In this case, parameter #7 contains opcode X25LIB (04h).

Autoconf: PARAMETER #8 DB0

If this bit equals 1, it indicates that the clear confirmation packet is processed in the card.

If this bit equals 0, it bit indicates that the clear confirmation packet is returned by the application with facilities.

This bit must be set to 1.

Evlib: PARAMETER #8 DB2

Evlib = 0 .. The end-of-clear event does not trigger an IT6 interrupt event

Evlib = 1 .. The end-of-clear event triggers an IT6 interrupt event if the interrupt is authorized by the MINTR command

This bit must be set to 1.

Clib: PARAMETER #9 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the reason for the clear. Warning: this parameter is not checked by the card.

Dlib: PARAMETER #10 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the clear diagnostic. Warning: this parameter is not checked by the card.

PARAMETER #11 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

PARAMETER #12 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

All these parameters must be set to 0.

T23l: PARAMETER #13 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

T23h: PARAMETER #14 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the most significant byte and the least significant byte of the T23 timeout for the clear confirmation. In the current version of the program, the timeout is set to 180s and cannot be changed. These 2 parameters must therefore be set to 0.

lsc: PARAMETER #15 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the length (in bytes) of the facilities field. The length must not exceed 109 bytes. If no facilities are required, this parameter must be null. The MCX card does not check the contents of the facilities field.

ld: PARAMETER #16 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the length (in bytes) of the user clear data. The length must not exceed 128 bytes if the “fast selection” additional service is selected, and 16 bytes if not. However, the MCX card simply checks that the length does not exceed 128 bytes. If no clear data is required, this parameter must be null.

DATA ZONE

The data zone contains the following information (in this order):

The optional facilities field

The optional user release data

ERROR CODE PRCTL+X25LIB

If the command executed normally, error code 0 is returned. If not, code 171 (ABh) or 188 (BCh) is returned to indicate an error in the parameters or a protocol error.

The cause of error 171 may be:

- Parameter #1: Invalid channel #
- Parameters #5 to 6: logical channel # null
- Parameters #5 to 6: logical channel # not assigned
- Parameter #8 invalid
- Parameters #11, 12, 13, or 14 not null

The cause of error 188 may be:

- Length of facilities greater than 109 bytes
- Length of user clear data greater than 128 bytes
- Invalid data link status
- Invalid logical channel status

VI.2.2.7. *PRCTL + opcode X25RESET (05h) command*

Opcode X25RESET sends a reset packet to an established virtual circuit and starts a timeout counter set to 180s.

This resets a virtual circuit, i.e. it deletes all the data packets received that have not been read and the data packets that have not been sent (ditto for the interrupt packets).

Example of a reset between the MCX DTE and a remote DTE

The call is reset as follows:

The MCX DTE sends a “reset request” packet, stipulating the logical channel to be reset.

The network then sends a “reset confirmation” packet to the MCX DTE. The MCX card can generate an IT6 interrupt event on the PC’s bus with the “reset confirmed” event code, specified by the value 11 at address 83 (53h) in the mailbox’s interrupt zone.

If the reset times out, the MCX card triggers an IT6 interrupt event on the PC’s bus with the “timeout reset” event code specified by the value 12 at address 83 (53h) in the mailbox’s interrupt zone. A final reset attempt is made using the same procedure. After both attempts have timed out, the logical channel is considered to be reset.

