ACCUTRACK PRODUCT SOLUTIONS

DataTrain VIII Code for the Alstom CenTraCode II (CTC2) System

User Manual

WARNING

Read and understand this manual and all included warnings before using this equipment.

Failure to follow the instructions presented in this manual could result in property damage, injury, and/or death.

User Manual
P2346E

ACCUTRACK PRODUCT SOLUTIONS

DataTrain VIII Code for the Alstom CenTraCode II (CTC2) System

User Manual

WARNING

Read and understand this manual and all included warnings before using this equipment.

Failure to follow the instructions presented in this manual could result in property damage, injury, and/or death.

User Manual
Alstom Signaling Inc.

P2346E, Rev. I, March 2015, Printed in U.S.A.

LIST OF EFFECTIVE PAGES

P2346E, DataTrain VIII Code for the Alstom CenTraCode II (CTC2) System User Manual Manual

ORIGINAL ISSUE DATE:	November 2001
CURRENT REVISION AND DATE:	Rev I, March 2015
PAGE	CHANGE OR REVISION LEVEL
Cover	Mar/15
Title page	Mar/15
Preface	Mar/15
i through viii	Mar/15
1-1 through 1-18	Mar/15
2-1 through 2-30	Mar/15
3-1 through 3-24	Mar/15
4-1 through 4-28	Mar/15
5-1 through 5-30	Mar/15
A-1 through A-6	Mar/15
B-1 through B-10	Mar/15

PREFACE

NOTICE OF CONFIDENTIAL INFORMATION

FOR QUESTIONS AND INQUIRIES, CONTACT CUSTOMER SERVICE

Address:	Alstom Signaling Inc. 1025 John Street West Henrietta, NY 14586 USA
Website:	www.alstomsignalingsolutions.com
Email:	websiteinfo@alstomsignalingsolutions.com
Phone:	1-800-717-4477

REVISION LOG

Revision	Release Date	Description of Change	Author	Checker	Approver
1(A)	June 1995	Original Issue	PD	PD	PD
2(B)	January 1998	Content Updated	PD	PD	PD
3(C)	September 2000	Content Updated	PD	PD	PD
4(D)	August 2001	Content Updated	PCH	JAR	NI
5(E)	November 2001	Major rewrite of Aug/01 revision to improve organization and content.	PCH	JAR	NI
6(F)	February 2002	Hexadecimal notation (not BCD) is used in A7/ 00 command.	PCH	JAR	NI
7(G)	March 2011	Added DT8 over Ethernet and CSEX4/NVSP content, removed CTC2s and CTC2v.	MAS	RH	NI
8(H)	August 2012	Added Special Message Bit 5 and reconciled engineering comments.	JF/LR/MN	MG	
I	March 2015	Updated §3.2.2 and added Appendix B	SG	MG	MS

ABOUT THIS MANUAL

The information in this manual is arranged into sections. The title and a brief description of each section follow:

Section 1 - GENERAL DESCRIPTION: This section provides general information about the DataTrain VIII (DT8) protocol.

Section 2 - CAAPE PACKAGE: This section describes how the DT8 protocol is incorporated into a non-vital application using Alstom's CAAPE software package.

Section 3 - THEORY OF OPERATION: This section describes the DT8 modes of operation.

Section 4 - DIAGNOSTIC AND USER INTERFACE: This section discusses the protocol-specified diagnostic functionality implemented by the DT8 protocol.

Section 5 - HARDWARE DESCRIPTION AND TROUBLESHOOTING GUIDE: This section describes the hardware platforms on which DT8 operates, and troubleshooting information.

Appendix A - CENTRACODE II SAMPLE CAAPE INPUT FILE: This section contains an input file to illustrate the inclusion of DT8 in a CenTraCode II non-vital application.

Appendix B - NON-VITAL ETHERNET NETWORK COMMUNICATION - BACKGROUND AND SETUP: This section provides background on TCP/IP networking and describes how to use CAAPE to configure non-vital network communication (using DT8, for example).

MANUAL SPECIAL NOTATIONS

In the Alstom manuals, three methods are used to convey special informational notations. These notations are warnings, cautions, and notes. Both warnings and cautions are readily noticeable by boldface type and a box around the entire informational statement.

Warning

A warning is the most important notation to heed. A warning is used to tell the reader that special attention needs to be paid to the message because if the instructions or advice is not followed when working on the equipment then the result could be either serious harm or death. The sudden, unexpected operation of a switch machine, for example, or the field maintainer contacting the third rail could lead to personal injury or death.

An example of a typical warning notice follows:

WARNING **DISCONNECT MOTOR ENERGY**
Disconnect the motor energy whenever the gear cover is removed. Otherwise, the switch machine may operate unexpectedly and can cause injury and/or death.

Caution

A caution statement is used when failure to follow the recommended procedure could result in loss or alteration of data. A typical caution found in a manual is as follows:

CAUTION
Turn power off before attempting to remove or insert PC boards into a module. PC boards can be damaged if power is not turned off.

Note

A note is normally used to provide minor additional information to the reader to explain the reason for a given step in a test procedure or to just provide a background detail. An example of the use of a note follows:

Note: This step should be done first to validate the correct information is used.

TABLE OF CONTENTS

TABLE OF CONTENTS

Topic **Page**

TABLE OF CONTENTS

Topic **Page**

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

LIST OF TABLES

SECTION 1 – GENERAL DESCRIPTION

1.1 SCOPE OF MANUAL

This section provides general information about the DataTrain VIII (DT8) protocol.

This manual describes the DT8 protocol and how it operates when it is deployed across Alstom wayside non-vital processing components. This manual also provides procedures to help diagnose and troubleshoot issues with the protocol which may occur when used in conjunction with these non-vital processing components.

WARNING

NON-VITAL COMMUNICATIONS SOFTWARE IS NOT FAIL-SAFE

The non-vital communications software is not designed for fail-safe application and must not be used for safety-critical operations.

Failure to comply can degrade the safety performance of the train control system resulting in property damage, injury, and/or death due to train collision or derailment.

1.2 INTRODUCTION

DT8 is a data communication protocol that enables information to move efficiently to and from a control office and field locations using a point-to-point data link or a switched IP network. An application engineer designs and configures the parameters of the DT8 protocol using Computer-Aided Application Programming Environment (CAAPE), a software program manufactured by Alstom Signaling Inc. The application engineer can then configure DT8 for a number of wayside non-vital processing components and assign it to a specific communication path.

The variable length message structures supported by DT8 maximize the efficiency of the communication channel by allowing only the parameters that changed to be transmitted. In addition, DT8 protocol supports full bit map data image transmission.

A Cyclic Redundancy Check-16 polynomial (CRC-16) provides error detection for messages. DT8 protocol acknowledges messages received with a correct CRC-16. If no response is received from the original device, DT8 assumes there is an error in transmission and re-transmits the message.

Each character in an asynchronous serial message consists of the following components:

- a start bit
- a parity bit (optional)
- seven or eight data bits called a byte
- one or two stop bits

DT8 supports the following message types:

- Subcommand
- Acknowledge
- Poll
- Change
- Bitmap Request
- Bitmap and Text

The Subcommand message type requests and sends time and date synchronization information. Each message type has its own variation in structure, but all include a station address and a termination byte. Subcommand, Change, Bitmap, and Text messages also include a CRC-16 for the data portion of the message.

The Maintenance Access (MAC) port on the supported hardware platforms provides all access needed to the code unit. A menu-driven command interface is provided to clearly identify available commands and options.

1.3 SOFTWARE OVERVIEW

Wayside non-vital processing components, such as CenTraCode II, are emulators that directly replace code equipment and provides the data path to move controls and indications between office and field and vice versa.

The embedded wayside non-vital processing component software provides a foundation for field input/output (I/O) operations, serial communication, data logging and diagnostic functions. The software allows for real-time, concurrent operation of multiple communication protocol emulations such as DT8.

The software incorporates an embedded diagnostic program that provides menu-driven analysis of DT8 protocol communication. The maintainer can access the diagnostic tools by connecting a VT-100-compatible terminal to the MAC port located on the Alstom wayside non-vital processor. The CAAPE software package provides the user with flexibility in the handling of controls and indications. By employing simple Boolean equations and/or advanced programming features, the user can pre-process controls from the office and indications from the field. The user can validate, distribute, or share controls among physical outputs. Users can also send indications to the office or transfer them as controls to outputs.

Using the CAAPE, the application engineer can specify and configure message framing, data transmission rates, station addresses, number of control/indication bits and mode of operation. For each field station, the application engineer uses the CAAPE to define site-specific message buffers. Refer to SECTION 2 –CAAPE Package for CAAPE details.

DT8 timing and configuration parameters are adjusted by using the Local Port Configuration (LPC) editor provided with the CAAPE. No mechanical adjustments are needed.

1.4 SUPPORTED HARDWARE PLATFORMS

DT8 can be configured on any of the available non-vital serial (NVS) ports provided by the wayside non-vital processing component. DT8 can also be configured for non-vital serial over Ethernet ports (NVSoE). For a complete list of ports and their functions, see the system manual for the corresponding platform.

Table 1–1 identifies some of the compatible supported platforms.

Table 1–1. DT8 Compatible Supported Platforms

Platform	Description
VPI	VPI can be composed of the following processor boards: • CSEX, CSEX2, or CSEX3 non-vital processor board (CSEX = Code System Emulator eXtended) Refer to Alstom Publication P2086B Volumes 1 and 2 for hardware information.
VPI II	VPI II can be composed of the following processor boards: • CSEX3, or CSEX4 non-vital processor board (CSEX = Code System Emulator eXtended) Refer to Alstom Publication P2511B, Volumes 1 through 5 for hardware information.
iVPI	iVPI can be composed of the following processor boards: • NVSP non-vital processor board (NVSP = Non Vital System Processor) Refer to Alstom Publication P2521B, Volumes 1 through 5 for hardware information.

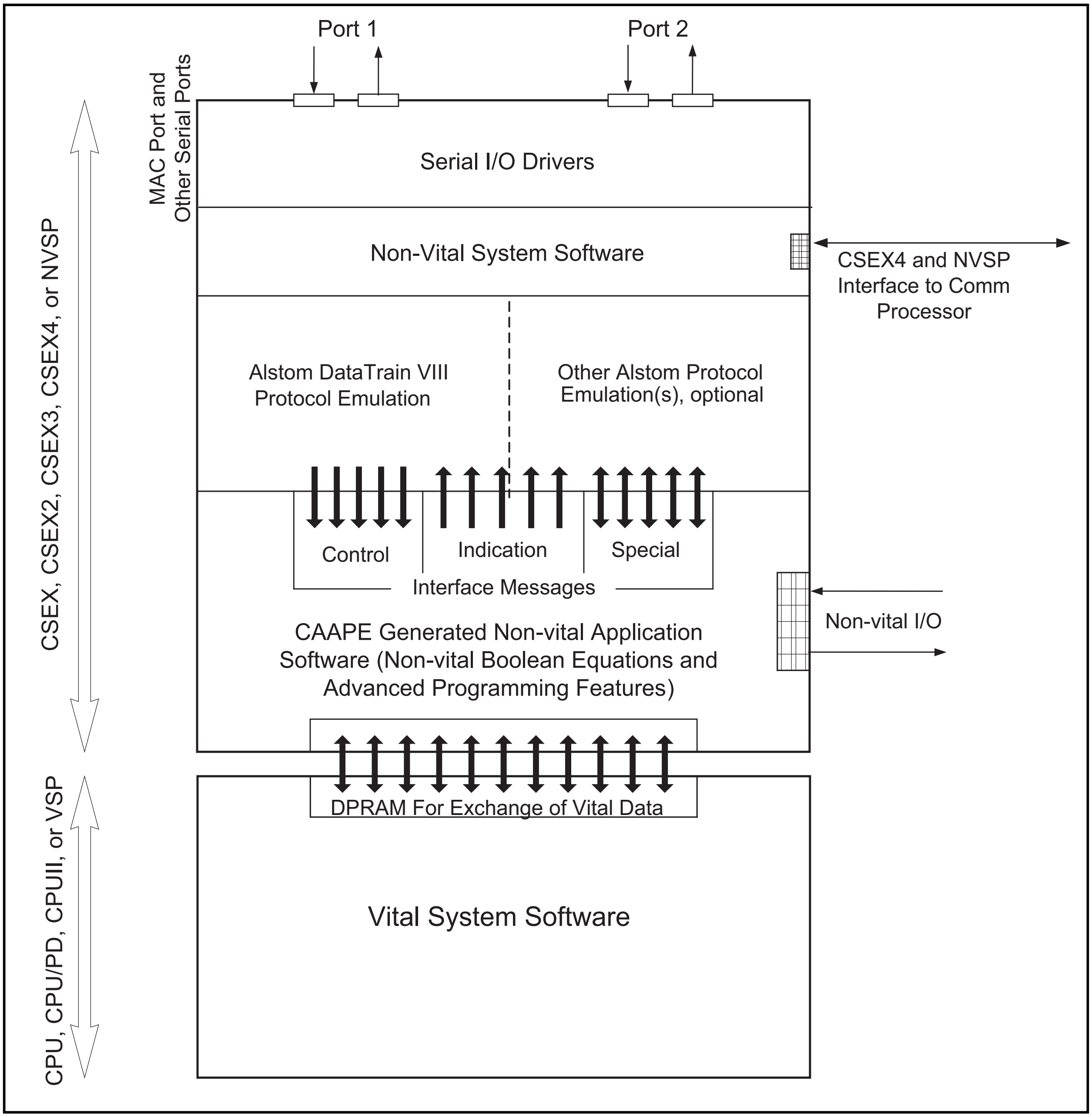


Figure 1–1. Non-Vital Processor System Structure

1.5 PROTOCOL SETTINGS AND LIMITATIONS

See Table 1–2 for the DT8 protocol configuration parameters.

Table 1–2. Configuration Parameter Settings for DT8 Protocol

Parameter	Setting
Message Framing	Parity: Even/Odd/None Stop Bits: 1 or 2
Data Transmission Rate	75 to 19,200 bits per second (up to 57,600 baud on CSEX3, CSEX4 and NVSP boards)
Maximum Number of Unique Station Addresses	255
Maximum Number of Control or Indication Bits per Message	1000 bits, limit imposed by CAAPE and protocol
Duplex	Full Duplex; adaptable to Half Duplex in Master/ Slave mode only with proper use of Request to Send (RTS) and Clear to Send (CTS)

Note: The asynchronous serial parameters such as frame format, baud rate and flow control are not applicable for non-vital over Ethernet serial port virtual serial connections.

1.6 COMMON ABBREVIATIONS AND GLOSSARY

This section contains definitions of the terms used in this manual. Terms and abbreviations used throughout this manual are provided in Table 1–3.

Table 1–3. Common Abbreviations and Glossary

Term	Definition
Acknowledge	To reply in a predetermined manner to the reception of a valid message, usually used in reference to a communication protocol.
Address	A unique numeric identifier for a station in a communication network; a specific location in RAM.
ANSI	Abbreviation for American National Standards Institute, an organization whose purpose is to set voluntary industry standards.
Application (Program or Logic)	A set of Boolean and conditional program code expressions (or instructions) that manipulate communication data and system inputs and outputs at a specific location in a communication network.
Assert	To set a signal to its true or on state.
Asynchronous	Transmission of data in which character synchronization is established by framing a unit of data with start and stop bits. The time between characters can be variable.
Backplane	Wiring connections made to the back of a CenTraCode II circuit board.
Baud (Rate)	A unit of signaling speed equal to the number of discrete conditions or signal events per second.
BBRAM	Abbreviation for Battery-Backed Random Access Memory, memory that is preserved by an on-board battery in the event of external power loss.
Binary	A number base containing only two digits, 0 and 1, and commonly used to express data as stored internally by a computer.
Bit	The smallest unit in a computer’s memory whose value is either zero or one; stands for binary digit.
Bitmap	A complete mapping of all bits in a control or an indication message. The value of each bit (or parameter) is either true (1) or false (0).
Broadcast (Message)	A message sent by a communication protocol intended for all remote stations, none of which should reply. Typically, a broadcast message uses a station address of zero.

Table 1–3. Common Abbreviations and Glossary *(Cont.)*

Term	Definition
Buffer	A block of RAM locations holding a related set of information.
Byte	A sequence of adjacent binary digits (usually eight) operated on as a unit.
CAAPE	Alstom’s Computer-Aided Application Programming Environment.
CenTraCode II	Alstom’s non-vital communications processor boards: CSEX, CSEX2, CSEX3, CSEX4, and NVSP.
Checksum	A value mathematically computed for (and appended after) a packet of data, and regenerated and verified by the receiver to more likely ensure correct message transmission; sometimes called a checkword.
Clear	To assign a false value (zero) to one or more parameters or flags.
Clock Signal	In synchronous communication, a square-wave signal of a fixed frequency used to time each bit of transmitted and received messages.
Code	A series of software instructions executed by the CPU; synonymous with Program.
Code-line	A communication link over which controls and indications are exchanged.
Code Unit	A device, specifically the CTC2, whose primary purpose is the efficient transfer and processing of serial communication data (controls and indications).
Command	The portion of a communication protocol’s serial message (usually near the start) that defines the message type.
Compile	In reference to Alstom’s CAAPE software package, a user-initiated process in which the CAAPE evaluates logic statements in a non-vital application and produces machine-readable code for programming into an EPROM for execution by the CPU.
Component	In reference to Alstom’s CAAPE software package; a separate, distinct unit that comprises a portion of a non-vital application; a portion of hardware on a circuit board.
Configuration	A specific combination of user-defined settings.
Control	Any message received at a communication protocol’s serial port (an incoming message). Contrast with Indication.
Corruption	The mutilation or perturbation of data caused by a hardware or software failure.

Table 1–3. Common Abbreviations and Glossary *(Cont.)*

Term	Definition
Counter	An upwards-counting value, often maintained by protocol emulation, used to report the number of times an event occurred, such as reception of a certain message type.
CPU	Abbreviation for Central Processing Unit, the computer section that handles the actual processing of computer instructions.
CRC	Abbreviation for Cyclic Redundancy Check, an error checking technique used to ensure the accuracy of transmitted serial data over a communication channel.
CSEX	Abbreviation for Code System Emulator Extended, a group of Alstom non-vital communications processor boards, part of the CenTraCode II-v family of circuit boards. Includes the original CSEX (sometimes referred to as CSEX1 or CSEX[1]), CSEX2, CSEX3, and CSEX4.
CTC2	Abbreviation for CenTraCode II system hardware or software.
CTS	Clear To Send, a hardware control signal used by DCE to indicate it is ready to accept data from DTE; typically paired with RTS.
Current Loop	A means of communicating serial data via the presence or absence of current in a 2-wire cable.
Data	The value of one or more bits of information.
Data Communications Equipment (DCE)	Interface equipment (e.g., a modem) used to couple Data Terminal Equipment (DTE) into a transmission channel.
Data Logging	The recording of selected data in the processor’s battery-backed memory (BBRAM), usually on the basis of change of state to one or more application parameters, for later evaluation by the user.
Data Terminal Equipment (DTE)	Typically a computer (specifically CenTraCode II) or a terminal connected to Data Communications Equipment (DCE).
De-assert	To set a signal to its false or off state.
Decimal	A number base containing ten digits, 0 through 9.
Destination	The recipient of a serial message. Contrast with Source.
Digit	A single numeric character.
Disable	To disallow, turn off or otherwise deactivate.
DPRAM	Abbreviation for Dual Port Random Access Memory, a memory device having bi-directional data and address connections suitable for low-level communications.