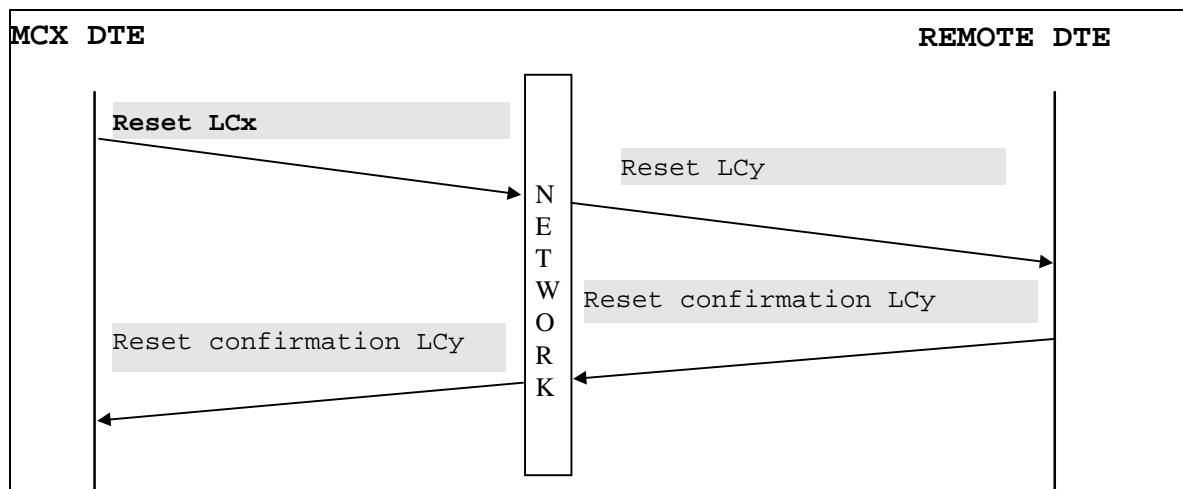


Figure VI-7

The **PRCTL+X25RESET** command has the following parameters:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 3	0	0	0	0	0	0	0	0
PARAMETER 4	0	0	0	0	0	0	0	0
PARAMETER 5	LC	LC	LC	LC	LC	LC	LC	LC
PARAMETER 6	0	0	0	0	LCG	LCG	LCG	LCG
PARAMETER 7	0	0	0	0	0	1	0	1
PARAMETER 8	0	0	0	0	0	<i>Evrest</i>	0	0
PARAMETER 9	Cse	Cse	Cse	Cse	Cse	Cse	Cse	Cse
PARAMETER 10	Diag	Diag	Diag	Diag	Diag	Diag	Diag	Diag
PARAMETER 11	0	0	0	0	0	0	0	0
PARAMETER 12	0	0	0	0	0	0	0	0
PARAMETER 13	<i>T22l</i>	<i>T22l</i>	<i>T22l</i>	<i>T22l</i>	<i>T22l</i>	<i>T22l</i>	<i>T22l</i>	<i>T22l</i>
PARAMETER 14	<i>T22h</i>	<i>T22h</i>	<i>T22h</i>	<i>T22h</i>	<i>T22h</i>	<i>T22h</i>	<i>T22h</i>	<i>T22h</i>

PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #3 must be set to 0.

PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #4 must be set to 0.

LC: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the logical channel number.

LCG: PARAMETER #6 DB3.DB2.DB1.DB0

This parameter contains logical channel group number.

OP: PARAMETER #7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

In this case, parameter #7 contains opcode X25RESET (05h).

Evrest: PARAMETER #8 DB2

Evrest = 0 The "end of reset" event does not trigger an IT6 interrupt event

Evrest = 1. The "end of reset" event triggers an IT6 interrupt event if
the interrupt is authorized by the MINTR command

This bit must be set to 1.

Cse: PARAMETER #9 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the reason for the reset. Warning: this parameter is not checked by the card.

Diag: PARAMETER #10 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the reset diagnostic. Warning: this parameter is not checked by the card.

PARAMETER #11	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
PARAMETER #12	DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Both these parameters must be set to 0.

<i>T22l: PARAMETER #13</i>	<i>DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0</i>
----------------------------	--

<i>T22h: PARAMETER #14</i>	<i>DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0</i>
----------------------------	--

These parameters define the most significant byte and the least significant byte of the T22 reset confirmation timeout. In the current version of the program, the timeout is set to 180s and cannot be changed. These 2 parameters must therefore be set to 0.

ERROR CODE PRCTL+X25RESET

If the command executed normally, error code 0 is returned. If not, the 171 (ABh) or 188 (BCh) is returned to indicate an error in the parameters or a protocol error.

The cause of error 171 may be:

- Parameter #1: Invalid channel #
- Parameters #5 to 6: Logical channel # null
- Parameters #5 to 6: Logical channel # not assigned
- Parameter #8 invalid
- Parameters #11,12,13 or 14 not null

The cause of error 188 may be:

- Invalid data link status
- Invalid logical channel status

VI.2.2.8. PRCTL + opcode X25FCALL (06h) command

Opcode X25FCALL manually confirms an incoming call.

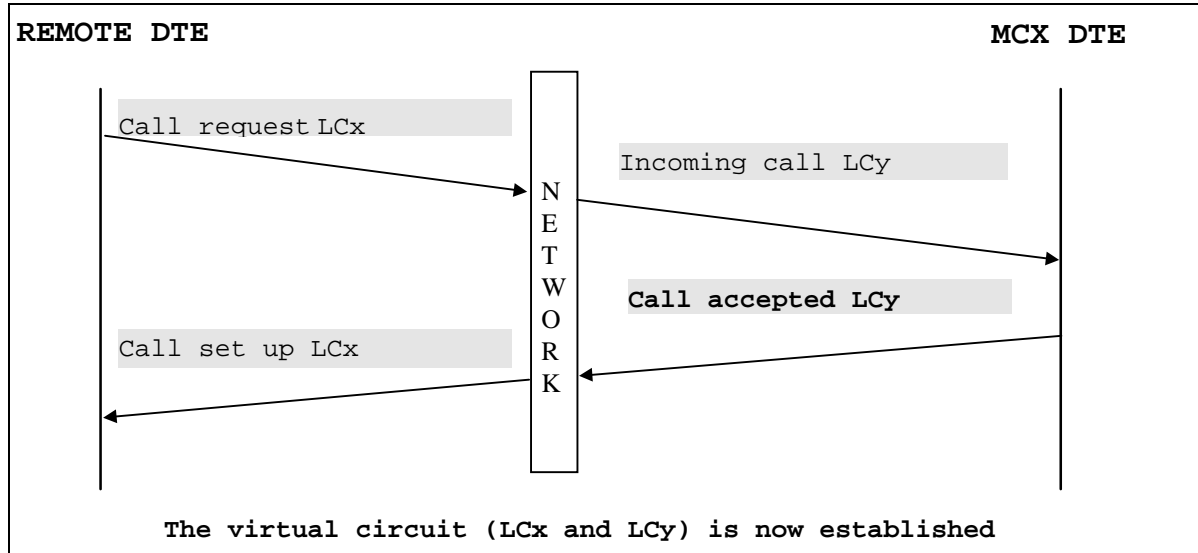


Figure VI-8

The PRCTL +**X25CFCALL** command has the following parameters:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 3	0	0	0	0	0	0	0	0
PARAMETER 4	0	<i>D</i>	0	0	0	0	0	0
PARAMETER 5	LC	LC	LC	LC	LC	LC	LC	LC
PARAMETER 6	0	0	0	0	LCG	LCG	LCG	LCG
PARAMETER 7	0	0	0	0	0	1	1	0
PARAMETER 8	lsc	lsc	lsc	lsc	lsc	lsc	lsc	lsc
PARAMETER 9	ld	ld	ld	ld	ld	ld	ld	ld
DATA ZONE	Facilities Call confirmation data							

PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
Parameter #3 must be set to 0.

D: PARAMETER #4 DB6
The state of bit D in the call confirmation packet header.
D = 0: D bit equals 0
D = 1: D bit equals 1
This bit must be set to 0.

LC: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
This parameter contains the logical channel number.

LCG: PARAMETER #6 DB3.DB2.DB1.DB0
This parameter contains the logical channel group number.

OP: PARAMETER #7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
In this case, parameter #7 contains opcode X25CFCALL (06h).

lsc: PARAMETER #18 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
This parameter contains the length (in bytes) of the facilities field. The length must not exceed 109 bytes. If no facilities are required, this parameter must be null. The MCX card does not check the contents of the facilities field.

ld: PARAMETER #19 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
This parameter contains the length (in bytes) of the user's call data. The length must not exceed 128 bytes if the "fast selection" additional service is selected and 16 bytes if not. However, the MCX card simply checks that the length does not exceed 128 bytes. If no call data is required, this parameter must be null.

DATA ZONE

The data zone contains the following information (in this order):
The optional facilities field
The optional user call confirmation data

ERROR CODE PRCTL+X25CFCALL

If the command executed normally, error code 0 is returned. If not, code 171 (ABh) or 188 (BCh) is returned to indicate an error in the parameters or a protocol error.

The cause of error 171 may be:

- Parameter #1: Invalid channel #
- Parameter #3 not null
- Parameter #4 invalid

The cause of error 188 may be:

- Length of facilities greater than 109
- Length of the user's call data greater than 128
- Invalid data link status
- Invalid logical channel status

VI.2.2.9. PRCTL + opcode X25CFLIB (07h) command

Opcode X25CFLIB manually confirms a clear indication.