Table 1–3. Common Abbreviations and Glossary *(Cont.)*

Term	Definition
DUART	Abbreviation for Dual Universal Asynchronous Receiver/ Transmitter, a device containing two asynchronous-only serial communication ports.
Duplex	See Full Duplex or Half Duplex.
DUSART	Abbreviation for Dual Synchronous-Asynchronous Receiver/ Transmitter, a device containing two serial communication ports settable for synchronous or asynchronous operation.
Dynamic	To update information as it occurs (in real time).
EEPROM	A device containing Electrically Erasable Programmable Read-Only Memory that holds its contents without power.
Embedded	Software contained within a sub-component of a processor board, usually stored on a programmable memory device such as a PROM and containing code executed by the CPU.
Emulation (Software)	A computer program that mimics existing hardware or another software module to assure compatibility between two or more naturally different pieces of equipment, usually used in reference to a software implementation of a communication protocol.
Emulator	A device designed to perform the function of another.
Enable	To allow, turn on or otherwise activate for use.
EPROM	Abbreviation for Erasable/Programmable Read-Only Memory, a device that holds its contents without power and typically contains firmware.
Event	A notable occurrence, usually the result of a change to a system parameter.
Execute	To perform the instructions in a software program.
Field Location	A set of equipment (e.g., CenTraCode II) controlled from the office and located along a railroad, whose purpose is to operate a specific section of the railroad.
FIFO	First In First Out
Firmware	Computer instructions stored on a programmable memory device, such as an EPROM.
Flag	A location in a computer’s memory used to pass information between two software programs, such as the protocol emulation and the non-vital application logic. Flags are used to indicate the current status of a system buffer (such as “buffer in use”).

Table 1–3. Common Abbreviations and Glossary *(Cont.)*

Term	Definition
Flash Device	A memory device whose contents (code and data) can be modified by the CPU by following a well-defined and secure series of steps, and are retained without power.
Frame	Information bracketed around a unit of data to signal its start and end. In asynchronous transmission, a frame consists of a start bit, data bits, an optional parity bit and one or more stop bits.
Full Duplex	A serial communication mode in which networked devices can send data to each other at the same time.
Half Duplex	A serial communication mode in which only a single networked device can send data at a given time.
Handshaking	The use of hardware electrical signals (typically Request To Send, RTS, and Clear To Send, CTS) to control the flow of serial data, typically through a modem.
Hardware	Any of the machinery the makes up a digital computer installation; includes the circuit boards therein.
Header	A hardware component containing multiple pins (wires) onto which jumpers can be placed for board configuration.
Hexadecimal	A number base containing sixteen digits, 0 through 9 and A (10) through F (15), commonly used to represent bytes of data stored in a computer.
HHT	Abbreviation for Hand Held Terminal, a small video display device that provides limited system diagnostics.
Indication	Any message sent out a communication protocol’s serial port (an outgoing message). Contrast with Control.
Initialization	A process usually performed at startup that resets all of a system’s operating parameters to preset default values.
Input/Output (I/O)	The process of transmitting information from an external source such as counters, switches and addresses to a system, or from a system to an external source.
Install	The act of assigning a communication protocol to a specific CenTraCode II serial port, usually done as part of the design of a non-vital application.
Intel-Hex Address Notation	A specific format (ssss:oooo) for entering a system memory address, wherein the address is expressed in two parts: the segment (ssss) and the offset (oooo). For example, the physical hexadecimal address C045A is entered as C045:000A.

Table 1–3. Common Abbreviations and Glossary *(Cont.)*

Term	Definition
Interface	The equipment that enables one kind of hardware to be recognized and processed by another kind of hardware.
Interrupt	A computer instruction that causes the computer to stop the current process and perform a time-sensitive task before resuming the interrupted process.
Invoke	To activate a function, process, program or routine.
iVPI	Abbreviation for integrated Vital Processor Interlocking, part of Alstom’s family of Vital and non-vital interlocking processors.
Jumper	A device that closes a circuit, specifically a short length of wire or a plastic covered metal block that is pushed onto two pins to complete the circuit.
Latched	A mode of operation for a circuit in which an output, even a momentary one, is maintained “on”.
Log	To record an event in system memory, often with a time-stamp.
LPC File	A Local Port Configuration File containing user-defined settings for a communication protocol assigned to a serial port on a CenTraCode II board.
MAC	Abbreviation for Maintenance Access, and refers to the MAC port connector on the CenTraCode II processor boards.
Menu	A set of user options offered on a single CenTraCode II diagnostics screen.
Menu Selection	A choice made by the user from one of several menu options.
Message	A group of data referred to as a single unit, especially with reference to a control or an indication.
Microprocessor	See CPU.
Modem	A device that modulates and demodulates signals transmitted over data-communication facilities (e.g., telephone lines).
Modulus	The integer remainder of the division of two integer values. The percent sign (%) is used to represent the modulus operator.
Monitor	The act of observing message traffic, or a device used for this purpose.

Table 1–3. Common Abbreviations and Glossary *(Cont.)*

Term	Definition
Multi-Tasking	The running of two or more programs (tasks) by one computer seemingly at the same time. Usually each task relinquishes control of the computer after some short period of time to allow the next task to run, or each task is run for a given time slice and is then automatically suspended.
Noise	An unexpected and undesirable signal on a communication port's data line.
Non-vital	A process whose function does not affect the safety of train operation. Contrast with Vital.
NVS	Non-vital serial: a communication channel via an RS-232/422/485 asynchronous serial link.
NVSoE	Non Vital Serial over Ethernet: a virtual communications channel via a TCP/IP connection (single or redundant).
NVSP	Abbreviation for Non Vital System Processor, the non-vital processor board used in the iVPI system.
Office	The control center for one or more field locations along a railroad.
Operating Mode	A protocol setting that determines its behavior with respect to transmitted and received messages. For example, Master/Slave Mode or Peer Mode.
Parameter	In reference to Alstom's CAAPE software package, an application-specific entity whose value can vary over time depending upon the current state of related variables.
Parity	The use of a check bit, appended to each frame of asynchronous data, for error checking purposes.
PC	Abbreviation for Personal Computer.
Pending	Waiting to be processed, especially in reference to a change to an indication data parameter resulting in the transmission of an indication message.
Poll	To query a remote station for current information, usually used in reference to a communication protocol.
Port	A portion of a processor board used for the transfer of digital data (messages) in a serial manner; an access point for a device where energy can be applied or withdrawn, or where the device can be observed or measured.

Table 1–3. Common Abbreviations and Glossary *(Cont.)*

Term	Definition
Post	The act of presenting a message (usually a control or an indication) to either the non-vital application logic or the protocol emulation for processing by the recipient.
Processor	See CPU.
Program	A series of software instructions for the computer (CPU) to follow. Synonymous with Code.
Protocol	Rules that define how two or more communication devices “talk” to each other. A formal set of conventions governing the format and relative timing of message exchange between two communication devices.
PROM	Abbreviation for Programmable Read-Only Memory, an integrated memory device that can be programmed with computer instructions or data.
Queue	A buffer in the system’s memory (RAM) used to chronologically hold multiple messages of the same type (for example, an indication queue).
RAM	Abbreviation for Random Access Memory, and used by a computer for storing data that can change. RAM can be written to or read by the computer.
Real-Time	In a communication system, the processing and use of data at time of occurrence and within a given time frame.
Record	In reference to Alstom’s CAAPE software package, an input statement with a specific format and options.
Reset	To change a bit’s value to zero or an output to an inactive (off) condition (see Clear); to restart a process, as in restarting a processor board.
Round-Robin Task Loop	A multi-tasking technique wherein each task is sequentially processed and purposely relinquishes control of the CPU after some short (and typically variable) period of time.
RS-232	An industry standard for a 25-pin communication interface that connects computers and various forms of peripheral equipment.
RS-422, RS-423 or RS-485	Standards for serial transmission that extends distance and speed beyond the RS-232 standard.
RTS	Request To Send, a hardware control signal used by DTE to control the flow of serial data through a communication port; typically paired with CTS.

Table 1–3. Common Abbreviations and Glossary *(Cont.)*

Term	Definition
Set	To assign a true value (one) to one or more parameters or flags.
Serial Interface	A data channel (usually a serial port) that transfers digital data one bit after another.
Site-Specific	A set of instructions or operating parameters appropriate for a certain field location.
Software	Programs used in operating a digital computer.
Source	The originator of a serial message. Contrast with Destination.
SRAM	Abbreviation for Static Random Access Memory.
Stack	An area of system memory (RAM) used by a subroutine for the temporary storage of information.
State	The current true or false value of an application parameter or device.
Station	A uniquely-addressed location in a communication network, containing a set of site-specific controls and indications.
Subroutine	A portion of a computer program that carries out a specific operation or processing function.
Switch Block	A hardware component containing multiple user-settable switches for board configuration.
Synchronous	Transmission of serial data in which clocking information is transmitted along with the data. Data are sent at a defined rate controlled by a timing source at the transmitter.
System Software	The full set of embedded operating code on a CenTraCode II processor board.
Task	A computer program that is run as an independent unit (see Multi-Tasking).
TCP/IP	Abbreviation for Transmission Control Protocol/Internet Protocol, a connection oriented protocol.
Terminal	A video display device (see VT100 Terminal).
Timer	A specific fixed time constant (often user selectable) used by the CenTraCode II System Software or the protocol emulation to time an event, such as the assertion of RTS.
Traffic	Transmit and receive activity (messages) occurring on a communication port over a period of time.

Table 1–3. Common Abbreviations and Glossary *(Cont.)*

Term	Definition
UART	Abbreviation for Universal Asynchronous Receiver/Transmitter, a device containing one asynchronous-only serial communication port.
UDP	Abbreviation for User Datagram Protocol, a connectionless Internet Protocol.
Unlatched	A mode of operation for a circuit in which an output remains “on” for only a moment.
USART	Abbreviation for Universal Synchronous-Asynchronous Receiver/ Transmitter, a device containing one serial communication port settable for synchronous or asynchronous operation.
Vital	A process whose function affects the safety of train operation. Contrast with Non-vital.
VPI	Abbreviation for Vital Processor Interlocking, Alstom’s family of Vital and non-vital interlocking processors.
VT100 Terminal	A video display device that employs a standard set of instructions (ANSI escape sequences) for displaying information.
Watchdog Timer	An internal timer, usually a hardware device, used to detect a possible malfunction of the software and to initiate an automatic restart of the system.
Word	A pair of adjacent bytes.

1.7 RELATED PUBLICATIONS

For further information on the wayside non-vital processor components hardware and configuration as well as other available emulations and protocols, refer to the Alstom publications listed in Table 1–4.

Table 1–4. Related Publications

Document No.	Title
P2086B	VPI® Vital Processor Interlocking Control System Operation and Maintenance Manual (Volumes 1 and 2)
P2346	Series CenTraCode II Protocol Emulation Manuals
P2509	Maintenance Management System (MMS) User's Manual
P2511B	VPI® II Vital Processor Interlocking II Control System Operation and Maintenance Manual (Volumes 1 through 5)
P2512A	Computer-Aided Application Programming Environment (CAAPE) Software Package User's Manual
P2512D	VPI® CAA Reference Manual
P2512E	DataLogger Event Recording Utility for CenTraCode and VPI® Systems User's Manual
P2512F	iVPI™ CAA Reference Manual
P2521B	iVPI™ integrated Vital Processor Interlocking Control System Operation and Maintenance Manual (Volumes 1 through 5)

THIS PAGE INTENTIONALLY LEFT BLANK.

SECTION 2 – CAAPE PACKAGE

2.1 GENERAL

This section describes how the DT8 protocol is incorporated into a non-vital application using Alstom's CAAPE software package.

The CAAPE allows the application engineer to specify and configure DT8 message framing, data transmission rates, station addresses, number of control/indication bits and mode of operation.

For more detailed information about entering records in the CAAPE, refer to the CAAPE's online help. Appendix A provides examples of typical CAAPE input files.

2.2 CAAPE RECORDS

The DT8 timing and configuration parameters are adjusted by using the Local Port Configuration (LPC) editor described later in this section.

The top-level input file for a CenTraCode non-vital application has a three-character filename extension (CSI). This file is used to document the non-vital program and revision history, and to list all the files to be included as part of the non-vital application.

The CAAPE records discussed in this section are part of the Non-vital Serial (NVS) Communication and Non-vital Serial over Ethernet (NVSoE) Communication sections of a non-vital application.

- NVS is used to configure data communication through the board's serial ports.
- NVSoE is used to configure data communication through the virtual serial connections supported by the CSEX4/NVSP platform. Virtual serial connections also require that a number of network settings be specified using additional records in other sections.

Note: Network configuration records are outside the scope of this manual, but can be found in the appropriate VPI (P2512D) or iVPI (P2512F) CAA reference manual.

The following record marks the start of the NVS Communication Section, and is usually placed at the start of the application's serial communication file whose extension is CSS (formerly COM with earlier versions of the CAA compiler).

```
SERIAL COMMUNICATIONS SECTION
```

or

```
CSEX CODE SYSTEM SECTION
```

The second record format applies to a specific hardware platform (CTC2 in a VPI) and is functionally equivalent to the NVS Communication Section record.

The following record marks the beginning of the NVSoe Communication Section, and is usually placed at the start of the application network serial communications file (with extension .NSS):

```
NETWORK SERIAL COMMUNICATIONS SECTION
```

The CAAPE records that follow comprise an application's NVS Communication Section or NVSoE Communication Section. When a record offers more than one choice, they are listed with each choice shown in quotation marks. In this case, use the desired choice exactly as shown herein, but without the quotation marks.

2.3 SERIAL PORT OR NETWORK PORT NUMBER AND PROTOCOL EMULATION TYPE

The SERIAL PORT or NETWORK PORT record defines the use of a CenTraCode II serial port for code system emulation or serial communications. This record also identifies the type of code system protocol to be emulated (in this case DT8), or the type of serial communications to perform, and can specify certain options available for the communication protocol. It is the first in a series of records that define the configuration and messages of the serial port. One SERIAL PORT record is required for each serial port used in the CenTraCode II system.

```
SERIAL PORT port = TYPE (type), options
```

or

```
NETWORK PORT port = TYPE (type), DEVICE (device), options
```

See Table 2–1 for descriptions of each record option.

Table 2–1. Serial/Network Port Record Descriptions

Option	**Description**
port	The number of the serial port to be used for receiving and transmitting DT8 messages; it is a 1-digit number from 1 to 5 (or 1 to 3 with CSEX4/ NVSP).
type	The name of the code system to be emulated on the port, or protocol for serial communications with another device. The following DT8 types are supported: • "DT8", "DT8 CODE", or "DT8 CODE SLAVE" – DT8 Slave Mode with default unlatched controls • "DT8 CODE MASTER" – DT8 Master Mode with default unlatched controls • "DT8 SLAVE" – DT8 Slave Mode with default latched controls • "DT8 MASTER" – DT8 Master Mode with default latched controls • "DT8 PEER" – DT8 Peer Mode with default latched controls • "DT8 SYNC" – DT8 Sync Mode with default latched controls

Table 2–1. Serial/Network Port Record Descriptions *(Cont.)*

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>device</td><td>The network device to be used: ENET1, ENET2 or REDUNDANT for path-redundant MMS links using both network devices.

The default for latched vs. unlatched controls applies if the options parameter is omitted. See Section Modes of Operation for detail on the DT8 operating modes.</td></tr>
<tr><td>options</td><td>These options are specific to the DT8 protocol and are used to override the default latching of controls:
• “UNLATCHED CONTROLS” – All controls for this port are automatically cleared once per application cycle
• “LATCHED CONTROLS” – Controls are <u>not</u> automatically cleared and the state of control bits is retained by DT8.

Examples:

<code>SERIAL PORT 1 = TYPE (DT8)</code>

<code>SERIAL PORT 2 = TYPE (DT8 PEER)</code>

<code>SERIAL PORT 3 = TYPE (DT8 MASTER), UNLATCHED CONTROLS</code>

<code>NETWORK PORT 1 = TYPE(DT8 SYNC), DEVICE (ENET1), LATCHED CONTROLS</code>

Typically, the SERIAL PORT or NETWORK PORT record is immediately followed by records describing the protocol configuration file, operating mode, default baud rate, and other characteristics.</td></tr>
</table>

2.4 MAC/TCP PANEL NETWORK PORT

The MAC/TCP PANEL PORT record is used to define a special network port for communicating panel messages to MMS with the MAC/TCP protocol. It is typically followed by one control and/or one indication message.

2.5 SERIAL, NETWORK, MAC/TCP PORT CHARACTERISTICS

The SERIAL PORT, NETWORK PORT, or MAC/TCP PORT record is immediately followed by records describing the protocol configuration file, operating mode, default baud rate, and other port characteristics.

2.5.1 Serial, Network, MAC/TCP Port Records

See Table 2–2 for descriptions of the optional records that describe describing the protocol configuration file, operating mode, default baud rate, and other port characteristics.

Table 2–2. Serial, Network, MAC/TCP Port Record Descriptions

Record	Description
Configuration File	This record specifies the name of a user-created configuration file containing DT8-specific settings used to override default settings assigned by the CAAPE. See Section 2.7 for details on default protocol settings and creating a protocol configuration file. `CONFIGURATION FILE = `*`filename`* *filename* is the complete path and filename of the user configuration file, usually with an LPC extension. For example: `CONFIGURATION FILE = D:\CONFIG\DT8_USER.LPC`
Operating Mode	This record determines the electrical mode of serial port 1 or 2 of the CSEX3/CSEX4/NVSP board. This record is only valid for serial ports 1 and 2 of a CSEX3/CSEX4/NVSP board (or a warning is generated when the application is compiled, and the record is ignored). If this record is omitted, the default operating mode is RS-232. This record is not used in the Network Serial Communications section. `OPERATING MODE  = `*`type`* *type* specifies the type of electrical connection for serial port 1 or 2 of the CSEX3/CSEX4/NVSP board. Valid entries are “RS-232”, “RS-422” and “RS-485”. For example: `OPERATING MODE = RS-422`

Table 2–2. Serial, Network, MAC/TCP Port Record Descriptions *(Cont.)*

Record	Description
Default Baud Rate	This record identifies the baud rate for serial communications on the port. If this record is omitted, the rate is 1200 baud. This record is not used in the Network Serial Communications section. `DEFAULT BAUD RATE =` *`rate`* *rate* is the baud rate (75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 or 57600). 38400 and 57600 baud are valid for CSEX3/ CSEX4/NVSP boards only (or a warning is generated when the application is compiled, and the rate is set to 9600 baud). For example: `DEFAULT BAUD RATE = 2400`
Baud Rate Control	This record is used to identify a set of non-vital input parameters whose values are used by DT8 to determine the baud rate at run time. When this record is present, these values override all other baud rate specifications. This record is not used in the Network Serial Communications section. `BAUD RATE CONTROL =` *`name-3`*`,` *`name-2`*`,` *`name-1`* *name-1* through *name-3* are the names of input or logic parameters whose values determine the baud rate. Only the baud rates listed in Table 2–3 are available using this record. For example: `BAUD RATE CONTROL = NVI-03, NVI-02, NVI-01`

Data Format	This record identifies the serial data format. If the record is not present, the default data format is 8 data bits, 1 stop bit and odd parity. This record should be used if these defaults are not satisfactory. This record is not used in the Network Serial Communications section.
	
	`DATA FORMAT = databits, stopbits, parity`
	
	databits specifies the total number of data bits in each serial character (7 or 8).
	
	stopbits specifies the number of stop bits in each serial character (1 or 2).
	
	parity specifies the type of parity to use for each serial character (E = Even; O = Odd; N = None)
	
	For example:
	
	`DATA FORMAT = 8, 1, N`
	
	`DATA FORMAT = 7, 2, E`

Table 2–3. Baud Rates Settable Using Baud Rate Control Record

Rate	Code	name-1	name-2	name-3
default	000	FALSE	FALSE	FALSE
300	001	FALSE	FALSE	TRUE
600	010	FALSE	TRUE	FALSE
1200	011	FALSE	TRUE	TRUE
2400	100	TRUE	FALSE	FALSE
4800	101	TRUE	FALSE	TRUE
9600	110	TRUE	TRUE	FALSE
19200	111	TRUE	TRUE	TRUE

2.5.2 Site-Specific Message Definitions

For each field station, use the CAAPE to define the following message buffers:

- A Control buffer for the incoming message.
- An Indication buffer for the outgoing message.
- An optional Special Message buffer to facilitate or control the interchange of data between the DT8 protocol and the non-vital application logic.
- An optional Text Message buffer.

Some operating modes of DT8 allow multiple, uniquely addressed stations to exist on a single serial port. To accomplish this, define a separate set of message buffers for each station having a unique address.