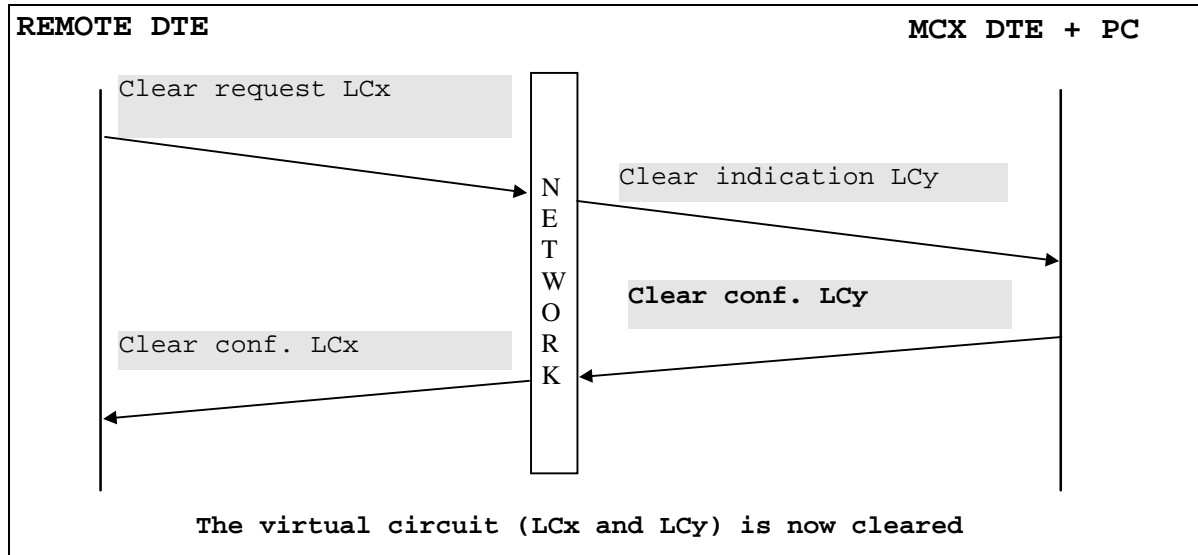


Figure VI-9

The PRCTL +**X25CFLIB** command has the following parameters:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 3	0	0	0	0	0	0	0	0
PARAMETER 4	0	0	0	0	0	0	0	0
PARAMETER 5	LC	LC	LC	LC	LC	LC	LC	LC
PARAMETER 6	0	0	0	0	LCG	LCG	LCG	LCG
PARAMETER 7	0	0	0	0	0	1	1	1

PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0
 PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameters #3 and 4 must be set to 0.

LC: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the logical channel number.

LCG: PARAMETER #6 DB3.DB2.DB1.DB0

This parameter contains the logical channel group number.

OP: PARAMETER #7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

In this case, parameter #7 contains opcode X25CFLIB (07h).

ERROR CODE PRCTL+X25CFLIB

If the command executed normally, error code 0 is returned. If not, code 171 (ABh) or 188 (BCh) is returned to indicate an error in the parameters or a protocol error.

The cause of error 171 may be:

- Parameter #1: Invalid channel #
- Parameter #3 not null
- Parameter #4 not null

The cause of error 188 may be:

- Invalid data link status
- Invalid logical channel status

VI.2.2.10. PRCTL + opcode X25CFINT (08h) command

Opcode X25CFINT manually confirms an interrupt packet.

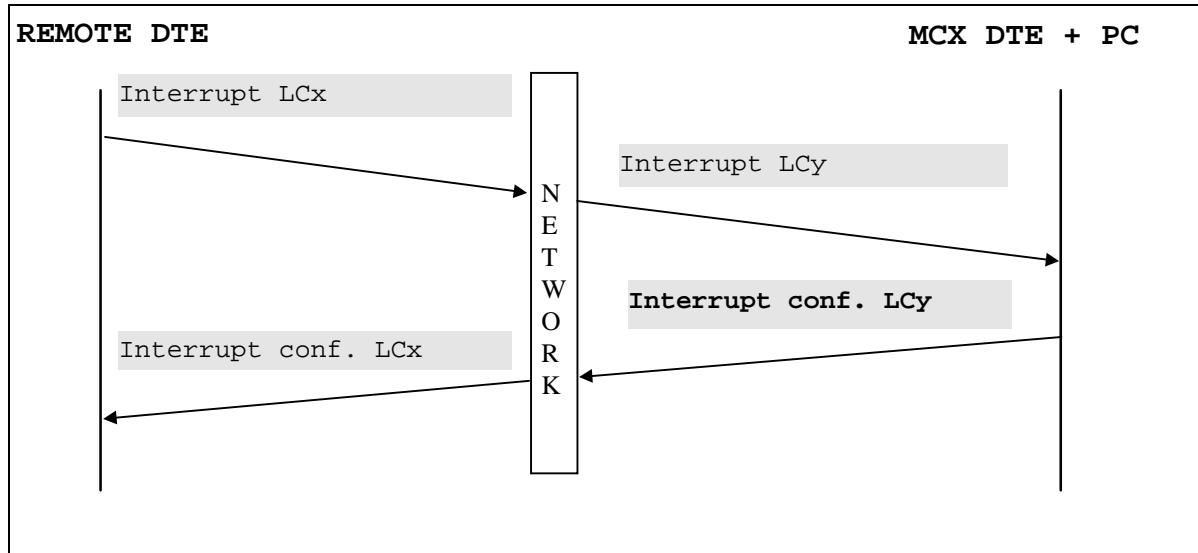


Figure VI-10

The PRCTL +**X25CFINT** command has the following parameters:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 3	0	0	0	0	0	0	0	0
PARAMETER 4	0	0	0	0	0	0	0	0
PARAMETER 5	LC	LC	LC	LC	LC	LC	LC	LC
PARAMETER 6	0	0	0	0	LCG	LCG	LCG	LCG
PARAMETER 7	0	0	0	0	1	0	0	0

PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameters #3 and 4 must be set to 0.

LC: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the logical channel number.

LCG: PARAMETER #6 DB3.DB2.DB1.DB0

This parameter contains the logical channel group number.

OP: PARAMETER #7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

In this case, parameter #7 contains opcode X25CFINT (08h).

ERROR CODE PRCTL+X25CFINT

If the command executed normally, error code 0 is returned. If not, code 171 (ABh) or 188 (BCh) is returned to indicate an error in the parameters or a protocol error.