Two methods are provided for defining serial messages and their contents. Both can be used interchangeably in the Serial Communications Section or Network Serial Communications Section of the non-vital application after a port’s SERIAL PORT record:

- Identify an incoming message with a CONTROL record, and an outgoing message with an INDICATION record. The CAAPE processes all such records encountered.
- Use a MESSAGE record to identify each new message, and SOURCE and DESTINATION records to provide a text name for the message sender and receiver. The CAAPE processes a message only if the SOURCE or DESTINATION name matches the name specified in the CSEX (or CTC2) ID record in the application’s CSI file. This method is most useful in a serial communications protocol, such as DT8 PEER, where both ends of the serial link could be CSEX boards. A single Serial Communications file can be provided for all serial links, and data is assigned based on the system currently being compiled.

Note: For virtual serial connections, a specific network port can be designated as SOURCE or DESTINATION by appending its number to the board’s ID name. For example, network port 1 on the CSEX4 board with ID “THIS-CSEX4” can be identified as “THIS-CSEX4 : 1” in a SOURCE or DESTINATION record.

Each item in a message is referred to with the generic term “parameter”. The application engineer assigns a name of 1 to 16 characters to each parameter.

2.5.2.1 Control Definition Record

The Control Definition record marks the start of the definition of an incoming message (control), giving the station address and message length. The CAAPE processes this record and the message data records that follow it.

Below is the syntax for the Control Definition record followed by an example:

```
CONTROL = ADDRESS (address), LENGTH (length)
```

```
CONTROL = ADDRESS (00000110), LENGTH (72)
```

In this example, the address is decimal 6 (six).

See Table 2–4 for descriptions of the Control Definition parameters.

Table 2–4. Control Definition Parameters

Parameter	Description
address	The station address, expressed in binary from 1 to 8 bits, ordered most to least significant from left to right; if omitted, the address is set to zero or to the value read from on-board switches.
length	The total number of parameters (bits) in the control message, normally an even multiple of eight; this is a required field.

2.5.2.2 Control Contents Record

Multiple control message data records follow a CONTROL or MESSAGE record (described later) and name each parameter in the incoming control message. The LENGTH specified in the preceding CONTROL or MESSAGE record determines the total number of control message data records. One control message data record is required for each parameter in the control message.

Below is the syntax for the Control Contents record followed by some examples:

```
pos = name

1 = INPARM1

2 = PERMZERO
```

See Table 2–5 for descriptions of the Control Contents parameters.

Table 2–5. Control Contents Parameters

Parameter	Description
pos	The position of this parameter in the control message; control message data must be numbered and named sequentially starting with position 1 (one).

name	The name assigned to the parameter (usually a true/false logic parameter) for this position (*pos*) in the control message for use in the non-vital application; if this position is a spare bit, it can be assigned the system reserved name PERMZERO. For example:
	`1 = INPARM1`
	`2 = PERMZERO`

2.5.2.3 Indication Definition Record

The Indication Definition record marks the beginning of the definition of an outgoing message (indication), giving the station address and message length. The CAAPE processes this record and the message data records that follow it.

Below is the syntax for the Indication Definition record followed by some examples:

```
INDICATION = ADDRESS (address), LENGTH (length)

INDICATION = ADDRESS (01101110), LENGTH (32)

INDICATION = ADDRESS (00000110), LENGTH (72)
```

In the first example, the address is decimal 110.

See Table 2–6 for descriptions of the Indication Definition parameters.

Table 2–6. Indication Definition Parameters

Parameter	Description
address	The station address, expressed in binary from 1 to 8 bits, ordered most to least significant from left to right and usually the same as the CONTROL address; if omitted, the address is set to zero or to the value read from on-board switches.
length	The total number of parameters (bits) in the indication message, normally an even multiple of eight; this is a required field.

2.5.2.4 Indication Contents Record

Multiple indication message data records follow an INDICATION or MESSAGE record (described later) and name each parameter in the outgoing indication message. The LENGTH specified in the preceding INDICATION or MESSAGE record determines the total number of indication message data records. One indication message data record is required for each parameter in the indication message.

Below is the syntax for the Indication Contents record followed by some examples:

```
pos = name

1 = MANUAL-PB

2 = MC-CONTROL

3 = PERMZERO

4 = BOOLARRAY[32]
```

See Table 2–7 for descriptions of the Indication Contents parameters.

Table 2–7. Indication Contents Parameters

Parameter	Description
pos	The position of this parameter in the indication message; indication message data must be numbered and named sequentially starting with Position 1 (one).
name	The name assigned to the parameter (usually a true/false logic parameter) for this position (*pos*) in the indication message for use in the non-vital application; if this position is a spare bit, it can be assigned the system reserved names PERMZERO or PERMONE; Boolean array elements can be used as indication parameters.

2.5.2.5 Defining Control/Indication Messages with SOURCE and DESTINATION Records

The DESTINATION record provides the ID (identification) name of the board that is to receive a message. This record is paired with a SOURCE record for each message defined in the Serial Communications Section. The CAAPE processes the message only if either the SOURCE or DESTINATION name matches the name specified in the CSEX (or CTC2) ID record in the application's CSI file.

Below is the syntax for the Destination record followed by an example:

```
DESTINATION = id-name
```

```
DESTINATION = MAIN CODE UNIT
```

The SOURCE record is similar to the DESTINATION record, except that it identifies the ID name of the board from which the message originates. It is paired with a DESTINATION record for each message defined.

Below is the syntax for the Source record followed by an example:

```
SOURCE = id-name
```

```
SOURCE = STANDBY CODE UNIT
```

See Table 2–8 for descriptions of the Destination and Source parameters.

Table 2–8. Destination and Source Parameters

Parameter	Description
id-name	Must match the 40-character (maximum) board ID supplied on a CSEX (or CTC2) ID documentation record for the board that receives the message.

The MESSAGE record identifies the station address and message length for either an incoming (control) or outgoing (indication) message. This record follows either a SOURCE or DESTINATION record for the message and precedes the message's data records described earlier in this section.

Below is the syntax for the Message record followed by some examples:

```
MESSAGE = ADDRESS (address), LENGTH (length)

MESSAGE = ADDRESS (01010101), LENGTH (32)

MESSAGE = ADDRESS (00101100), LENGTH (200)
```

In the second example, the address is decimal 44.

See Table 2–9 for descriptions of the Message parameters.

Table 2–9. Message Parameters

Parameter	**Description**
address	The station address, expressed in binary from 1 to 8 bits, ordered most to least significant from left to right; if omitted, the address is set to zero or to the value read from on-board switches.
length	The total number of parameters (bits) in the message, normally an even multiple of eight; this is a required field.

2.5.2.6 Special Control Record

The SPECIAL CONTROL record is optional. It defines a Special Message buffer used to transfer information (by way of flags) concerning communication events between the DT8 protocol and the non-vital application. A Special Message Flag can be set to a true state by the application to instruct DT8 to perform a one-time action, such as forcing the transmission of an indication bit map. Conversely, the DT8 protocol sets Special Message Flags to inform the application of events, such as the receipt of a change to controls. Generally the notified system resets the flag after the requested action has been completed.

If used, the SPECIAL CONTROL record must follow the control and indication message definitions for a given station on a DT8 port, and, if multiple stations are defined, precede the next station's message definitions.

Below is the syntax for the Special Control record followed by an example:

```
SPECIAL CONTROL = LENGTH (length)
```

```
SPECIAL CONTROL = LENGTH (24)
```

See Table 2–10 for descriptions of the Special Control parameter.

Table 2–10. Special Control Parameter

Parameter	**Description**
length	The total number of parameters (flags) in the Special Message buffer, either 16 or 24. The *length* must be 16 if using DT8 version F31 or earlier, otherwise it must be 24.

2.5.2.7 Special Message Contents Record

Multiple Special Message data records follow the SPECIAL CONTROL record and name each parameter (flag) in the Special Message buffer. The LENGTH specified in the preceding SPECIAL CONTROL record determines the total number of Special Message data records. One Special Message data record is required for each flag in the Special Message buffer.

Below is the syntax for the Special Message Contents record followed by some examples:

```
pos = name

1 = BIT_MAP_IN

2 = CHANGE_IN

3 = ACK_IN

4 = POLL_IN

5 = MAP_REQ_IN or 5 = CAA ACK

6 = BIT_MAP_OUT

7 = CHANGE_OUT

8 = ACK_OUT

9 = POLL_OUT

10 = MAP_REQ_OUT

11 = STATION_ALIVE

12 = POLL_ENABLE

13 = TEXT_IN

14 = TEXT_OUT

15 = SEND_MAP

16 = PERMZERO

17 = SEND_TIME_REQ

18 = TIME_REQ_IN

19 = SEND_TIME

20 = TIME_IN

21 = CLR_LATCHED_CTL

22 = AUTOCLR_LATCH_CTL

23 = PERMZERO

24 = PERMZERO
```

See Table 2–11 for descriptions of the Special Message Contents parameters.

Table 2–11. Special Message Contents Parameters

Parameter	Description
pos	The position in the Special Message buffer; Special Message data must be numbered and named sequentially starting with position 1 (one).
Name	The name assigned to the Special Message Flag (a true/false logic parameter) for this position (*pos*) in the Special Message buffer; if this position is a spare or is to be unused, it can be assigned the system reserved name PERMZERO.

Table 2–12 describes the Special Message Flags used by DT8. The flag names in the table are recommendations only; the Application Engineer is free to use any appropriate names. The “Set By” and “Cleared By” columns indicate whether the non-vital application (NVA) logic or the DT8 protocol is responsible for setting (to 1) or clearing (to 0) each Special Message Flag.

Special message flags can be set at any time by the DT8 protocol while the application task is running. For example: at the start of the application task flag is false then midway through the application the flag could go true.

See SECTION 3 –Theory of Operation for details on the various DT8 message types and modes of operation.

Table 2–12. Special Message Flags

Bit	Flag Name	Set By	Cleared By	Description
1	BIT_MAP_IN	DT8	NVA	Bit map received (xAE).
2	CHANGE_IN	DT8	NVA	Change received (xAC).
3	ACK_IN	DT8	NVA	Acknowledge received (xAA).
4	POLL_IN	DT8	NVA	Poll received (xAB).
5	MAP_REQ_IN or CAA ACK	DT8	NVA	**Slave Mode:** Bit map request received (xAD). **Peer/Sync Mode:** CAA Change Acknowledge. This flag is set by DT8 to inform the application that the posted data change is accepted.

Table 2–12. Special Message Flags *(Cont.)*

Bit	Flag Name	Set By	Cleared By	Description
6	BIT_MAP_OUT	DT8	NVA	Bit map sent (xAE).
7	CHANGE_OUT	DT8	NVA	Change sent (xAC).
8	ACK_OUT	DT8	NVA	Acknowledge sent (xAA).
9	POLL_OUT	DT8	NVA	Poll sent (xAB).
10	MAP_REQ_OUT	DT8	NVA	Bit map request sent (xAD).
11	STATION_ALIVE	DT8	DT8	**Station Alive:** This flag is used only in Master Mode to inform if slave stations are responding. If a slave station responds, DT8 clears the Station Alive flag to 0 (zero). If the slave station does not respond to three consecutive messages, the Station Alive flag is set to 1 (one) to indicate that the slave station has not sent a valid response to the master.
12	POLL_ENABLE	NVA	none	**Poll Enable:** This flag is used only in Master Mode to enable or disable the polling of slave stations. If a slave station's Poll Enable flag is set, the master communicates with the slave station. If the flag is clear, the slave station is dropped from the master's polling sequence. If a Special Message buffer is defined and the Poll Enable flag is not actively used by the application, it must be assigned a variable name and that variable must be set to PERMONE in a non-vital equation in order for the master to poll the slave station.
13	TEXT_IN	DT8	NVA	Text received (xAF).
14	TEXT_OUT	DT8	NVA	Text sent (xAF).
15	SEND_MAP	NVA	DT8	Send a bit map (xAE).

Table 2–12. Special Message Flags *(Cont.)*

Bit	Flag Name	Set By	Cleared By	Description
With Slave, if the application program requests a SEND_MAP, then the protocol clears the output buffer and sends the latest data.				
16	*unused*	*none*	*none*	Should be set to PERMZERO.
Note: Special Message Flags 17 through 20 apply only when using DT8 version F32 or later on CSEX, CSEX2, CSEX3 or when using DT8 version A1 or later on CSEX4 and NVSP.				
17	SEND_TIME_REQ	NVA	DT8	Send a request for date/time update (xA7/x01).
18	TIME_REQ_IN	DT8	NVA	Request for date/time update received (xA7/x01).
19	SEND_TIME	NVA	DT8	Send date/time update (xA7/x00).
20	TIME_IN	DT8	NVA	Date/time update message received (xA7/x00).
Note: Special Message Flags 21 through 24 apply only when using DT8 version C10 or later on CSEX4 and NVSP.				
21	CLR_LATCHED_ CTL	NVA	DT8	Clear the latched Control data (Latched mode only, CSEX4/NVSP only)

Table 2–12. Special Message Flags *(Cont.)*

Bit	Flag Name	Set By	Cleared By	Description
22	AUTOCLR_LATCH_CTL	NVA	none	This bit enables the enhanced control data management for Peer Sync mode. When this bit is set by the Application, incoming control data and queued indication data is cleared whenever the link loss is detected due to a timeout and whenever link is re-initialized by either side. When the bit is not set, incoming latched control data is not cleared. **Note:** It is highly recommended that bit 22 is set for all new applications to maximize the coherence of data on both sides of the link. This enhanced functionality was made optional in order to avoid conflict with deployed applications configured with earlier versions of the protocol lacking this feature.
23-24	*unused*	*none*	*none*	Should be set to PERMZERO.

2.5.2.8 Text Messages

DT8 is capable of handling text character messages, and a method is provided for defining Text Messages in the non-vital application. Text Messages defined for different ports can be linked so that text data received on one port is automatically sent out the other port.

To define a Text Message, preface the CONTROL, INDICATION or MESSAGE record with the word “TEXT”, as shown below:

```
TEXT CONTROL = ADDRESS (address), LENGTH (length), NAME (name)

TEXT INDICATION = ADDRESS (address), LENGTH (length), NAME (name)

TEXT MESSAGE = ADDRESS (address), LENGTH (length), NAME (name)
```

See Table 2–13 for descriptions of the Special Message Contents parameters.

Table 2–13. Text Message Parameters

Parameter	Description
address	The station address, expressed in binary from 1 to 8 bits, as for other serial messages.
length	The total number of characters in the Text Message buffer.
name	An optional message identifier of up to 16 characters used when linking Text Messages.

Unlike control and indication messages, message data records do not follow a Text Message definition.

Examples:

```
TEXT CONTROL = ADDRESS (00000001), LENGTH (100), NAME (TRK-IN)

TEXT INDICATION = ADDRESS (00000001), LENGTH (100), NAME (TRK-OUT)
```

2.5.2.9 Text Message Linking

An incoming Text Message can be linked to an outgoing one to provide automatic transfer of text data between ports. Messages can be linked on a one-to-one basis only.

Linking is accomplished by using the same message name when specifying text messages on different ports. In this case, only one of the messages needs a length designation and a separate address designation is required only if the incoming and outgoing message addresses are actually different.

For example:

```
SERIAL PORT 2 = TYPE (DT8)

TEXT CONTROL = ADDRESS (00001), LENGTH (100), NAME (HHT-OUT)

TEXT INDICATION = ADDRESS (00010), LENGTH (100), NAME (HHT-IN)


SERIAL PORT 4 = TYPE (DT8)

TEXT CONTROL = ADDRESS (11110), NAME (HHT-IN)

TEXT INDICATION = NAME (HHT-OUT)
```

This example allows a control message received at Port 2 to be transmitted from Port 4 with the same address, and the control message received at Port 4 to be transmitted from Port 2 with a different address.

2.6 REQUIRED DT8 SUPPORT FILES

The following files are required in order to compile DT8 with a non-vital application. All of these files are contained in the \CTCFILES folder of the CAAPE.

- PROTOCOL.DAT and PROTNAME.DAT: Used by the CAAPE when compiling an application to obtain references to DT8 files and other data pertinent to DT8.
- DT8.LNK: Contains the DT8 protocol emulation software linked with the non-vital application.

> **Note:** This file is not a shortcut as is identified by Windows, but contains executable code.

- DT8_P.LNK: Contains the DT8 protocol emulation software linked with the non-vital application.

> **Note:** This file is not a shortcut as is identified by Windows, but contains executable code.

- DT8.LPC: The default DT8 configuration file, used unless another configuration file is specified in the application via a CONFIGURATION FILE record.

2.7 CONFIGURATION EDITOR

Protocol-specific settings for a non-vital serial communication port are contained in an LPC (Local Port Configuration) file. The name of a user-created protocol configuration file is specified in a CONFIGURATION FILE record in the non-vital application (described earlier in this section). Application-specific protocol settings are defined using the DT8 configuration editor which produces a protocol configuration file (an LPC file) containing the desired settings. Otherwise, the CAAPE assigns default DT8 settings to the serial port. The extension for the configuration file is LPC to avoid conflicts with other file extensions used by the CAAPE.

If the application does not specify a protocol configuration file for the serial port, the default file (DT8.LPC) is used with the following DT8 settings:

Table 2–14. DT8 Settings

Parameter	Setting
Baud Rate	1200
Data Format	8 data bits, 1 stop bit, odd parity
Quiet Poll Interval	100 milliseconds (ms)
Expedite Mode	Off (0)
Message Timeout Interval	100 ms
RTS-to-CTS Timer	0 ms
Hold RTS Timer	10 ms
Multi-drop Flag	Off (0)
CTS Flag	Off (0)
Invert RTS	No (0)
Invert CTS	No (0)
AD/AE Poll Flag	Off (0)
AE Timeout	0 ms
Sync Message Limit	3 retries
Output Buffer	Enabled (1)

All of these settings, except for the baud rate and data format, are configurable using the configuration editor. The default baud rate (1200) and data format (8/1/odd) are used unless overridden by DEFAULT BAUD RATE, BAUD RATE CONTROL and/or DATA FORMAT records in the application’s Serial Communications Section. These records are described earlier in this section.

2.7.1 DT8 Configuration Settings

The DT8 configuration editor allows the user to specify various protocol settings. Use of DT8 interval timers and flags, as well as the protocol's operating modes are described in detail in SECTION 3 –Theory of Operation. Consider system response requirements and the data communication rate when selecting time values. For example, a timeout value less than the time necessary to transmit a complete message causes a problem.

Table 2–15. DT8 Configuration Settings

Protocol Setting	Default	Range	Description
Quiet Poll Interval	250 ms	10 – 5000 ms	Sets the time (in milliseconds) between polls if the port is a master station and there are no messages pending for any slave station.
Expedite Mode	Off (0)	0 = Off 1 = On	Controls the protocol's use of the A8/A9 terminator byte (see Section 3.2.3.1) and applies to Master/Slave Mode only.
Message Timeout Interval	100 ms	10 – 5000 ms	Sets the time (in milliseconds) the system waits for a message before faulting the remote unit.
RTS-to-CTS Timer	0 ms	0 – 5000 ms	Defines the maximum time (in milliseconds) after asserting RTS (Request to Send) to wait for CTS (Clear to Send). If CTS does not become true after this time delay, CTS is ignored and transmission takes place regardless. If this timer is set to 0 (zero), however, DT8 waits three seconds for CTS before aborting message transmission.
Hold RTS Timer	10 ms	0 – 5000 ms	Defines the time (in milliseconds) to hold RTS true after the last byte of a message has been transmitted. This guards against the transmit medium (such as modem) from shutting down before message completion.
Multi-Drop Flag	False (0)	0 = False 1 = True	If this flag is true, RTS is asserted (true) before message transmission and is de-asserted (false) upon message completion. If this flag is false, RTS is asserted all the time.

Table 2–15. DT8 Configuration Settings *(Cont.)*

Protocol Setting	Default	Range	Description
CTS Flag	False (0)	0 = False 1 = True	If this flag is true, DT8 waits for CTS to become true before message transmission begins; used in concert with the RTS-to-CTS Timer described above. If this flag is false, DT8 ignores the state of CTS.
RTS Invert Flag	False (0)	0 = False 1 = True	If this flag is false, the asserted (true) level of the RTS control line is HIGH, otherwise the asserted level is LOW. The RTS Invert Flag must be set to true (1) for Ports 1 and 2 in RS-422 mode when using CSEX[1] or CSEX2, and set to false (0) for all other ports in RS-422 mode on these hardware platforms. This stipulation does not apply to CSEX3.
CTS Invert Flag	False (0)	0 = False 1 = True	If this flag is false, the asserted (true) level of the CTS control line is HIGH, otherwise the asserted level is LOW.
AD/AE Poll Flag	False (0)	0 = False 1 = True	If this flag is false, a port defined as a master polls unresponsive slave stations with AD messages (Bitmap Request), otherwise it polls with AE messages (Bitmap). DT8 message types are explained in SECTION 3 –Theory of Operation.
AE Timeout	0 secs	0 – 65000	In Master Mode, the AE Timeout defines the number of seconds between each bit map. If this timeout is not equal to zero, a master station periodically transmits a Bitmap Request or a Bitmap (depending upon the AD/AE Poll Flag) to slave stations.

Table 2–15. DT8 Configuration Settings *(Cont.)*

Protocol Setting	Default	Range	Description
Sync Message Limit	3	1 – 1000	In Sync Mode, DT8 attempts to deliver a message or establish communication this many times before polling the remote peer every 10 seconds.
Output Buffer	Enabled (1)	0 = Disabled 1 = Enabled	This flag enables or disables use of the eight-deep indication queue by DT8. This queue is described in Section 3.2.3.1. If this flag is false, no buffering of output messages occurs. If the flag is true, up to eight indication messages are stored if communication fails or changes occur rapidly. If the application requests a SEND_MAP then protocol clears the output buffer and sends the latest message.

Table 2–16. DT8 Configuration Settings As Used By Type

Configuration Setting	DT8 Slave	DT8 Master	DT8 Peer	DT8 Sync
Quiet Poll Interval	N	Y	N	N
Expedite Mode	Y	Y	N	N
Message Timeout Interval	Y	Y	Y	Y
RTS-to-CTS Timer	Y	Y	Y	Y
Hold RTS Timer	Y	Y	Y	Y
Multi-Drop Flag	Y	N	N	N
CTS Flag	Y	Y	Y	Y
RTS Invert Flag	Y	Y	Y	Y
CTS Invert Flag	Y	Y	Y	Y
AD/AE Poll Flag	Y	Y	Y	Y
AE Timeout	N	Y	N	N
Sync Message Limit	Y	Y	N	Y
Output Buffer	Y	Y	Y	Y

2.7.2 DT8 Configuration for Non-vital Serial over Ethernet Connections

When configuring DT8 on the NVSoE connections, a number of parameters have no effect. Specifically, the CTS/RTS Invert Flag parameters, Hold RTS and RTS-to-CTS timer parameters, CTS flag and Multi-drop flag are ignored for virtual serial over TCP connections since these parameters are not applicable in that case.

Additionally, care must be taken when setting polling and timeout intervals on the NVSoE connections to accommodate the expected worst-case network propagation delays and transmission times. Specifically, the message timeout must not be set any shorter than twice the sum of the expected worst-case transmission time and propagation delay on the network path between the two DT8 nodes. As a general guideline to avoid network congestion, the polling interval should not be set any shorter than twice the message timeout interval.

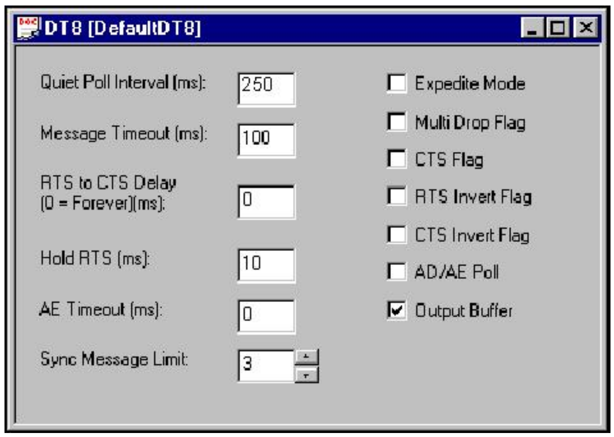
- Alstom offers the user a Microsoft Windows®-based CAAPE graphical LPC editor as a method for selecting DT8 port settings and generating LPC files

2.7.3 Windows-Based CAAPE Graphical LPC Editor

The CAAPE provides a built-in graphical editor for creating DT8 LPC files.

Table 2–17. Create an LPC File With CAAPE

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Open a non-vital application project, and configure the LPC file.</td></tr>
<tr><td>2</td><td>Open the LPC component, and a form is displayed for entry of the protocol’s option settings.


</td></tr>
<tr><td>3</td><td>Enter the desired settings.</td></tr>
<tr><td>4</td><td>Save the component.</td></tr>
<tr><td>5</td><td>Convert the LPC graphical component’s data to a LPC text file and refer to that file in a CONFIGURATION FILE record in the non-vital application (described earlier in this section):

If the application is created entirely using CAAPE graphical editing, link the LPC component to the appropriate serial port in the system’s hardware component. The application’s Make Files process converts the component to an LPC text file and creates the file reference.
If the application is created using manually-edited text input files, enter a CONFIGURATION FILE record as part of the serial port definition in the CSS file.
</td></tr>
</table>

See the CAAPE’s on-line help for further details on creating and using LPC components.

THIS PAGE INTENTIONALLY LEFT BLANK.

SECTION 3 – THEORY OF OPERATION

3.1 GENERAL

This section describes of DT8 modes of operation.

3.2 APPLICATIONS

3.2.1 Protocol Description

The DT8 communication protocol functions on any NVS or NVSoE port of the wayside non-vital processor component. See Table 3–1 for the modes of operation available in DT8.

Table 3–1. DT8 Modes of Operation

Mode	Description
Peer	A pair of units connected at opposite ends of the code-line, where either can send data when necessary. Neither unit is superior, as opposed to a Master/Slave configuration.
Sync	Same as Peer Mode, except in the event of communication failure. Sync Mode supports multiple field stations in a single unit.
Master	Any unit that is the master on a Master/Slave line. The master unit can speak at any time and must poll slaves for information.
Slave	Each unit that is not a master on a Master/Slave line. Slave units speak only when polled by the master unit, and only if properly addressed.

Throughout this manual, the notation of a lower case ‘x’ followed by two numbers, letters, or a combination of both represents a hexadecimal value. Use Table 3–2 to convert between decimal (base 10), hexadecimal (base 16), and binary (base 2).

Table 3–2. Decimal, Hexadecimal and Binary Values

Dec	Hex	Bin	Dec	Hex	Bin	Dec	Hex	Bin	Dec	Hex	Bin
0	0	0000	4	4	0100	8	8	1000	12	C	1100
1	1	0001	5	5	0101	9	9	1001	13	D	1101
2	2	0010	6	6	0110	10	A	1010	14	E	1110
3	3	0011	7	7	0111	11	B	1011	15	F	1111

For example, the DT8 Acknowledge command byte is xAA (or 1010 1010 in binary) and the bitmap command byte is xAE (or 1010 1110 in binary).

3.2.2 Modes of Operation

The application engineer specifies the DT8 operating mode when he or she designs the non-vital application. In all operating modes, DT8 can queue up to eight non-concurrent indication parameter changes if communication fails or controls are either latched or unlatched.

If communication fails, changes occur quicker than the reporting capabilities of DT8 can handle. An example is when several application parameters change state within a time period shorter than the polling interval, especially in Master/Slave Mode. DT8's eight-deep indication is known as First-In, First-Out (FIFO) queue or a "hold buffer". It prevents loss of information that might otherwise happen when frequent changes occur relatively quickly. If the queue becomes full and a ninth change occurs, the entire queue is cleared before the new message is buffered. This indication queue can be disabled.

When a control is unlatched, DT8 posts a received control to the control buffer for use by the non-vital application logic. Then, the entire control buffer is automatically cleared to zeros. When a control is latched, DT8 retains the state of control bits. Each time a change is received, the buffer is updated to reflect the new state of any changed control bits. If a control bit is set high or to 1, it remains in the latched or unlatched state until a message with that bit reset to 0 is received. Each DT8 mode of operation has a default setting for latched vs. unlatched controls, although the application engineer can override the default setting.

3.2.2.1 Peer Mode

If only two units occupy a line, the Peer Mode of operation ensures more efficient delivery of information. Both peer units function in full-duplex operation and each has the same unlimited opportunity to transmit whenever necessary. Normally, peers sit quietly on the line, only transmitting when the non-vital application logic posts a change to the indication buffer. When a change is posted, one peer unit transmits either a bitmap or Change message to the other remote peer unit. When peer receives a bitmap or Change message, it sends an Acknowledge message. Whenever a peer receives a bitmap message, it acknowledges it and responds with a bitmap message of its own.

A peer unit may send its bitmap or Change message up to three times. It waits for a preset and user-configurable Message Timeout Interval between transmissions if no Acknowledge message is received. The Message Timeout Interval is 100 milliseconds by default. If there is no response, the local peer periodically sends a bitmap, with the same time interval between each transmission, until the message is acknowledged and the remote peer sends a bitmap message. For more information on how to configure the message time interval, see SECTION 2 –CAAPE Package.

Note: The Peer Mode is supported for compatibility with earlier installations and legacy products. However, for new applications it is superseded by the more efficient Sync mode.

3.2.2.2 Sync Mode

As with Peer Mode, Sync Mode can be used when two units occupy a line and both have the same unlimited opportunity to transmit whenever necessary. In Sync Mode, both units are referred to as “peers”. Sync Mode improves upon Peer Mode in the following ways:

- Sync Mode supports more than one station at a given field location. Each station is assigned a unique address and set of message buffers: control, indication, and special. See SECTION 2 –CAAPE Package for details on defining DT8 message buffers.

  Note: When multiple stations are used on a single Sync link, the Indication messages are sent out one at a time, with the subsequent message not sent out until an acknowledgment for the previous message arrives. Therefore, when changes occur in data for multiple stations simultaneously, the time to transmit all messages is increased by the amount of time it takes the opposite side of the link to validate all messages and issue acknowledgments. Only indication messages are affected by this restriction – incoming control messages for all stations continue being processed normally, including issuing the acknowledgment messages.

- The link activation and maintenance algorithm is improved:
  - When link is not active, a poll (xAB) message is issued every 10 seconds.
  - A link becomes active when a poll (xAB) is received or when an acknowledgment to a transmitted poll is received.
  - While link is active, each message is retransmitted up to a configured number of times (Sync Message Limit) at configured intervals (Message Timeout), and link is declared lost after the last retransmission times out.
  - The link activation occurs separately for each station; however, loss of link on any station due to retransmission timeout causes links for all stations to be dropped.
  - When a poll (xAB) message is received, or when a link becomes active, all queued change indications are discarded and a full current bitmap (xAE) is transmitted.
  - An optional feature of the control data management can be enabled via the Special Message Bit 22 (AUTOCLR_LATCH_CTL).
    - When a poll (xAB) message is received, or when a link is lost, all previously received control values are cleared.

Note: It is highly recommended that bit 22 is set for all new applications to maximize the coherence of data on both sides of the link. This enhanced functionality was made optional in order to avoid conflict with deployed applications configured with earlier versions of the protocol lacking this feature.

3.2.2.3 Master/Slave (Multi-Drop) Mode

The Master/Slave Mode of operation is used when one station, the master, controls one or more slave stations on a line, which each have a unique destination address. An example of this is when an office station sends messages to several field stations. The master station sends each slave station, in turn, a Poll, Change, bitmap Request, or bitmap message. Slave stations receiving information from a master must be operating in Slave Mode. Each slave decodes the address portion of the message and only the station that matches the destination address acts on the message. The one exception is the “All Stations” broadcast message. It contains an address of 0 (zero) to which all slaves respond but do not acknowledge. The “All Stations” broadcast message applies only to a master when sending an xA7 message for clock synchronization.

The master polls slaves only if the application logic sets the Poll Enable Flag in the Special Message buffer. For more information, see the Poll Enable Flag section.

A master unit repeatedly sends a message until the expected response is received, waiting for a preset and user-configurable Message Timeout Interval for the response. If a message is sent three times without receiving a response, the master reverts to polling the failed slave once every ten polling cycles.

Master/Slave Mode typically uses full-duplex operation where both the master and slave units have their own separate transmit and receive lines. With the proper use of the RTS and CTS signals, however, it is adaptable to half-duplex operation, which requires only single transmit and receive lines.

Note: The NVSoE connections are inherently point-to-point in nature. However, these connections support the master-slave mode. Additionally, a single slave can be configured as several logical slave stations.

This functionality is used when a slave DT8 NVSoE port relays information from a master DT8 serial port controlling a number of slave DT8 stations. In this case, each virtual station on the slave DT8 NVSoE port can represent a physical slave station attached to the master DT8 serial port. As a result, a TCP/IP to RS-485 DT8 gateway is implemented easily.

A slave's response depends upon the type of message received:

- **Poll:** If the slave station has a pending Change or bitmap, it sends it at this time. If no messages are pending, an Acknowledge is sent. If the slave's reply is either a Change or a bitmap, no separate Acknowledge is sent by the slave because the Acknowledge is implied. When a slave replies to the master with a Change or bitmap, the master responds with an Acknowledge.
- **Change or Bitmap:** The slave station posts the new information to the non-vital application logic and then replies the same as for a Poll message.
- **Bitmap Request:** The slave station replies with a bitmap. The protocol clears the output buffer and sends the latest bitmap.

3.2.3 DT8 Messages

3.2.3.1 Message Structure

All DT8 messages have a common format:

- Each message starts with a command byte, xA7 or xAA through xAF.
- The command byte is followed by the address of the destination station, which ranges from 1 to 4 bytes.
- A variable length body of data, whose length can equal zero bytes depending upon the message type, follows the station address. The body of a message, if present, ends with a CRC-16 checksum calculated using the polynomial $x^{16} + x^{15} + x^2 + 1$.
- Each message ends with a terminator byte. The terminator byte is normally xA9, but can be xA8 when Expedite Mode is turned on. Expedite Mode applies to Master/ Slave Mode only. In this case, when Expedite Mode is turned on, each slave's Change message or bitmap ends with an xA9 if no other changes are pending. It ends with an xA8 when additional changes are queued and ready to be sent. If the master responds with an Acknowledge ending with an xA9, the message exchange is terminated. If the master's Acknowledge ends with an xA8, the slave is granted permission to send the next Change message immediately.

3.2.3.2 Station Addressing

DT8 allows a station's address to range from x01 to xFE. An address of x00 (zero) is a special "All Stations" broadcast address used in xA7 messages only. For more information, see Section 3.2.3.5 Description of the xA7 Message Type.

Station Addresses are specified separately for each message assigned to a station. Slave and Peer nodes support a single station, which is a collection of different message types such as Control, Indication, Text Control, Text Indication and Special. Master and Peer Sync nodes support multiple stations. Each station has a matching node on a remote side of a communication link: a peer for Peer and Peer Sync nodes, a Slave for Master nodes, and the Master for Slave nodes.

The address assigned to each station message must match the address assigned to a corresponding station message on the remote side of the communication link. Specifically:

- The address assigned to a Control message must match the address assigned to the remote Indication message.
- The address assigned to a Text Control message must match the address assigned to the remote Text Indication message.
- A simple arrangement of assigning the same address value to all messages on both sides of a communication link is supported. The only arrangements where different addresses are required are a multi-slave Master node or a multi-station Peer Sync node.

However, it is recommended to assign a unique number to each Control-Indication or Text Control-Text Indication message pair. This facilitates diagnosing communications problems when multiple station data exchange streams flow through the same communication link.

3.2.3.3 Split Bytes

If a byte of data in a message's body is the same as a command byte (ie: has a value of xA*n*, where *n* is any hexadecimal digit), the data is sent as two bytes: xA0 followed by x0*n*. This includes bytes in the CRC-16 checksum that can be part of the body. As a result, data byte xAB is sent as two split bytes xA0 and x0B. On reception, the split bytes are recombined into a single byte. In the diagnostic screens for DT8, the actual data byte, and not the split bytes, is shown.

3.2.3.4 Message Types

DT8 supports the following messages: Acknowledge, Poll, Change, Bitmap Request, Bitmap, A7 with Subcommand, and Text. The nomenclature in Table 3–3 describes the message content.

Table 3–3. DT8 Messages

Message	Description
St	The station address, 1 to 4 bytes in length.
Bn	The position of the byte in the control or indication buffer, the 0-based relative index, which contains the parameter that changed state. The state of each control or indication parameter can be true, which is a binary 1, or false, which is a binary 0. Eight parameters are sent as a unit or a byte. The true/false states of the first eight parameters in a control or indication buffer are contained in byte 0. The states of parameters 9 – 16 are in byte 1, etc. Each byte position, known as Bn, is paired with a byte value or Bi.
Bi	The new value of the changed control or indication byte. Bi is the state of eight control or indication parameters. The state of the first parameter in a control or indication buffer is in the least significant bit of byte 0. The state of the eighth parameter is in the most significant bit of byte 0.
{}	Indicates that the message contains a variable number of byte pairs, which is dependent on the total number of parameter changes.
bi[x]	The value of byte x in the control or indication buffer, comprising the state of eight control or indication parameters.
cl, ch	The low and high bytes, respectively, of the CRC-16 checksum.
xA9/xA8	The message's terminator byte. It can be xA9 or xA8,which is only possible when Expedite Mode is turned on.

Table 3–4 describes these messages in detail.

Table 3–4. DT8 Message Types

Cmd Byte	Message Type	Message Description	Message Format
xA7	Subcommand	Transports Subcommands (see Section 3.2.3.5). For example: `A7 02 01` c_1 c_h `A9`	xA7 st Subcommand c_1 c_h xA9
xAA	Acknowledge	Informs the remote unit that a message was received without error and was accepted. For example: `AA 01 A9`	xAA st xA9/xA8
xAB	Poll	Request for information (any parameter changes) from the remote unit. For example: `AB 03 A9`	xAB st xA9
xAC	Change	Sends changed indication or control parameters to the remote unit. Eight adjacent parameters are grouped and sent as a full byte. The message body contains data pairs (byte number b_n and byte value b_i) for only those bytes that have changed state since the last message was sent. For example: `AC 01 04 0C 1B 82` c_l c_h `A9` If more than 50% of all indication or control parameters change state, DT8 sends a bitmap instead of a Change message.	xAC st {b_n b_i…b_n b_i} c_1 c_h xA9/xA8
xAD	Bitmap Request	Requests that a bitmap be transmitted by a remote slave unit. For example: `AD 03 A9`	xAD st xA9

Table 3–4. DT8 Message Types *(Cont.)*

Cmd Byte	Message Type	Message Description	Message Format
xAE	Bitmap	Informs the remote unit of all indication or control parameters for a station. This message contains data for the entire indication or control buffer. For example: `AE 05 01 5A 08 F3 91` c_l c_h `A9`	xAE st b_i [0] b_i [1]…b_i [n] c_l c_h xA9/xA8
xAF	Text	Transfers text buffers between DT8 locations. This message type is used primarily by optional DataLogger software module for downloading event logs to a PC. Refer to Alstom publication P2512F. For example: `AF 01 57 41 52 4E 49 4E 47` c_l c_h `A9`	xAF st b_i [0] b_i [1]…b_i [n] c_l c_h xA9/xA8

3.2.3.5 Description of the xA7 Message Type

The xA7 message type is used by the DT8 protocol to request and send time synchronization messages. This message type was added to DT8 in CSEX, CSEX2, CSEX3 software version F32. Its general format is:

```
xA7 st Subcommand xA9
```

where:

- **xA7** is the command byte.
- **st** is the station address (1 to 4 bytes in length). For the xA7 message type, an address of 0 (zero) can be used by a master in Master/Slave mode for an “All Stations” broadcast message. The “All Stations” broadcast message is accepted, but not acknowledged, by all slave stations.
- **Subcommand** is the actual xA7 command, a single byte from x00 (zero) to xFF. It can be followed by a body of data including a CRC-16 checksum.
- **xA9** is the message terminator byte.

The following valid xA7 Subcommands are supported by DT8:

- Subcommand x00
- Subcommand x01

3.2.3.5.1 Subcommand x00 – Date/Time Update

The following items comprise the rest of this Subcommand:

second, minute, hour, day, month, (year % 100), (year / 100), time zone, c_l, c_h

See Table 3–5 for subcommand descriptions.