The cause of error 171 may be:

- Parameter #1: Invalid channel #
- Parameter #3 not null
- Parameter #4 not null

The cause of error 188 may be:

- Invalid data link status
- Invalid logical channel status

VI.2.2.11. PRCTL + opcode X25GETPKT (09h) command

Opcode X25GETPKT reads the contents of the following packets:

- Call packet and call confirmation packet
- Clear packet and clear confirmation packet

Important note concerning the call packet:

This packet can be read by the PC if all the following conditions are met:

- A “call packet received” interrupt event was sent to the PC
- The call packet was not filtered by the card (see X25LISTN)
- The call packet was not processed automatically by the card (see X25LISTN)
- A call confirmation packet (see X25CFCALL) was not sent by the PC

Important note concerning the clear packet:

This packet can be read by the PC if all the following conditions are met:

- A “clear requested by the DCE” interrupt event was sent to the PC
- The clear packet was not processed automatically by the card (see X25LISTN or X25CALL)
- A clear confirmation packet (see X25CFLIB) was not sent by the PC

Important note concerning the call confirmation packet:

This packet can be read by the PC if all the following conditions are met:

- A “call confirmed” interrupt event was sent to the PC
- The call confirmation packet was not processed in the card (see X25CALL).
- A clear procedure with manual response was not initiated by the DCE.

Important note concerning the clear confirmation packet:

This packet can be read by the PC if all the following conditions are met:

- A “clear confirmed” interrupt event was sent to the PC
- The clear confirmation packet was not processed in the card (see X25LIB)
- A call request with manual response was not received.

The PRCTL +**X25GETPKT** command has the following parameters:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PARAMETER 3	Lnl	Lnl	Lnl	Lnl	Lnl	Lnl	Lnl	Lnl
PARAMETER 4	Lnh	Lnh	Lnh	Lnh	Lnh	Lnh	Lnh	Lnh
PARAMETER 5	LC	LC	LC	LC	LC	LC	LC	LC
PARAMETER 6	0	0	0	0	LCG	LCG	LCG	LCG
PARAMETER 7	0	0	0	0	1	0	0	1
DATA	X25 Packet							

Lnl: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Lnh: PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameters #3 and #4 are output parameters. After the command executes, they contain the least significant bytes (Lnl) and most significant bytes (Lnh) of the length of the X25 packet read. A length equal to 0 indicates that there are no packets to read.

LC: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the logical channel number.

LCG: PARAMETER #6 DB3.DB2.DB1.DB0

This parameter contains the logical channel group number.

OP: PARAMETER #7 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

In this case, parameter #7 contains opcode X25GETPKT (09h).

DATA ZONE

The data zone in the dual-access memory contains the X25 packet received. The data zone is located at a positive offset of 100 (64h) from the base address of the dual-ported memory.

ERROR CODE PRCTL+X25GETPKT

If the command executed normally, error code 0 is returned. If not, code 171 (ABh) is returned to indicate an error in the parameters.

The cause of error 171 may be:

- Parameter #1: Invalid channel #
- No packets to read

VI.2.3.BTRAN (08h): Send a data packet or an interrupt packet

OPCODE = 8 (08h)

DESCRIPTION

This command loads a data packet or an interrupt with the user data placed in the mailbox, and starts the transmission of this packet on the logical channel and the data link (channel) placed in the parameters.

The size of the user data block must be specified in parameters #2 and #3.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	1	0	0	0
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 3	Ln1h	Ln1h	Ln1h	Ln1h	Ln1h	Ln1h	Ln1h	Ln1h
PARAMETER 4	0	0	0	0	0	0	0	0
PARAMETER 5	LC	LC	LC	LC	LC	LC	LC	LC
PARAMETER 6	0	0	0	0	LCG	LCG	LCG	LCG
PARAMETER 7	Q	D	M	0	0	0	0	Int
DATA ZONE	USER DATA TO TRANSMIT							

C#: PARAMETER #1 DB6.DB5.DB4.DB3.DB2.DB1.DB0

These bits in parameter #1 indicate the number of the X25 data link. This number must be between 1 and the number of lines installed.

Ln1: PARAMETER #2 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Ln1h: PARAMETER #3 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

These parameters define the most significant byte and the least significant byte of the size of the user data block to transmit.

The maximum length of the user data is limited to 32 for an interrupt packet; for a data packet, it depends on the size allocated in the PROTO command.

PARAMETER #4 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

Parameter #4 must be set to 0.

LC: PARAMETER #5 DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0

This parameter contains the logical channel number.

LCG: PARAMETER #6 DB3.DB2.DB1.DB0

This parameter contains the logical channel group number.

Q: PARAMETER #7 **DB7**

Status of the data qualifier bit in the data packet header

Q = 1: Q bit equals 1

Q = 0: Q bit equals 0

D: PARAMETER #7 **DB6**

Status of D bit in the data packet header.