Table 3–5. Subcommand x00 Descriptions

Subcommand	Description
second	The seconds of the current time, in hexadecimal from x00 to x3B (00 to 59).
minute	The minutes of the current time, in hexadecimal from x00 to x3B (00 to 59).
hour	The hours of the current time, in hexadecimal from x00 to xA7 (00 to 23 Military Time).
day	The current day of the month, in hexadecimal from x01 to x1F (01 to 31).
month	The current month of the year, in hexadecimal from x01 to X0C (01 to 12).
(year % 100)	The current year (in hexadecimal), operated on by a modulus of 100. This represents the last two digits of the year. For example, if the year is 2001, then (2001% 100) results in x01 (% is the modulus operator).
(year / 100)	The integer division of the year by 100, resulting in the century portion (the first two digits) of the year (in hexadecimal). Thus, (2001/100) results in x14 (20).
time zone	The time zone, in hexadecimal from x00 to x17 (00 to 23).
c_l and c_h	The low and high bytes, respectively, of the CRC-16 checksum.

DT8 can truncate the sequence of data in the minutes through the time zone subcommands anywhere along the sequence. In this case, the terminator byte (xA9) signals the end of the message. Therefore, the shortest possible message body, excluding the Subcommand byte, is:

second, c_l, c_h

DT8 uses the date and time received in this message to update the component's real-time clock (RTC). The RTC does not support the time zone, so the non-vital application must handle the time zone.

To use the time zone in an application, the DT8 serial port must contain an additional station and its 8-bit address must be set to 0 (zero). This station's control and indication buffers must both be exactly eight bits long. For more information on defining DT8 message buffers, see SECTION 2 – CAAPE Package. DT8 automatically initializes these buffers to xFF. Upon receipt of the time zone in a Date/Time Update message, DT8 stores the 8-bit (one byte) time zone value to this station's control buffer for use by the application logic.

For the time zone to be sent in a transmitted Date/Time Update command, the application logic must store the desired time zone value in this station's indication buffer before the message is sent. When DT8 transmits a Data/Time Update command, it compares this station's indication buffer to xFF. If it is equal to xFF, no time zone value is included in the message. Otherwise, whatever value is found in the buffer is sent as the time zone.

Note: The master ignores the Date/Time Update command when it is received in Master/Slave Mode. A master can update a slave's clock, but a slave cannot update a master's clock. Date/Time Update commands can be sent between peer units.

The following is an example Date/Time Update command sent on 7/13/2001 at 11:29:57 from time zone 23:

A7 00 00 39 1D 0B 0D 07 01 14 17 c_l c_h A9

This example is an "All Stations" broadcast message (the station address is 0).

3.2.3.5.2 Subcommand x01 – Request Date/Time Update

When the non-vital application logic requires a Date/Time Update, only a slave station in Master/Slave Mode sends this message. To send this message, the application logic must set a Special Message Flag. After the flag is set and the next poll message is received from the master, DT8 sends a Request Date/Time Update message. In this case, a pending indication change takes precedence over sending the Request Date/Time Update. This message contains no data components, therefore the complete message is xA7 st 01 xA9 (for example: A7 02 01 c_1 c_h A9).

Table 3–6 summarizes the Special Message Flags used in conjunction with the xA7 Subcommands. The flag names in the table are recommendations only. The Application Engineer is free to use any appropriate parameter names. The "Set By" and "Cleared By" columns indicate whether the non-vital application (NVA) logic or the DT8 protocol is responsible for setting to 1 or clearing to 0 each Special Message Flag. Setting a Special Message Flag by the NVA instructs the DT8 protocol to perform a one-time action, such as transmitting a Request Date/Time Update message. Conversely, DT8 sets a Special Message Flag to inform the NVA of an event, such as the receipt of a Date/Time Update message.

Table 3–6. Special Message Flags Used With xA7 Commands

Bit	Flag Name	Set By	Cleared By	Description
17	SEND_TIME_REQ	NVA	DT8	Send Request Date/Time Update message (Subcommand x01).
18	TIME_REQ_IN	DT8	NVA	Request Date/Time Update message received (Subcommand x01).
19	SEND_TIME	NVA	DT8	Send Date/Time Update message (Subcommand x00).
20	TIME_IN	DT8	NVA	Date/Time Update message received (Subcommand x00).

3.2.4 Modes Of Exchange

3.2.4.1 Typical Peer Mode and Sync Mode Exchange

The content that follows and Table 3–7 exemplify the interaction between two peer units when both units are exchanging messages and are operational. This message interaction is typical of both Peer and Sync Modes in DT8.

Table 3–7 shows a typical message exchange, byte by byte, between two peer units. The table is read left to right and then top to bottom.

The office station starts operating and sends a bitmap message (xAE). The field station, which is currently operating, acknowledges (xAA) and then responds with a bitmap message (xAE) to update the office. The office station acknowledges (xAA) the bitmap message, and then both ends of the code-line wait for another message (for a change to occur). When the field station detects a change in its non-vital inputs, for example, it sends a Change message to the office station. In the example, byte 1,the second byte of the station's indication buffer, changed to x55 (from x11) and byte 4 changed to xAA, from x44. This byte is sent as two split bytes: xA0 and x0A. The office station acknowledges (xAA) and both ends wait for another message. Additional messages are transmitted as subsequent changes occur. If more than 50% of all indication parameters change state, DT8 sends a full bitmap instead of a Change message.

Table 3–7. Message Exchange Between Peer Units

State	Office Station	Field Station
Power Up	AE 02 11 22 33 44 55 66 A0 03 59 A9	AA 02 A9
		AE 02 00 11 22 33 44 55 5D 7F A9
	AA 02 A9	
Running		AC 02 01 55 04 A0 0A F2 FF A9
	AA 02 A9	
	AC 02 02 34 80 27 A9	AA 02 A9
	AE 02 11 34 33 44 55 EF 2B 3C A9	AA 02 A9
		AE 02 55 11 44 A0 0A 22 55 C4 58 A9
	AA 02 A9	

Some non-vital applications may cause a periodic change to an indication parameter, forcing the transmission of a Change message, which informs the remote station that the communication link is active. In this case, a steady pattern of messages is observed. Otherwise, messages are sent only as the need arises.

3.2.4.2 Typical Master/Slave Mode Exchange

Table 3–8 shows an exchange between master and slave units, with only one slave responding. The table is read left to right and then top to bottom.

The master station polls slave stations 2, 3, 4 and 5, none of which have yet updated the master with their indication data. After sending a complete sequence of bitmap requests (xAD) to all slave stations, which do not answer, the master restarts the polling cycle. The second time, slave station 4 answers with the requested bitmap (xAE). The master acknowledges (xAA) this message and continues polling. When station 4 is polled again, the master updates this station's control buffer with a bitmap (xAE). Station 4 responds to this message by sending a Change message (xAC) since by this time a change is pending. The Change message indicates that byte 1 has changed to x55,from x11, and byte 4 has changed to xAA, from x44. This byte is sent as two split bytes, xA0 and x0A. The master acknowledges xAA this message and continues polling. Since station 4 has updated the master with its indication data, the next time station 4 is polled, and the master sends a Poll message (xAB), which is a request for information, instead of a bitmap Request (xAD).

Table 3–8. Master/Slave Polling and Message Exchange

Master	Slave 2	Slave 3	Slave 4	Slave 5
AD 02 A9				
AD 03 A9				
AD 04 A9				
AD 05 A9				
AD 02 A9				
AD 03 A9				
AD 04 A9			AE 04 00 11 22 33 44 55 3B 7F A9	
AA 04 A9				
AD 05 A9				
AD 02 A9				
AD 03 A9				
AE 04 11 22 33 44 55 66 C5 59 A9			AC 04 01 55 04 A0 0A 7A FF A9	
AA 04 A9				
AD 05 A9				

Table 3–8. Master/Slave Polling and Message Exchange *(Cont.)*

Master	Slave 2	Slave 3	Slave 4	Slave 5
AD 02 A9				
AD 03 A9				
AB 04 A9			AA 04 A9	
AD 04 A9				

Table 3–9 shows a typical exchange between a master and responsive slave units. In this example, all slave stations are sent Polls (xAB) (request for information) by the master, instead of bitmap requests (xAD). This implies that all stations had previously updated the master with their indication data. The master station polls slave stations 2, 3, 4 and 5, all of which respond with an Acknowledge (xAA) implying that no changes are pending. After the complete sequence, the master restarts the polling cycle. The second time, slave station 4 responds with a bitmap (xAE) implying that multiple indications (more than 50%) changed state since it was last polled. The master acknowledges (xAA) this message and continues polling. When station 4 is polled again, the master updates this station's control buffer with a bitmap (xAE). Station 4 responds to this message by sending a Change message (xAC) since by this time another change is pending. The Change message indicates that byte 1 has changed to x55 (from x11) and byte 4 has changed to xAA (from x44). This byte is sent as two split bytes, xA0 and x0A. The master acknowledges (xAA) this message and continues polling.

Table 3–9. Other Master/Slave Polling and Message Exchange

Master	Slave 2	Slave 3	Slave 4	Slave 5
AB 02 A9	AA 02 A9			
AB 03 A9		AA 03 A9		
AB 04 A9			AA 04 A9	
AB 05 A9				AA 05 A9
AB 02 A9	AA 02 A9			
AB 03 A9		AA 03 A9		
AB 04 A9			AE 04 00 11 22 33 44 55 3B 7F A9	
AA 04 A9				
AB 05 A9				AA 05 A9
AB 02 A9	AA 02 A9			
AB 03 A9		AA 03 A9		
AE 04 11 22 33 44 55 66 C5 59 A9			AC 04 01 55 04 A0 0A 7A FF A9	
AA 04 A9				
AB 05 A9				AA 05 A9
AB 02 A9	AA 02 A9			
AB 03 A9		AA 03 A9		
AB 04 A9			AA 04 A9	
AB 05 A9				AA 05 A9

3.2.5 DT8 Message Timing

This section describes the protocol’s use of message interval timers to determine message timeouts and the polling interval. These timers are specified in the application’s protocol configuration file. Their usage differs between Master/Slave and Peer (Sync) Modes.

- The hardware control signals RTS and CTS that control the flow of transmitted messages.

3.2.5.1 RTS-CTS Timing

Figure 3–1 shows DT8 use of the RTS and CTS signals to control the flow of data. See Section 2.7 for descriptions of the CTS Flag, Multi-Drop Flag, RTS-to-CTS Timer, and Hold RTS Timer.

When DT8 is ready to transmit a message, the system asserts RTS and starts the RTS-to-CTS Timer if the Multi-Drop Flag is set. If CTS comes true before the timer expires or if the timer expires without CTS coming true, the system begins message transmission. This assumes that the RTS-to-CTS Timer is set to a non-zero value. If its value is zero, DT8 allows three seconds for CTS to come true before aborting message transmission. If the CTS Flag is zero, the state of CTS is ignored by DT8.

After the message is transmitted, DT8 starts the Hold RTS Timer. When this timer expires, FT8 de-asserts RTS, which presumably causes CTS to drop shortly afterward. If the Hold RTS Timer is zero, DT8 de-asserts RTS immediately after transmitting the last byte of the message.

Note: RTS and CTS signals are not present on the NVSoE ports. The TCP/IP communications layer handles the handshaking and synchronization transparently.

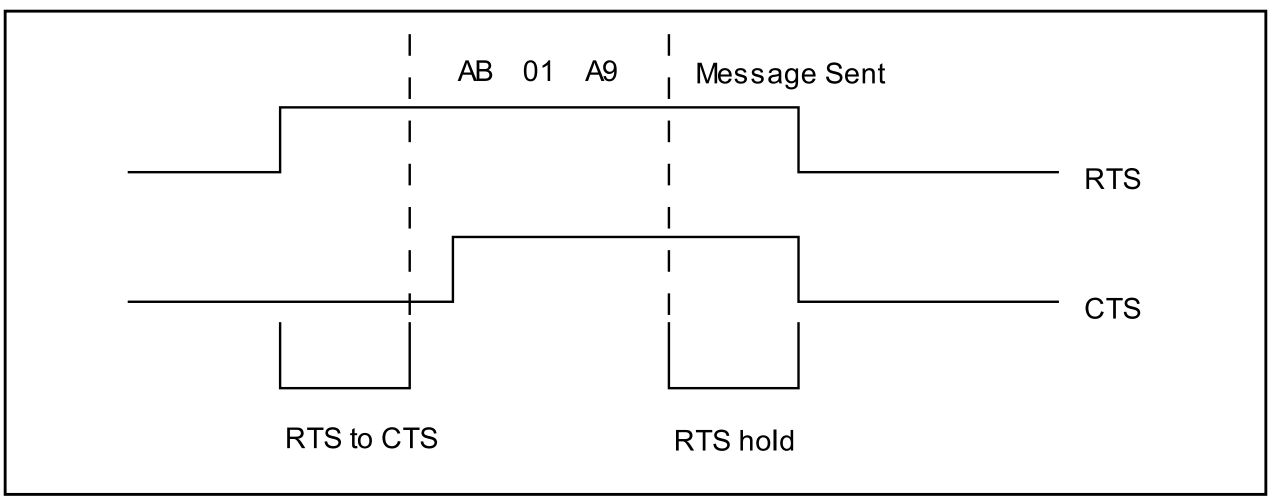


Figure 3–1. RTS-CTS Timing Diagram

3.2.5.2 Message Timing in Master/Slave Mode

DT8 maintains two user-configurable message timers in Master/Slave Mode: the Quiet Poll Interval and the Message Timeout Interval. Both intervals' timers start counting down upon transmission of the last byte of a poll. The Message Timeout Interval is the top priority timer. If the Message Timeout Interval is satisfied, meaning a valid response is received before expiration of this timer, the Quiet Poll Interval finishes timing out and then the next slave station in the sequence is polled. While waiting for the Quiet Poll Interval to time out so that the next slave station can be polled, DT8 checks for changes to controls for all slave stations. If a change is detected, DT8 immediately sends a Change message to the appropriate slave station before resuming the polling sequence.

This sequence is illustrated in Figure 3–2 through Figure 3–6. The two timers are shown by using the following line-types:

_________ = Message Timeout Interval ________ = Quiet Poll Interval

Figure 3–2 shows the poll that is not responded to, thus causing the Message Timeout Interval to expire and the next slave station in the sequence to be polled.

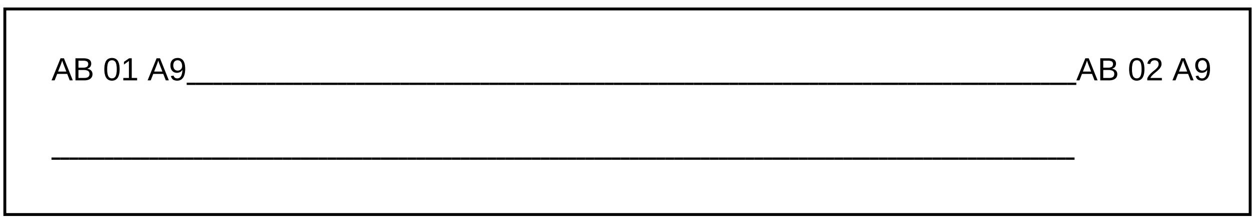


Figure 3–2. Message Timeout Interval and Quiet Poll Interval Diagram

Figure 3–3 shows a poll and the reception of an acknowledge before the expiration of the Message Timeout Interval, allowing the Quiet Poll Interval to take over. When the Quiet Poll Interval expires, the next slave station in the polling sequence (such as 2) is polled.

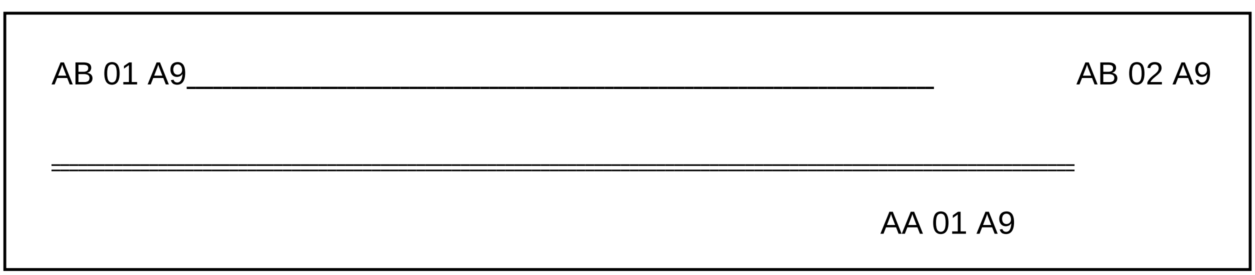


Figure 3–3. Quiet Poll Interval Takes Over

Figure 3–4 shows the same sequence as in Figure 3–3 except a change message is detected for station 3. The Change message is sent to station 3, it is acknowledged, and the next poll is sent.

```
AB 01 A9__________________          AC 03 02 DD xx xx A9      AB 02 A9

                       AA 01 A9                        AA 03 A9
```

Figure 3–4. Detection of Change Message

Figure 3–5 shows a poll, a Change message in response and the acknowledge to the slave station. Since the Quiet Poll Interval has not expired, the next poll is held off.

```
AB 01 A9______________________AB 02 A9                     AB 02 A9

                AC 01 01 CC xx xx A9
```

Figure 3–5. Delay of Poll of Other Stations

Figure 3–6 shows the same sequence as in Figure 3–5 except when the change is acknowledged by the master, no time remains in the Quiet Poll Interval. This results in an immediate polling of the next slave station.

```
AB 01 A9___________________________________________AA 01 A9 AB 02 A9
     ________________________________AC 01 01 CC xx xx A9
```

Figure 3–6. Condition for Scanning Other Stations

3.2.5.3 Message Timing in Peer and Sync Modes

In a peer-to-peer system (Peer Mode and Sync Mode), only the Message Timeout Interval is used by DT8. When a peer unit sends a message, the timer is started. If the remote unit does not send the required response before the timer expires, the message is sent repeatedly until the remote unit does respond properly. See Section 3.2.2 for more information on the use of the Message Timeout Interval in both Peer and Sync Modes.

3.2.6 Data Handling

Data flow in DT8 is controlled by four main tasks: I/O Handler, Receiver, Transmitter, and Control. See Table 3–10 for descriptions of each task's function.

Table 3–10. Data Flow Tasks

Task	Function
I/O Handler	This task has two functions: • The first function checks the application logic buffer to determine when a new indication is ready. If a new buffer is flagged as available by the application logic, the buffer is copied into the next available hold buffer (one of eight) and the application logic buffer is acknowledged and released. • The second function checks for any messages in the hold buffers waiting to be sent. If a message is waiting, the task checks the transmit buffer for an outgoing message in progress. If the transmit buffer is available, the task formats the hold buffer into the appropriate message type and flags that the message is ready to be transmitted. If no hold buffers are currently awaiting transmission and if the transmit buffer is empty, the task checks for a text indication message. If a text indication is found, the task formats a text message and set the flags.
Receiver	This task checks the incoming data stream for a message for this unit. When a message with the proper station address is detected, the message is checked for correct checksum and termination information. When all criteria are met, the message is passed to the application logic and flags are set to indicate the type of message received. In Slave Mode, the receiver itself determines the appropriate response to the message and triggers the transmitter to send that response or sets appropriate flags if the message is an acknowledgment to a former message transmission.
Transmitter	This task is activated whenever the protocol is ready to send a message. The task formats any non-data type message (xAD, xAB, xAA) and sends it. If a message requires a checksum, the task calculates it while sending the message's data, and then appends the checksum to the message. The task then checks the system flags and decides the appropriate terminator byte to use.
Control	This task is used in non-slave modes. Its function is to monitor system flags, timers and buffers to determine the sequence of messages needed to support the current mode of operation. This task activates the transmitter and directs which messages need to be sent.

Figure 3–7 and Figure 3–8 are flowcharts showing the assembly of either a received control or a sent indication message.

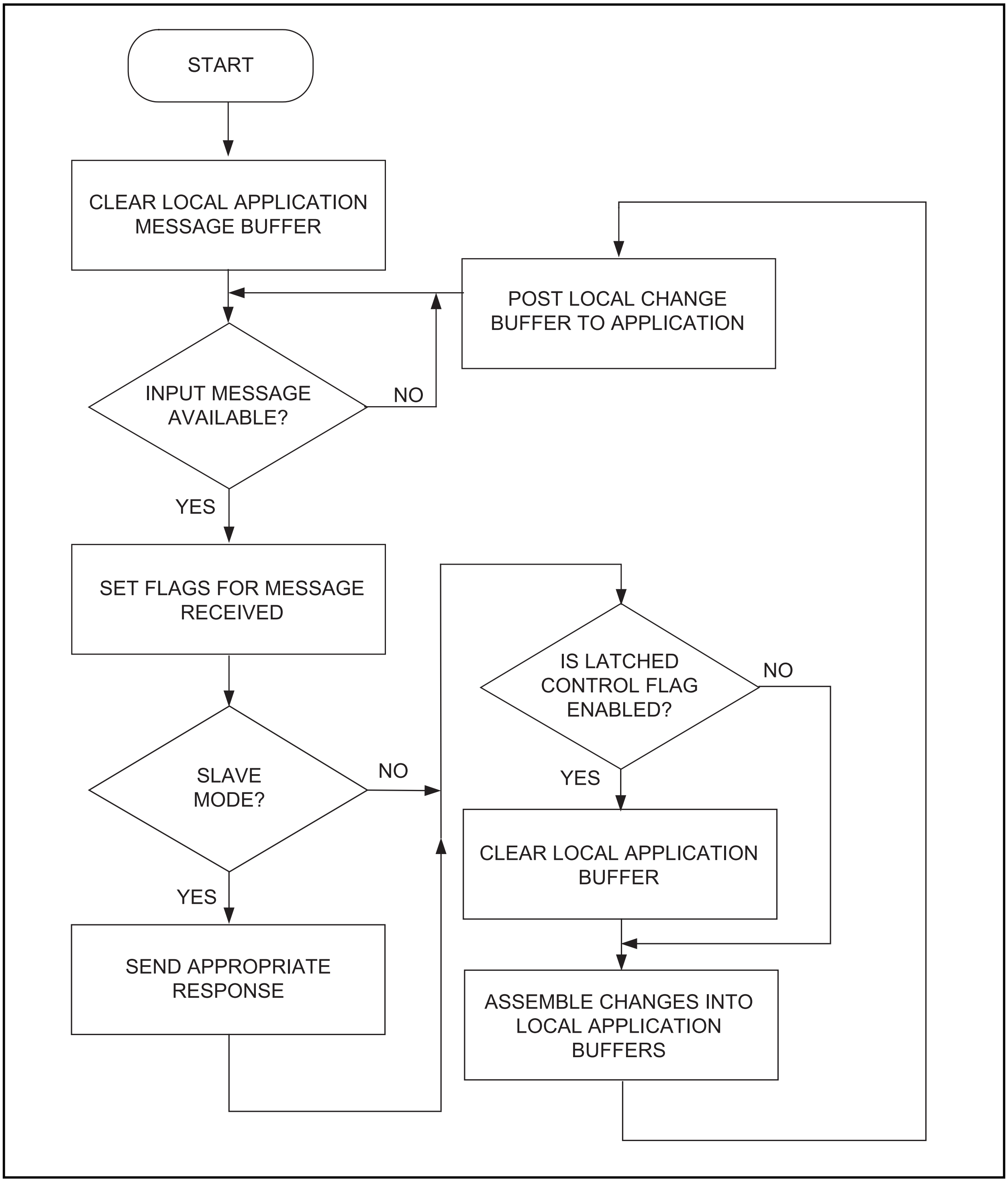


Figure 3–7. Control Message Data Flowchart

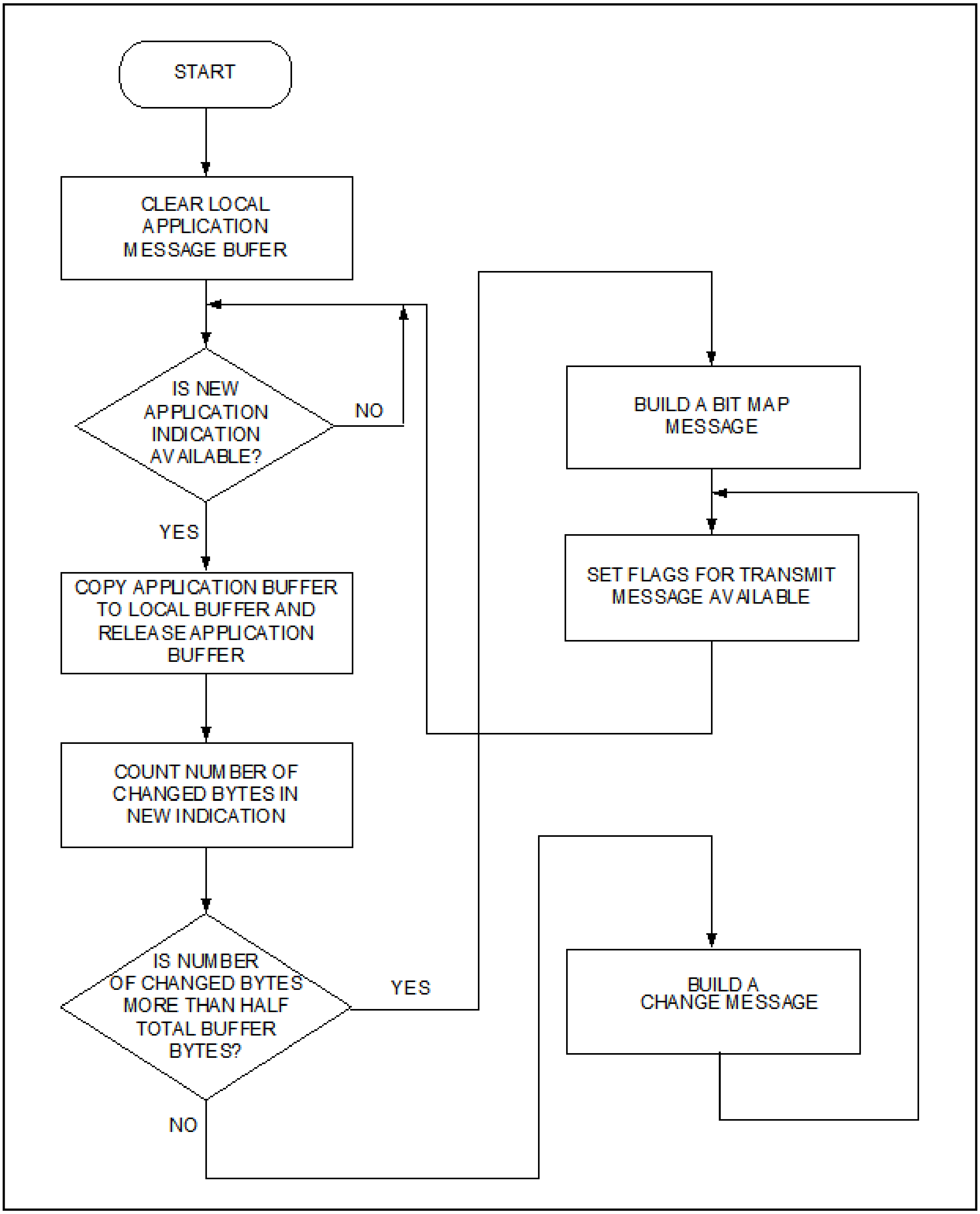


Figure 3–8. Indication Message Data Flowchart

SECTION 4 – DIAGNOSTIC AND USER INTERFACE

4.1 INTRODUCTION

This section discusses the protocol-specified diagnostic functionality implemented by the DT8 protocol. For further information about the diagnostic and user interface of the wayside non-vital processor components, refer to the system manuals.

WARNING

NON-VITAL COMMUNICATIONS SOFTWARE IS NOT FAIL-SAFE

The non-vital communications software is not designed for fail-safe application and must not be used for safety-critical operations.

Failure to comply can degrade the safety performance of the train control system resulting in property damage, injury, and/or death due to train collision or derailment.

4.2 MAC PORT FUNCTION

The MAC port provides access to the menu-driven displays of a wayside non-vital processor component. Some of these menus are used to report system status while others are used for manual control. MAC port users do not need a technical background or need to understand the details of the system design. However, users must understand the operation of the DT8 code system. Users connect to the MAC port via a VT-100, a video terminal that can communicate and exchange information with the wayside non-vital processing component. If a VT-100 is not available, the user can utilize a PC equivalent to emulate the VT-100, which must contain a software program that emulates the VT-100, and can capture the serial data stream. Display modes showing code-line traffic can be invoked, and the emulator enables the displayed information to be downloaded for future analysis.

The following sections provide more detail about the diagnostic menus.

4.3 EMULATION MENU

The Emulation Menu provides access to communication protocol emulations loaded in the wayside non-vital processing component. DT8 is assigned to a specific serial port in the non-vital application.

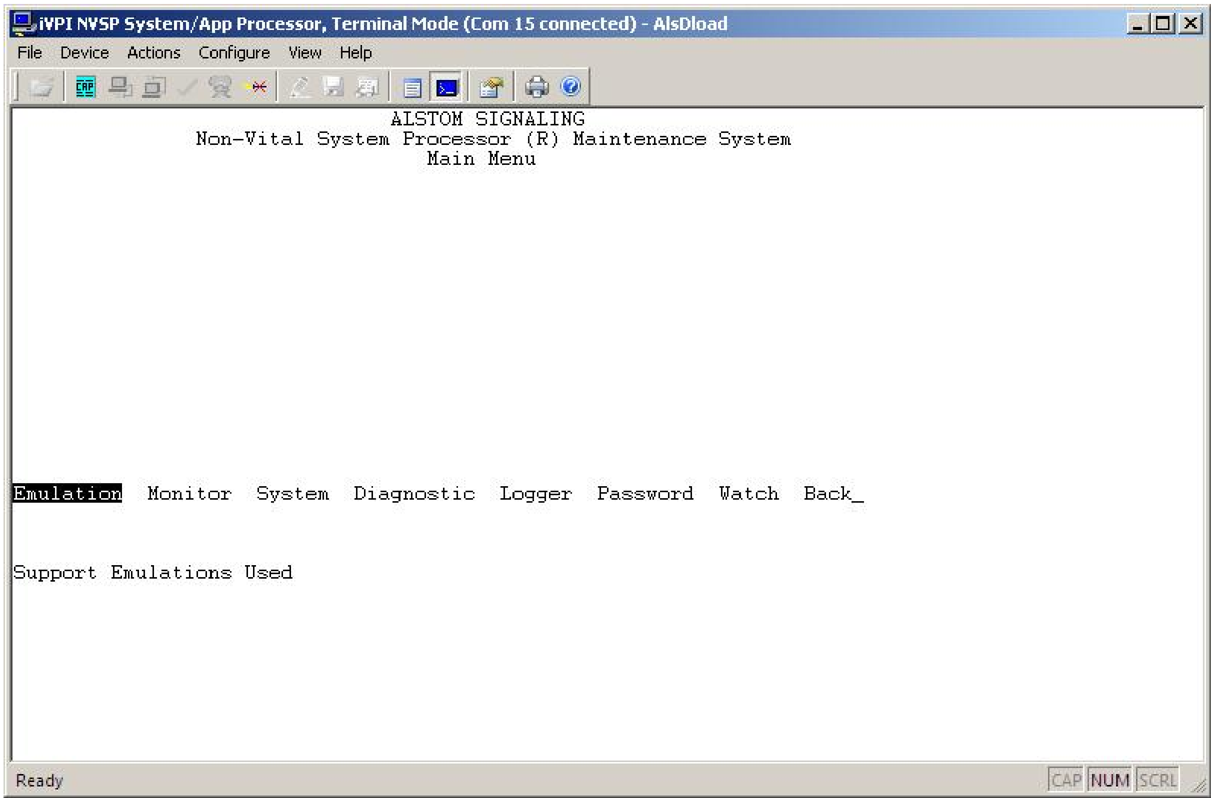


Figure 4–1. Emulation Menu

Table 4–1 contains a brief description of the options available in the Emulation Menu, after which these menu choices are described in more detail.

Table 4–1. Emulation Menu Choices

Option	Description
Port	Select the serial port of interest (enter the port number from 1 to 5) and view the port's setup. The serial port number must be specified before using the *Next*, *Last*, *Optns*, *Msg* or *Diags* menu choices.
Next	Display the addresses and current contents of control, indication and Special Message buffers for the next (or only) station on the selected serial port. If only one station is defined for the code-line port, the *Next* and *Last* options behave identically.
Last	Display the addresses and current contents of control, indication and Special Message buffers for the previous (or only) station on the selected serial port.
Optns	View the serial port setup including baud rate and data format (data bits, stop bits and parity).
Msg	Observe message exchange (controls and indications) between the non-vital application logic and the emulation, and enter and manually post a control or a Special Message to the application logic.
Diags	Execute protocol emulation specific diagnostics.
Reset	Reset the entire system. Use this choice with care since it causes the software to re-initiate its start-up sequence.
TWC	Access the Train-to-Wayside Communications (TWC) diagnostics menu (if applicable).
Vital_Diags	Access the optional Vital Diagnostic Protocol (VDP) diagnostics menu, described in Alstom publication P2346W.
Exit	Return to the Main Menu.

4.3.1 Emulation Menu Choice: *Port*

The *Port* menu choice allows the user to specify the serial port on which the communication protocol is installed. The application engineer must specify the serial port number before using the *Next*, *Last*, *Optns*, *Msg* or *Diags* options available in the Emulation Menu. Both the port usage and the type of protocol loaded are displayed.

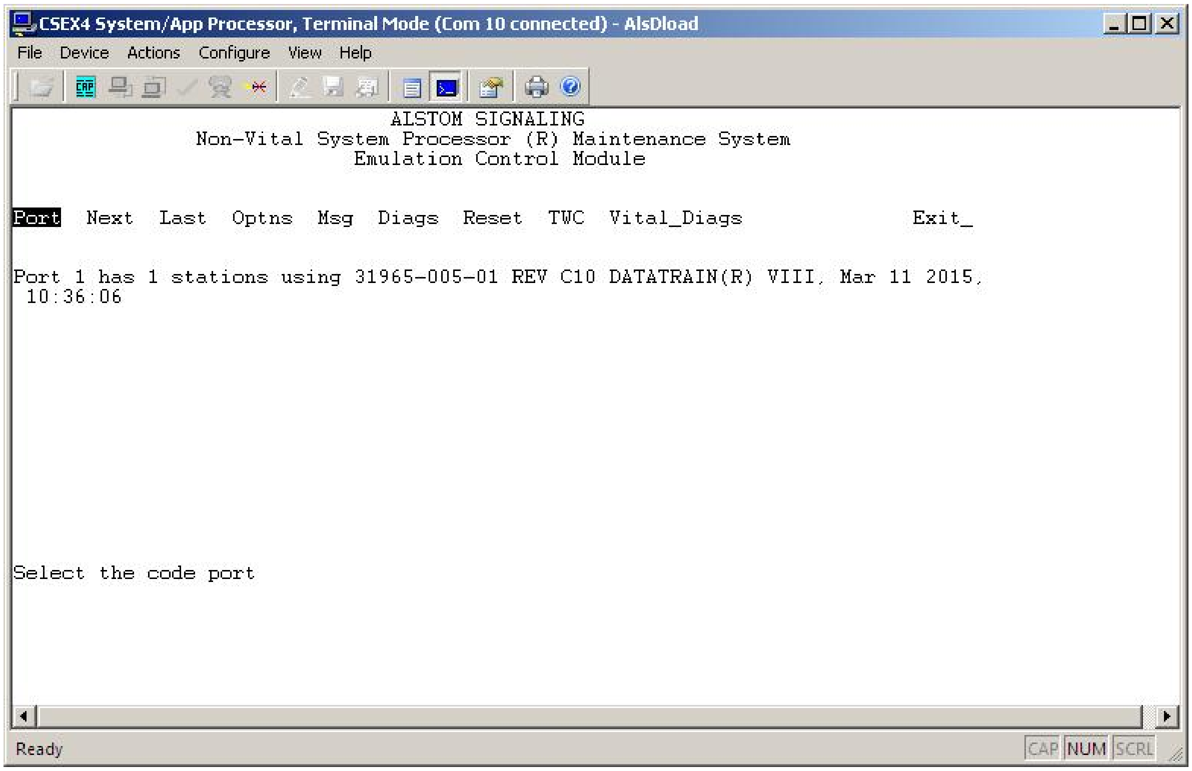


Figure 4–2. Emulation Menu – Select the Serial Port

4.3.2 Emulation Menu Choices: *Next* and *Last*

When the *Next* or *Last* menu choice is selected, the screen displays the current information for a single station once. This information is not updated in real time. The *Next* and *Last* menu options are most useful for a port having multiple stations. Otherwise information for station #1 is always displayed when *Next* or *Last* is selected.

The station information displayed includes:

- Control and indication station addresses and the total number of binary bits in each address
- The length of each message: control, indication, and special
- Each message's starting buffer address in RAM (intended for Alstom technical personnel only)
- The most recently posted control, indication and special messages

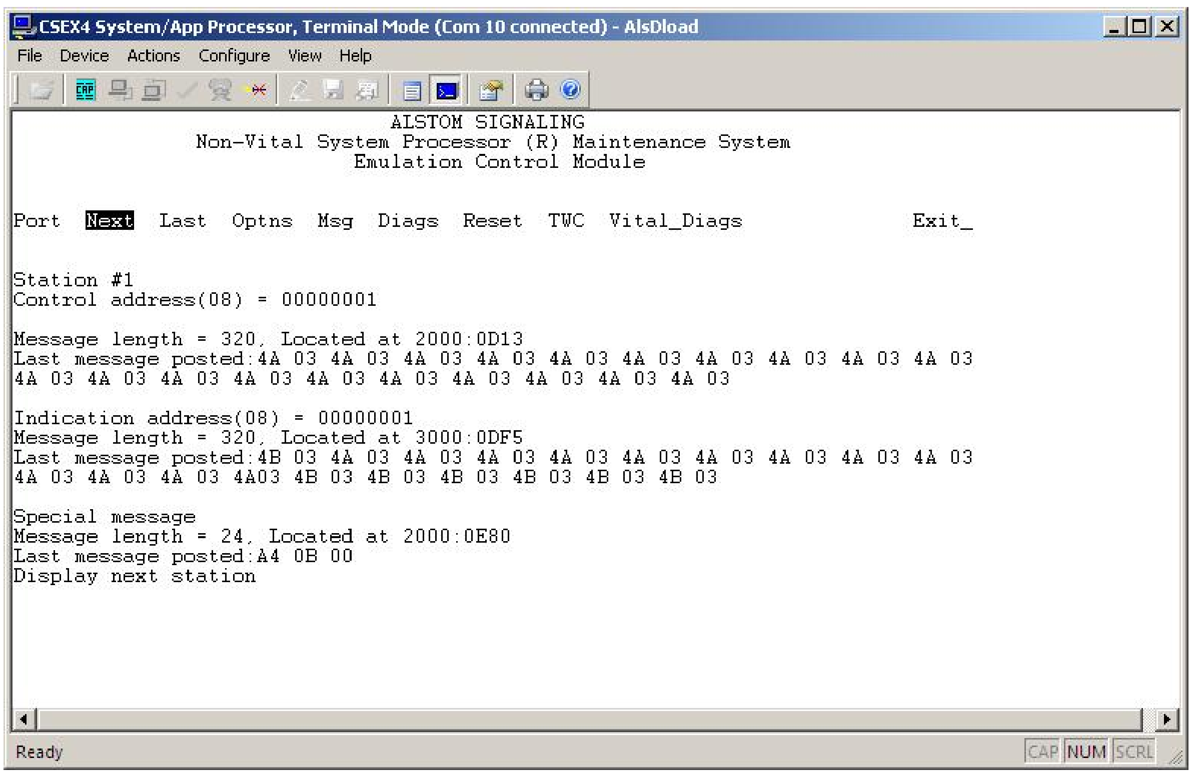


Figure 4–3. Emulation Menu – View a Station's Messages

Figure 4–3 shows that the control address for station #1 is 50 with a length of eight bits. The message itself contains 128 bits (parameters) and is located in memory beginning at address 0000:77EC. The message contents are shown as a series of 16 hexadecimal bytes, eight bits each. In this example, all 128 bits in the last received control message are false (0).