D = 0: D bit equals 0

D = 1: D bit equals 1

This bit must be set to 0.

M: PARAMETER #7 **DB5**

Status of M bit (delimit messages) in the data packet header.

M = 1: M bit equals 1

M = 0: M bit equals 0

Int: PARAMETER #7 **DB0**

The Int bit in parameter #7 is used to send data in:

a data packet if Int = 0

an interrupt packet Int = 1

Sending data to a remote DTE in an interrupt packet (from 1 to 32 bytes) allows you to send data without following the transfer and flow control procedures concerning data packets. Warning: the MCX DTE cannot send a new interrupt packet until the previous interrupt packet has been confirmed.

Example of an interrupt between the MCX DTE and a remote DTE

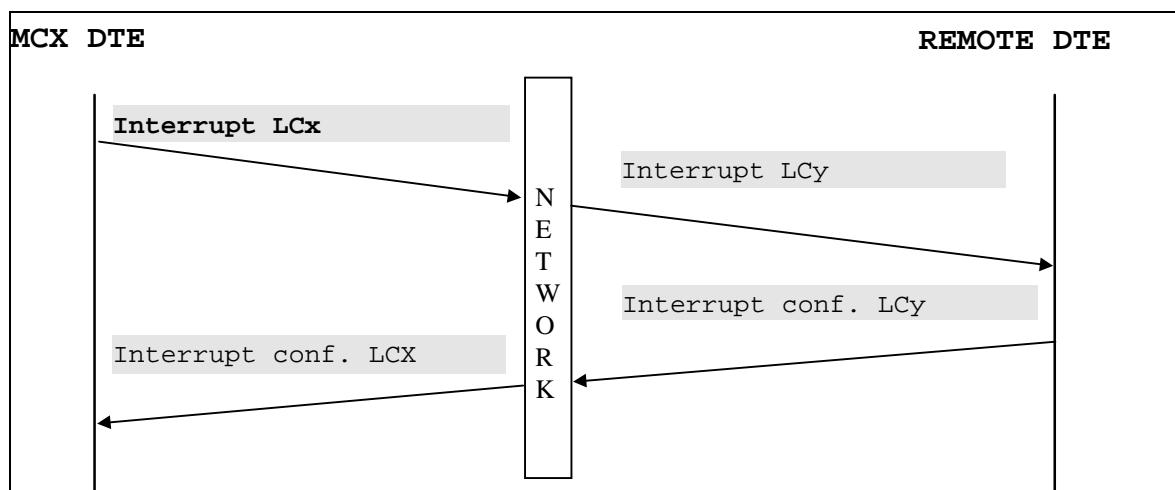


Figure VI-11

DATA ZONE

The data zone in the dual-ported memory contains the data to copy in the data packet's user data field. The data zone is located at a positive offset of 100 (64h) from the base address of the dual-ported memory.

ERROR CODE

If the command executed normally, error code 0 is returned. If not, code 137 (89h) is returned to indicate an error in at least one of the command parameters. If a protocol error is detected, the command returns error code 188 (BCh).

The cause of error 137 may be:

- Parameter #1: Invalid channel #
- Parameter #2 and 3: Size of the data block equal to 0
- Parameter #4 invalid
- Parameters #5 and 6: Logical channel # not assigned
- Parameters #5 and 6: Logical channel # null
- No more data buffers available (The "W=2" packets sent previously have not been acknowledged by the DCE)

The cause of error 188 may be:

- Length of user data greater than the size allocated in the PROTO command for a data packet
- Length of user data greater than 32 for an interrupt packet
- Invalid data link status (follow-on procedure in progress)
- Invalid logical channel status (no virtual calls are set up using this logical channel)

The diagnostic code written to address 87 (57h) in the INTERRUPT ZONE provides the exact cause of error 188. (See Appendix E in the X25 recommendation)

VI.2.4.RDBUF (09h): Read data in a data packet or an interrupt packet**OPCODE = 9 (09h)****DESCRIPTION**

This command reads the contents of the user data in a data packet or an interrupt packet received on a logical channel of an X25 data link. The card can store “W=2” data packets and one interrupt packet.

This command has the following format:

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPCODE	0	0	0	0	1	0	0	1
PARAMETER 1	0	C#	C#	C#	C#	C#	C#	C#
PARAMETER 2	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 3	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1	Ln1
PARAMETER 4	0	0	0	0	0	Typ	Typ	Typ
PARAMETER 5	LC	LC	LC	LC	LC	LC	LC	LC
PARAMETER 6	0	0	0	0	LCG	LCG	LCG	LCG
PARAMETER 7	Q	D	M	0	0	0	0	Int
DATA ZONE	DATA RECEIVED FROM THE REMOTE DTE							

C#: PARAMETER #1**DB6.DB5.DB4.DB3.DB2.DB1.DB0**

These bits in parameter #1 indicate the number of the X25 data link. This number must be between 1 and the number of lines installed.