The indication address is also 50 and has a length of eight bits. The message itself is 536 bits in length and is located in memory beginning at address 0000:786D. The last indication sent is 00 00 00 00 … 00 00 00 00 10. Refer to Table 3–2 to convert data from hexadecimal to binary. In this example, no Special Message buffer is defined in the application for station #1.

4.3.3 Emulation Menu Choice: *Optns*

The *Optns* menu allows users to view the selected communication port's setup or baud rate.

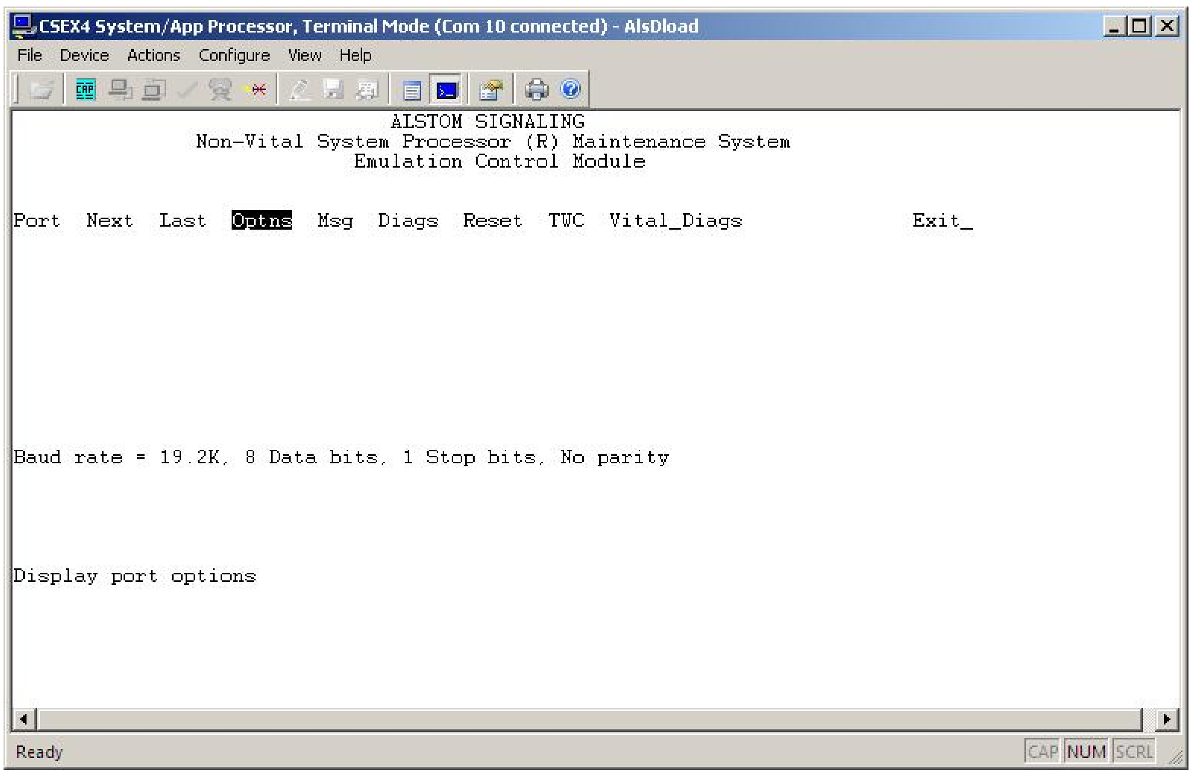


Figure 4–4. Emulation Menu – View Port Setup

4.3.4 Emulation Menu Choice: *Msg*

The Msg menu allows users to monitor the message flow between the communication protocol installed on a serial port and the non-vital application logic. The menu offers the following selections: *Post*, *Disp* (Display), *Spcl* (Special) and *Mode*.

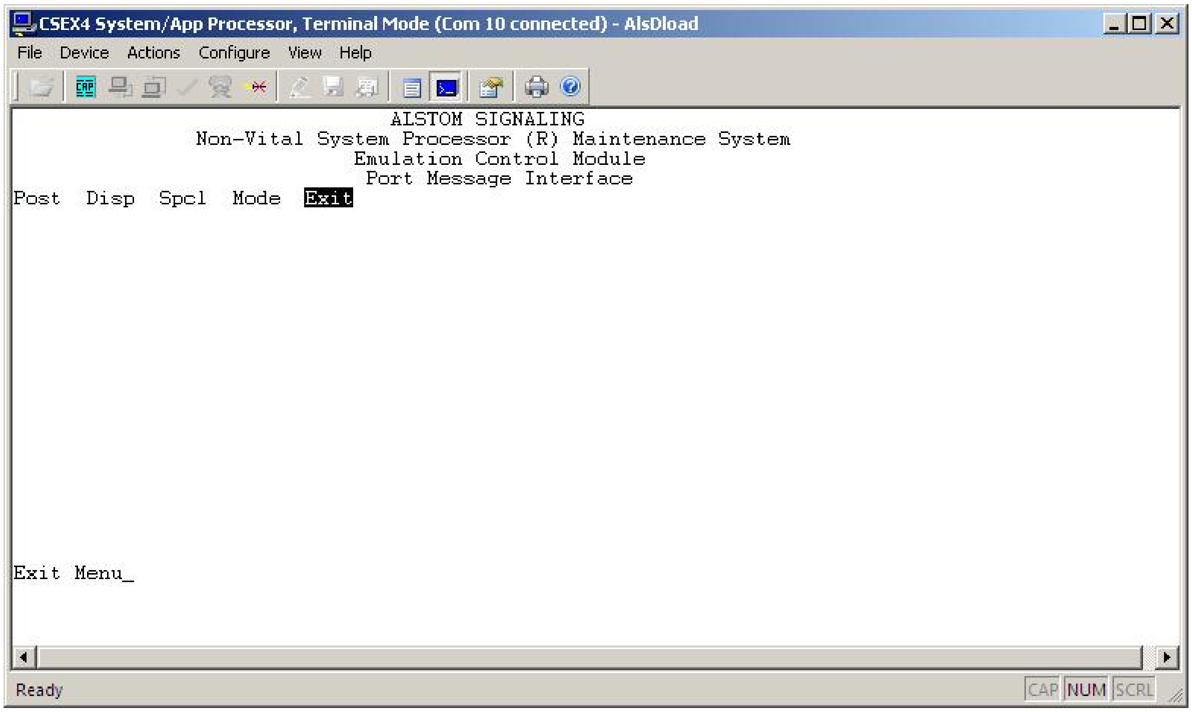


Figure 4–5. Emulation Menu – Message Submenu

4.3.4.1 Post Menu

The Post menu choice allows users to enter keyboard input to replace the control message normally posted by the protocol emulation. This feature helps determine if a problem is in the application or the emulation.

To post a control, select the Post option. If more than one station exists, the screen prompts the user for the station number. The user enters the proper station number. The screen displays the maximum number allowed. After a prompt showing the size of the control message displays, the user enters the new data as a series of hexadecimal bytes. When the entry is complete, the message is posted for processing by the application logic. At most 96 bits (12 bytes) can be manually posted. If DT8 receives a valid control message through the serial port, the protocol posts it and overwrites the control that was manually posted.

Figure 4–6 shows an example screen if the Post option is selected for a control message containing 16 bits (2 bytes).

Note: Posting control data in this manner causes changes to the operation of the system.

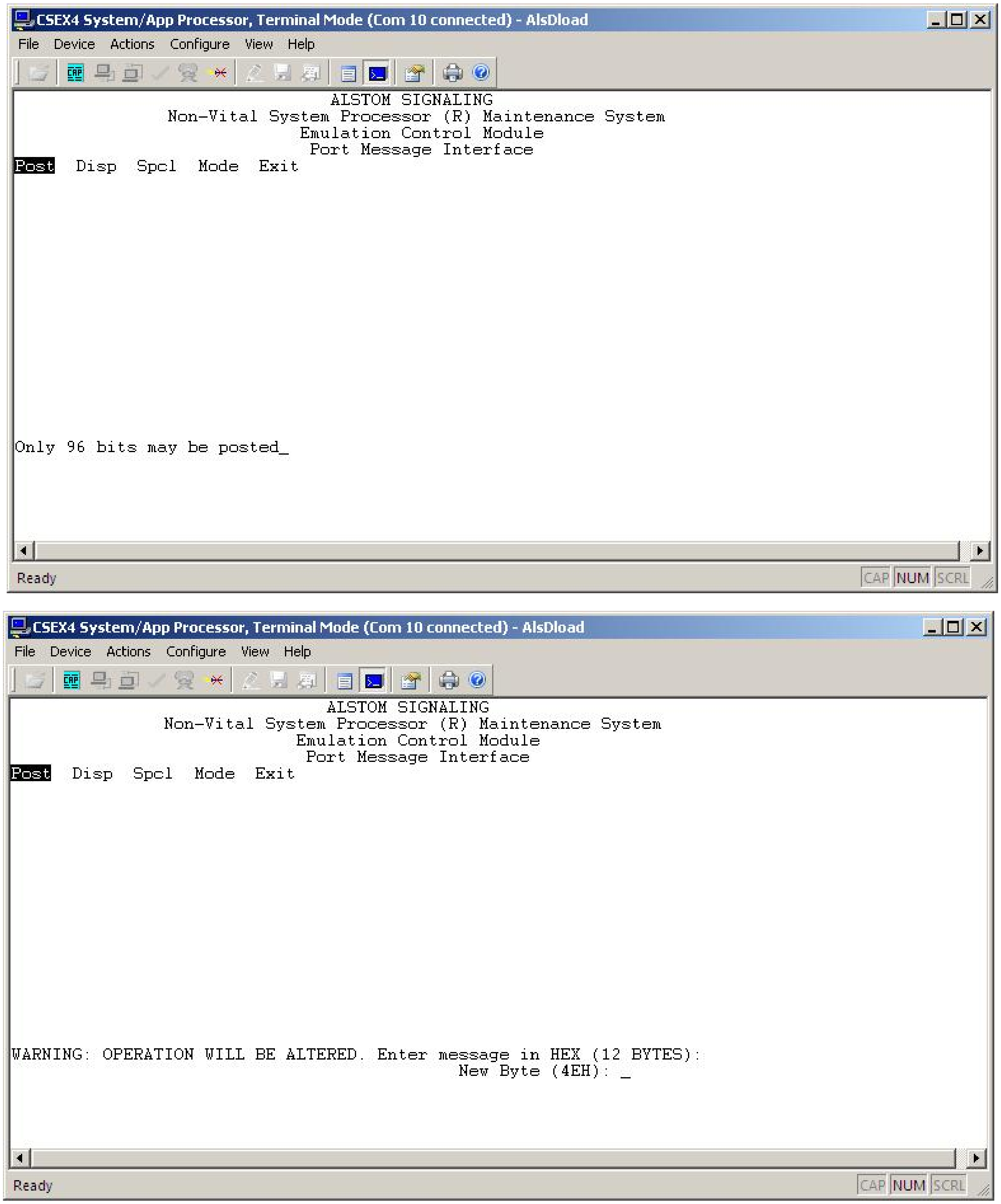


Figure 4–6. Post a Control Message

4.3.5 Disp Menu

The *Disp* menu choice displays the data portion of messages posted between the protocol emulation and the non-vital application logic, updated in real-time. Figure 4–7 shows a typical screen when the display format is set to “Original”.

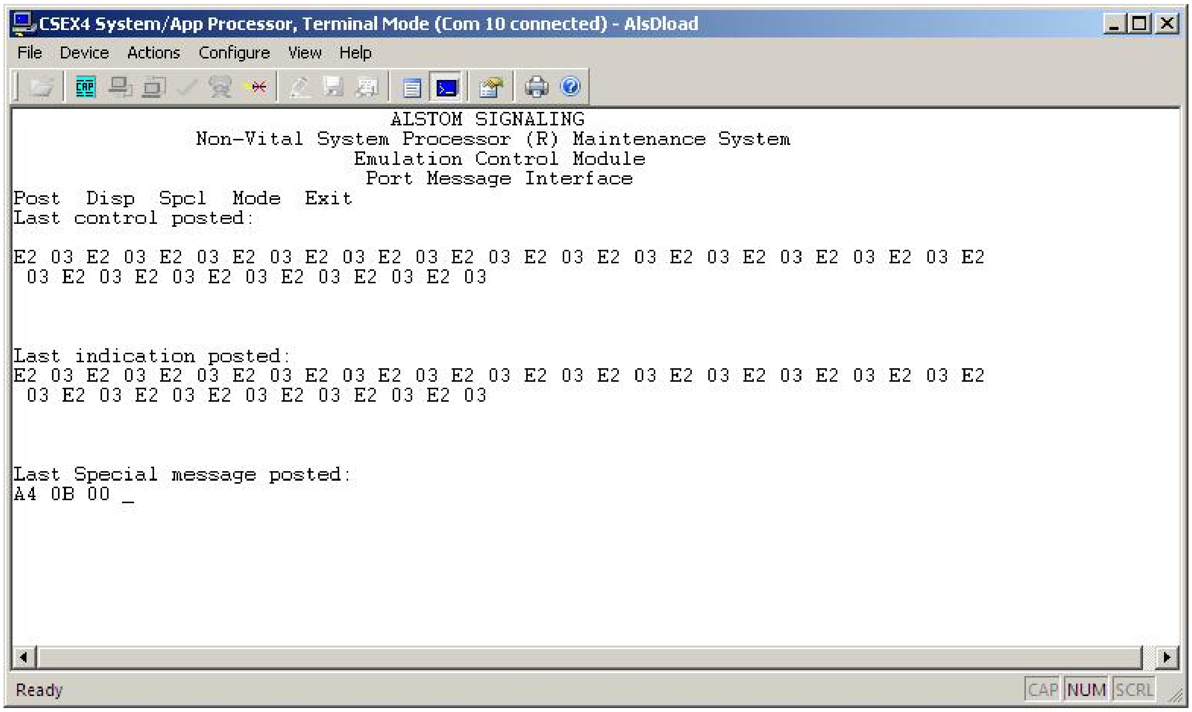


Figure 4–7. Display Messages in Real-Time

4.3.5.1 *Spcl* Menu

The *Spcl* menu choice allows user-entered keyboard input to be posted to the protocol's Special Message buffer. The use of the flags in the Special Message buffer is described in SECTION 2 – CAAPE Package.

To post a Special Message, select the *Spcl* option. If more than one station exists, a prompt for the station number appears. In this case, enter the proper station number (the maximum number allowed is displayed). After a prompt showing the size of the Special Message appears (typically three bytes containing 24 flag bits), enter the new data as a series of hexadecimal bytes. When the entry is complete, the application logic and the protocol emulation posts the message. If DT8 receives messages through the serial port, the protocol may write to the Special Message buffer and potentially overwrite the information that was manually posted.

Figure 4–8 shows a screen if the *Spcl* option is selected for a Special Message containing 24 bits (3 bytes).

Note: Posting Special Message data in this manner causes changes to the operation of the system.

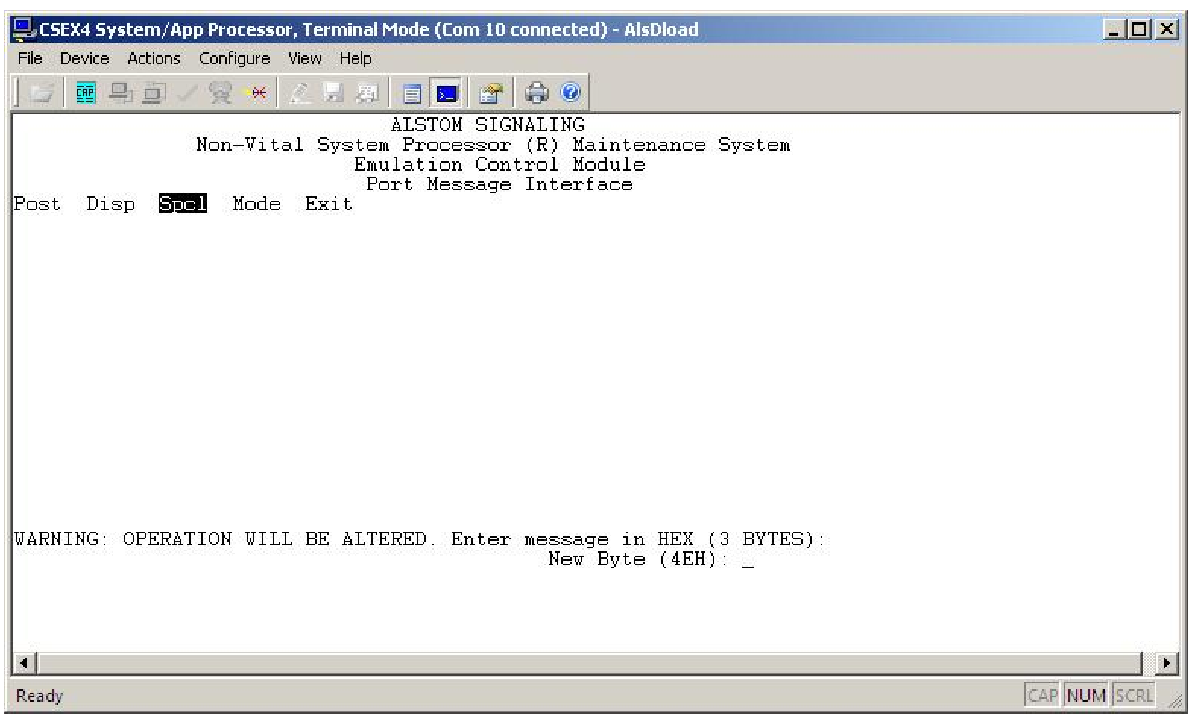


Figure 4–8. Post a Special Message

4.3.5.2 Mode Menu

The *Mode* menu choice provides several additional menu options that affect the format of control, indication and Special Message data viewed from the *Disp* option. After the display format has been set to Original, New Hex, Binary or Inverse, message contents are automatically displayed and updated in real-time as if the *Disp* option had been selected.

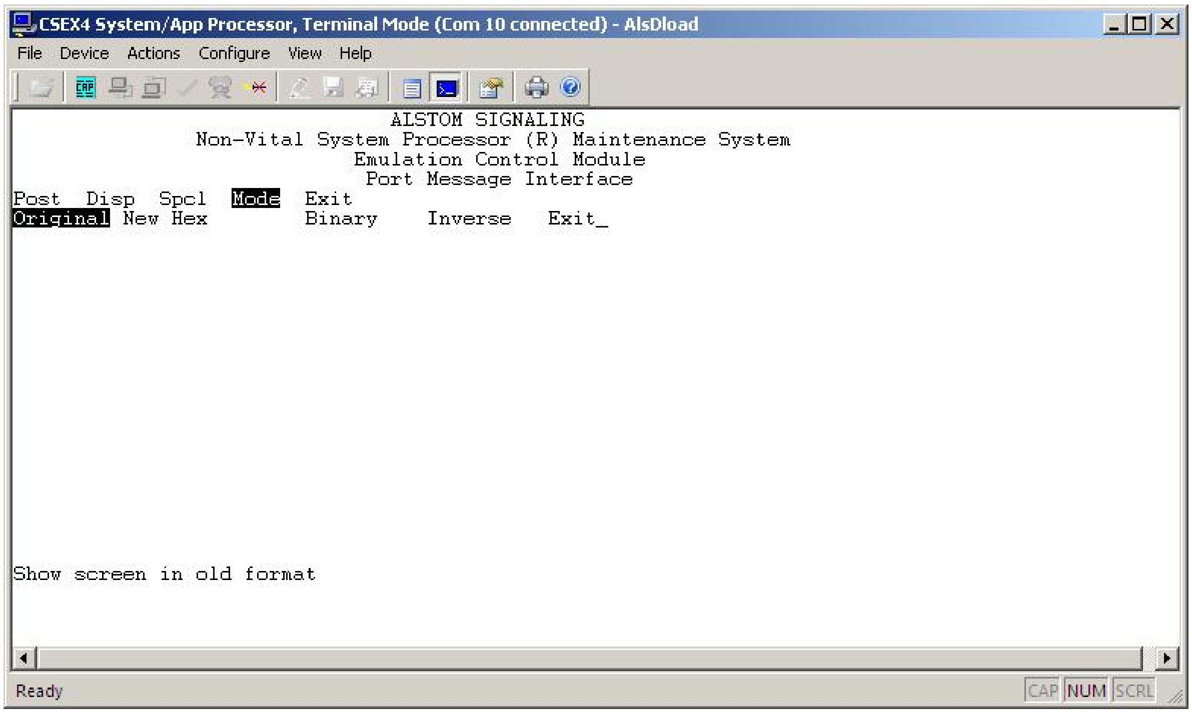


Figure 4–9. Select the Message Display Format

Table 4–2 describes the various display formats available from this menu.