Ln1: PARAMETER #2**DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0****Ln1: PARAMETER #3****DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

These 2 parameters are output parameters. After the RDBUF command executes, they contain the least significant bytes and the most significant bytes of the number of characters in the “user data field” of the data packet or interrupt packet received.

Typ: PARAMETER #4 **DB2.DB1.DB0**

These bits in parameter #4 indicate the type of read to perform, as shown below:

DB2	DB1	DB0	Type of read
0	0	0	Read data with acknowledgement
1	0	0	Read data with acknowledgement
0	0	1	Read data without acknowledgement

LC: PARAMETER #5 **DB7.DB6.DB5.DB4.DB3.DB2.DB1.DB0**

This parameter must contain the logical channel number.

LCG: PARAMETER #6 **DB3.DB2.DB1.DB0**

This parameter must contain the logical channel group number.

Q: PARAMETER #7 **DB7**

The Q bit is updated after the RDBUF command executes.

Status of the data qualifier bit in the data packet header received.

Q = 1: Q bit equals 1

Q = 0: Q bit equals 0

D: PARAMETER #7 **DB6**

The D bit is updated after the RDBUF command executes.

Status of bit D in the data packet header received.

D = 0: D bit equals 0

D = 1: D bit equals 1

In the current version of the program, this bit is equals 0.

M: PARAMETER #7 **DB5**

The M bit is updated after the RDBUF command executes.

Status of M bit (delimit messages) in the data packet header received.

M = 1: M bit equals 1

M = 0: M bit equals 0

Int: PARAMETER #7 **DB0**

The Int bit in parameter #7 has the following meaning:

Int = 0: Request to read data in a data packet

Int = 1: Request to read data in an interrupt packet

DATA ZONE

The data zone in the dual-ported memory contains the data packet's user data field received from the remote DTE. The data zone is located at a positive offset of 100 (64h) from the dual-ported memory's base address.

ERROR CODE

If the command executed normally, error code 0 is returned. If not, code 138 (8Ah) is returned to indicate a error in the command.

The cause of error may be:

- Parameter #1: Invalid channel #
- Parameter #4 invalid
- Parameters #5 and 6: Logical channel null
- Parameter #5 and 6: Logical channel not assigned
- Parameter #7 invalid
- No data to read

THIS PAGE INTENTIONALLY LEFT BLANK

VII. Appendices

VII.1. Command summary sorted by opcode

OPCODE	COMMAND	FUNCTION
00 / 00h	VINIT	Initialize communication parameters
01 / 01h	ALLOC	Initialize the size of asynchronous buffers
02 / 02h	ILOAD	Command not implemented
03 / 03h	VMODE	Program flow control modes
04 / 04h	RXENB	Enable or disable a receive channel
05 / 05h	MBOOT	Load a program into memory
06 / 06h	BDELE	Stop data transmission in progress
07 / 07h	ISEND	Command not implemented
08 / 08h	BTRAN	Transmit data
09 / 09h	RDBUF	Read data received
10 / 0Ah	CHDEF	Define string parameters
11 / 0Bh	STCNT	Define the size of blocks received
12 / 0Ch	MINTR	"Event interrupt" conditions
13 / 0Dh	RSTAT	Read the status of communication channels
14 / 0Eh	LTEST	Command not implemented
15 / 0Fh	BTEST	Command not implemented
16 / 10h	STIME	Command not implemented
17 / 11h	RMEMO	Read a memory block
18 / 12h	GOADR	Execute a program in memory
19 / 13h	RELRP	Read codes, revisions and identification
20 / 14h	ENDIT	Reserved
21 / 15h	DALOC	Deallocate asynchronous buffers
22 / 16h	HNGUP	Hang up modem
23 / 17h	XXXXX	Reserved
24 / 18h	BPARM	Read asynchronous buffer parameters
25 / 19h	RINIT	Reinitialize the card
26 / 1Ah	TFREE	Read space available in a TX buffer
27 / 1Bh	RXCNT	Read the number of characters or frames received
28 / 1Ch	TMRRP	Command not implemented
29 / 1Dh	CLRRX	Erase data received
30 / 1Eh	IMDWR	Command not implemented
31 / 1Fh	STTMO	Define receive "time out" values
32 / 20h	XXXXX	Reserved
33 / 21h	XXXXX	Reserved
34 / 22h	XXXXX	Reserved
35 / 23h	BREAK	Send a break on an asynchronous channel
36 / 24h	STSIG	Manual control of DTR and RTS signals
37 / 25h	XXXXX	Reserved (Debug monitor)
38 / 26h	XXXXX	Reserved
39 / 27h	XXXXX	Reserved
40 / 28h	ADDCM	Command not implemented
41 / 29h	ADDTX	Command not implemented
42 / 2Ah	ADDRX	Command not implemented

OPCODE (continued)	COMMAND (continued)	FUNCTION (continued)
43 / 2Bh	NOPER	NOP command (no operation)
44 / 2Ch	MSIZE	Command not implemented
45 / 2Dh	RSMDE	Initialize RS232D or RS422A mode
46 / 2Eh	FLASH	Command not implemented
47 / 2Fh	PROTO	Initialize a channel's protocol
48 / 30h	PRCTL	Communication procedures
50 / 32h	XXXXX	Reserved

VII.2. Error code summary

The following table indicates the type of error¹⁸ associated with each error code.

DEC =	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Return code
000	0	0	0	0	0	0	0	0	No error
129	1	0	0	0	0	0	0	1	VINIT error
130	1	0	0	0	0	0	1	0	ALLOC error
131	1	0	0	0	0	0	1	1	ILOAD error
132	1	0	0	0	0	1	0	0	VMODE error
133	1	0	0	0	0	1	0	1	RXENB error
134	1	0	0	0	0	1	1	0	MBOOT error
135	1	0	0	0	0	1	1	1	BDELE error
136	1	0	0	0	1	0	0	0	ISEND error
137	1	0	0	0	1	0	0	1	BTRAN error
138	1	0	0	0	1	0	1	0	RDBUF error
139	1	0	0	0	1	0	1	1	RDBUF error
140	1	0	0	0	1	1	0	0	CHDEF error
141	1	0	0	0	1	1	0	1	STCNT error
142	1	0	0	0	1	1	1	0	MINTR error
143	1	0	0	0	1	1	1	1	LTEST error
144	1	0	0	1	0	0	0	0	BTEST error
145	1	0	0	1	0	0	0	1	STIME error
146	1	0	0	1	0	0	1	0	RMEMO error
147	1	0	0	1	0	0	1	1	Reserved
148	1	0	0	1	0	1	0	0	BPARM error
149	1	0	0	1	0	1	0	1	TFREE error
150	1	0	0	1	0	1	1	0	RXCNT error
151	1	0	0	1	0	1	1	1	CLRRX error
152	1	0	0	1	1	0	0	0	IMDWR error
153	1	0	0	1	1	0	0	1	STTMO error
154	1	0	0	1	1	0	1	0	RINIT error
155	1	0	0	1	1	0	1	1	Reserved
156	1	0	0	1	1	1	0	0	Reserved
157	1	0	0	1	1	1	0	1	Reserved
158	1	0	0	1	1	1	1	0	BREAK error
159	1	0	0	1	1	1	1	1	HNGUP error
160	1	0	1	0	0	0	0	0	STSIG error
161	1	0	1	0	0	0	0	1	Reserved
162	1	0	1	0	0	0	1	0	RELRP error
163	1	0	1	0	0	0	1	1	Reserved
164	1	0	1	0	0	1	0	0	ADDCM error
165	1	0	1	0	0	1	0	1	ADDTX error
166	1	0	1	0	0	1	1	0	ADDRX error
167	1	0	1	0	0	1	1	1	DEBUG error
168	1	0	1	0	1	0	0	0	RSMDE error
169	1	0	1	0	1	0	0	1	FLASH error

¹⁸See the interrupt status zone.

170	1	0	1	0	1	0	1	0	PROTO error
171	1	0	1	0	1	0	1	1	PRCTL error
172 to 187									Reserved
188	1	0	1	1	1	1	0	0	Protocol error
189	1	0	1	1	1	1	0	1	Command not authorized ¹⁹
190	1	0	1	1	1	1	1	0	Command unknown
191	1	0	1	1	1	1	1	1	Command not implemented

If bit 6 is set to 1, the interrupt received does not have an end of command status, but a status related to an event interrupt.

If bit 7 is set to 1 and bit 6 is also at zero, the preceding command executed with no errors.

¹⁹This code indicates the use of a reserved command.

IX. Evaluation Sheet

We appreciate your comments and suggestions for improving the quality and ease of use of our documentation.

We would be grateful if you could take a few moments to fill out this evaluation sheet and return it to us. Thanks in advance.

COMPANY:		Telephone:
User:		Position:
Address:		
Zip Code:	City:	

Clearly indicate the version of the MCX card, the software and the documentation:

MCX Card version	<input type="text"/>
MCX- <i>Lite/S</i> Card version	<input type="text"/>
MCX- <i>Lite/570</i> Card version	<input type="text"/>
Document version	<input type="text"/>
Basic software version	<input type="text"/>
Multiprotocol software version	<input type="text"/>

Please give a grade from 0 (worst) to 10 (best) for each of the following characteristics:

Format
Clear

Presentation
Precise

Organization
Explanations

Errors :

If you find any errors in this document, please note the page number and describe the type of error, or write the modifications on the pages in question and send photocopies of those pages along with this report.