Table 4–2. Select the Message Display Format

Option	Description
Original	Restores the display format to the default setting. In this format, data for a single station is shown as a series of hexadecimal bytes, wrapping between lines on the display based on the message lengths for controls, indications and Special Messages.
New Hex	In this format, message data is shown 24 hexadecimal bytes at a time from left to right on a single line per station for up to four stations, with the station number shown at the start of each line of data. Multiple stations' data (up to four) are shown on subsequent lines as needed. If a message (control, indication or Special) exceeds 192 bits (24 bytes), use the 'V' (increment byte numbers) and 'B' (decrement byte numbers) keys to bring additional data bytes (one at a time) into view. At most 24 bytes of a message's data can be viewed at a time in this format. If more than four stations are defined on the port, press G+Space Bar to view data for additional stations. Press the 'H' key to bring the display back to the first station, first byte setting.
Binary	This display format is similar to New Hex except that message data is shown eight bytes at a time in binary (eight bits per byte). The least significant bit of each byte is shown at the left of each byte of data. For example, the hexadecimal byte 0x01 is shown as 10000000 in Binary format. Use this format to view message data exactly as posted to the non-vital application logic. Figure 4–11 shows the same data as in Figure 4–10 (New Hex format), but in Binary format instead.
Inverse	This display format is similar to Binary except that the least significant bit of each byte is shown at the right of each byte of data. For example, the hexadecimal byte 0x01 is shown as 00000001 in Inverse format.

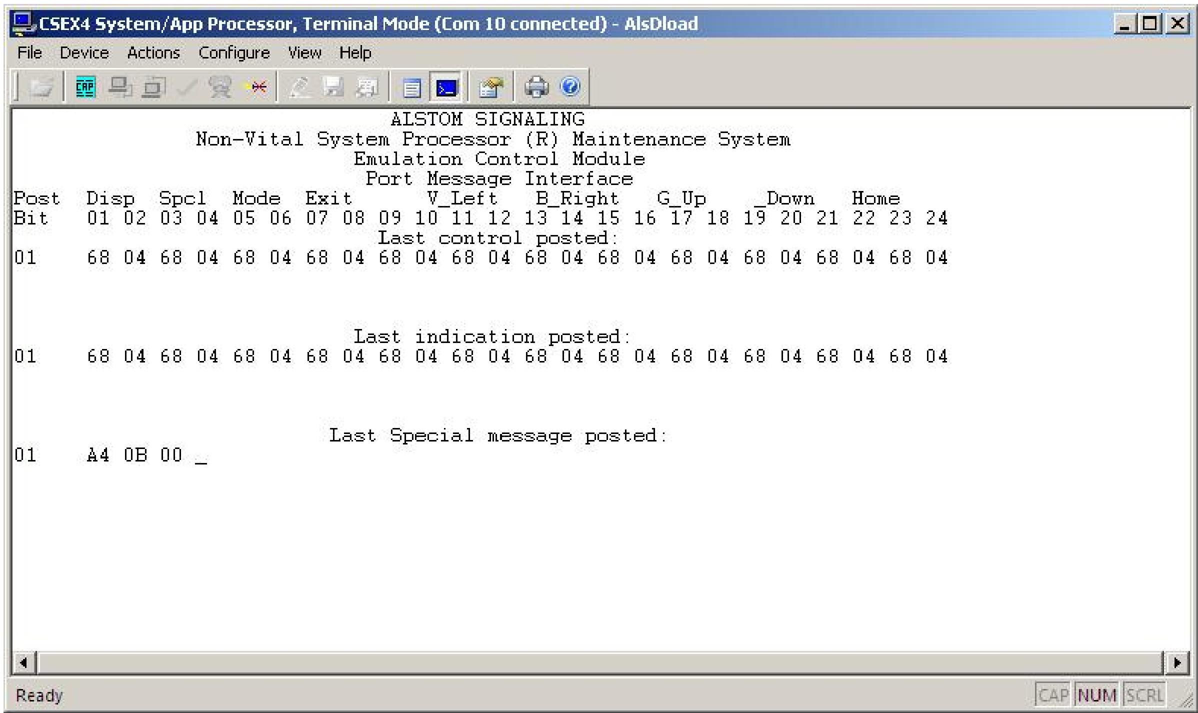


Figure 4–10. Display Messages – New Hex Format

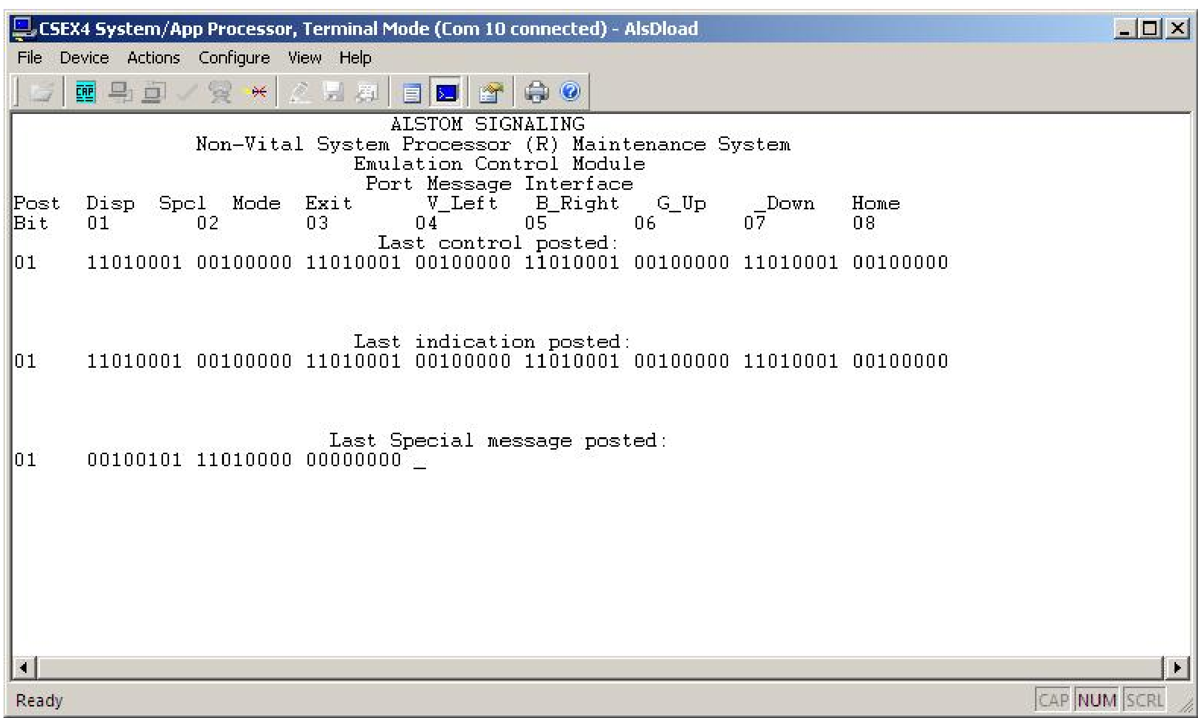


Figure 4–11. Display Messages – Binary Format

4.3.6 Diags Menu

The Diags menu provides access to protocol-specific diagnostic functions. Figure 4–12 shows the DT8 Diagnostics Menu.

Note: Some options in this menu allow the user to modify DT8 operating settings such as enabling or disabling the CTS Flag. These settings are normally specified in the application's LPC file and affect operations. Any changes made via the DT8 Diagnostics Menu are temporary only, and is lost upon system reset. If a user password has been specified in the non-vital application, it must be entered before modifying any DT8 operating setting.

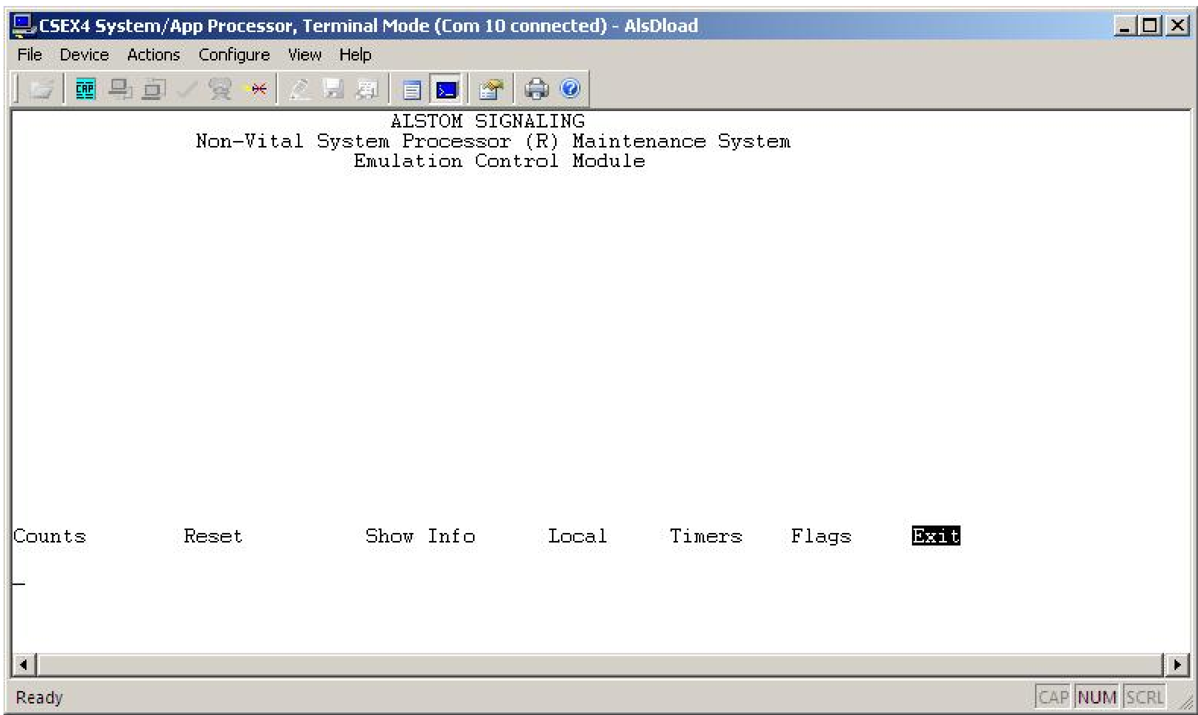


Figure 4–12. DT8 Diagnostics Menu

The following pages describe the options available in the DT8 Diagnostics Menu.

4.3.6.1 Counts Menu

The *Counts* menu choice enables users to check the status of communication on the selected port. This dynamically updated screen can be out of step with actual I/O because of a particular combination of application choices. Figure 4–13 shows a typical DT8 messages counts screen.

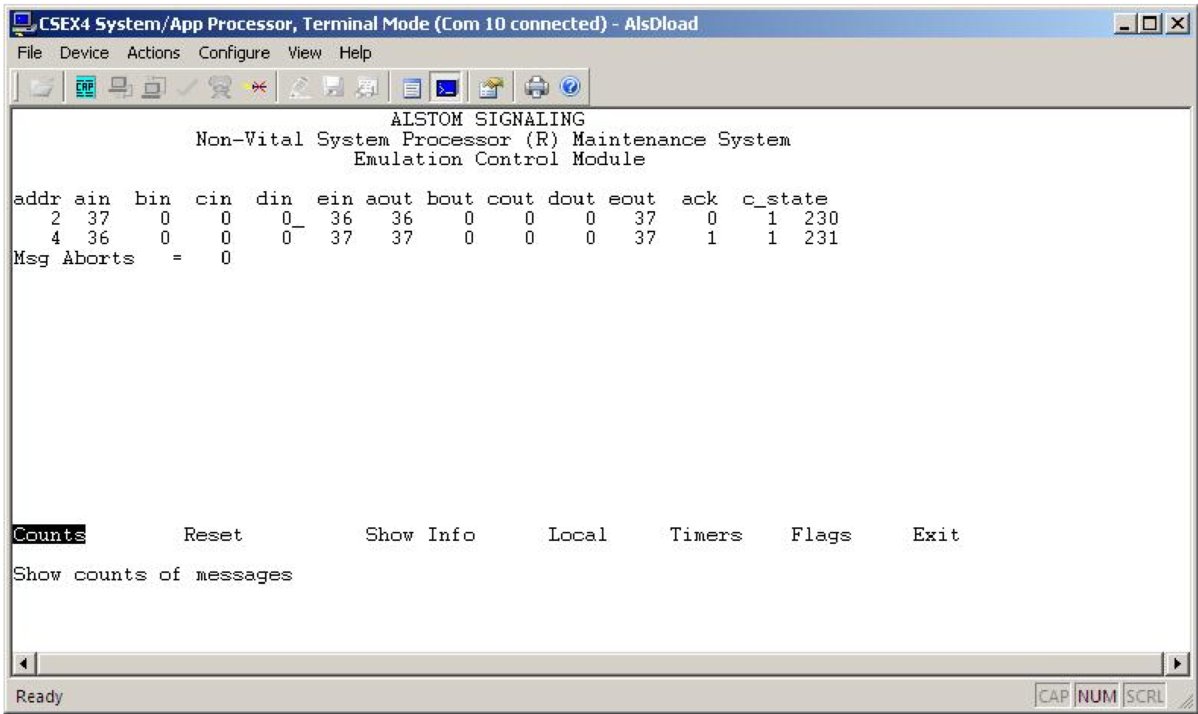


Figure 4–13. Messages Counts Screen

4.3.6.2 Reset Menu

The Reset menu allows the user to reset the number of messages both sent and received back to zero after they exceed 255. Depending upon the operating mode of DT8, some of these counts can remain zero. This action does not affect system operation.

Table 4–3. Key to DT8 Messages Counts Information

Item	Description
Addr	The station address, in decimal.
ain	The total number of Acknowledge (xAA) messages received.
bin	The total number of Poll (xAB) messages received (request for information).
cin	The total number of Change (xAC) messages received.
din	The total number of Bitmap Request (xAD) messages received.
ein	The total number of Bitmap (xAE) messages received.
aout	The total number of Acknowledge (xAA) messages sent.
bout	The total number of Poll (xAB) messages sent (request for information).
cout	The total number of Change (xAC) messages sent.
dout	The total number of Bitmap Request (xAD) messages sent.
eout	The total number of Bitmap (xAE) messages sent.
ack	The total number of messages sent since receipt of the last Acknowledge (xAA).
c_state	Monitors bit map conditions while operating in Peer Mode: - 0 = No bit maps have been sent - 1 = A bit map has been sent, and DT8 is awaiting a bit map in response - 3 = A bit map has been sent, and a bit map has been received in response The number displayed after the c_state value is a test counter that can be ignored.
Msg Aborts	The total number of times a received message was corrupted or interrupted before completion. This counter is an aggregate for all stations.

4.3.6.3 Show Info Menu

The *Show Info* menu choice shows a single station's messages counts and the contents of the station's eight-deep indication queue.

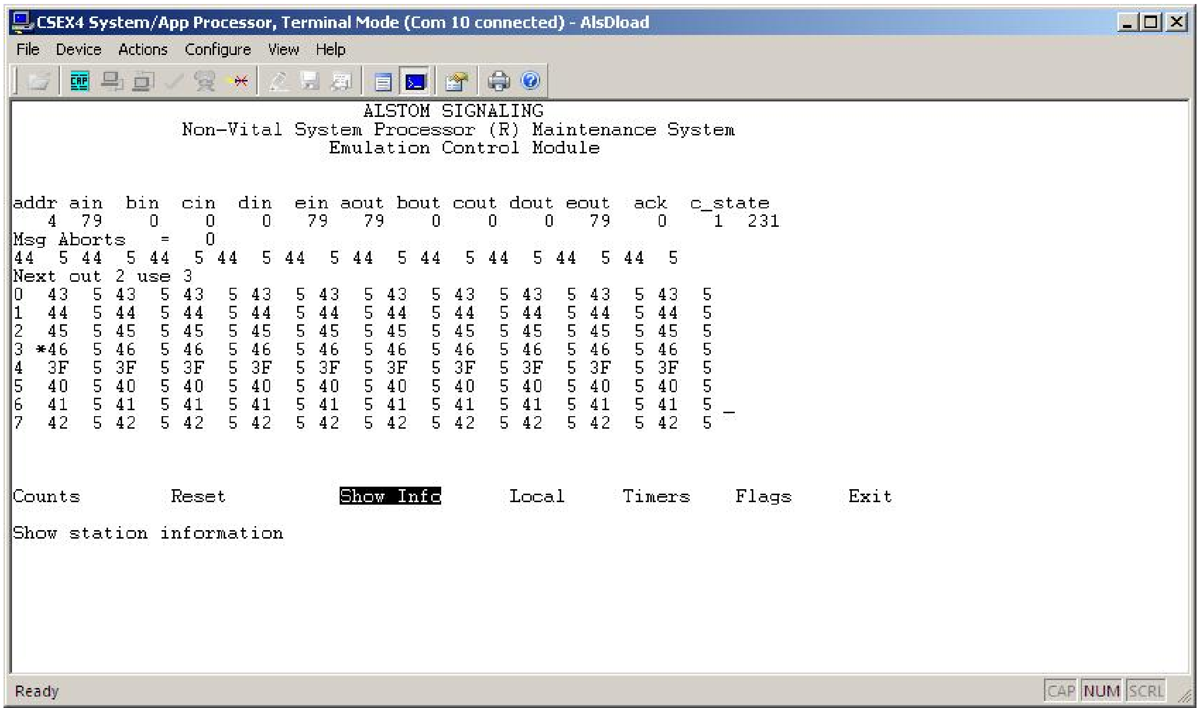


Figure 4–14. Show Info Screen

The following information also appears in the Show Info Screen:

- The current indication bit map as reported by the application logic (00 01 02 00 in 5-14).
- Next out: identifies the buffer number (0 to 7) in the indication queue that the system transmits next, buffer #3 in the example above.
- Next use: identifies the buffer number (0 to 7) that the system uses to build the next outgoing message, buffer #5 in the example above.
- The current contents of the eight-deep indication queue. The content of each buffer in the queue has either been sent or is waiting to be sent (or has not yet been used). If an asterisk (*) precedes a buffer's contents, the data is queued and ready to be sent (pending), or was sent but not yet acknowledged. The column of numbers from 0 to 7 indicates the buffer number in the queue, and the hexadecimal bytes following are the contents of the indication bit map in each buffer.
- In the Show Info Screen, the indication data shown are bit-inverted, meaning that bits are in reverse order on a byte boundary (as shown on a code chart). In Table 4–4, the Code Chart line contains numbers assigned to an indication message's parameters in the application's communication file (CSS). Since indication parameters are organized in groups of eight, the Hex Weight is the bit position (in hexadecimal) for each parameter in the message. Example True/False states for the sixteen parameters in the message are shown on the Bit State line.

Table 4–4. Bit Inversion of Displayed Indication Data

Parameter Number in Application Input File (Code Chart)																
Code Chart	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Hex Weight	01	02	04	08	10	20	40	80	01	02	04	08	10	20	40	80
Bit State	T	F	F	F	F	T	F	F	T	T	F	F	F	F	T	F

The hexadecimal value as shown in the Show Info Screen can be determined by taking the Hex Weight of each True bit and summing them for a given byte (eight consecutive parameters, starting with parameter #1). Given the parameter True/False states shown above, the hexadecimal values displayed in the Show Info Screen are x21 and x43. Using the data in Figure 4–15 for indication buffer #3 (for example 01 01 02 4A), it follows that the True/False states of these 32 parameters, beginning with parameter #1, is TFFFFFFF TFFFFFFF FTFFFFFF FTFTFFTF.

4.3.6.4 Local Menu

The *Local* menu choice displays the current DT8 operating mode, Peer, Master, Slave, or Sync. The user can change the local unit's mode of operation.

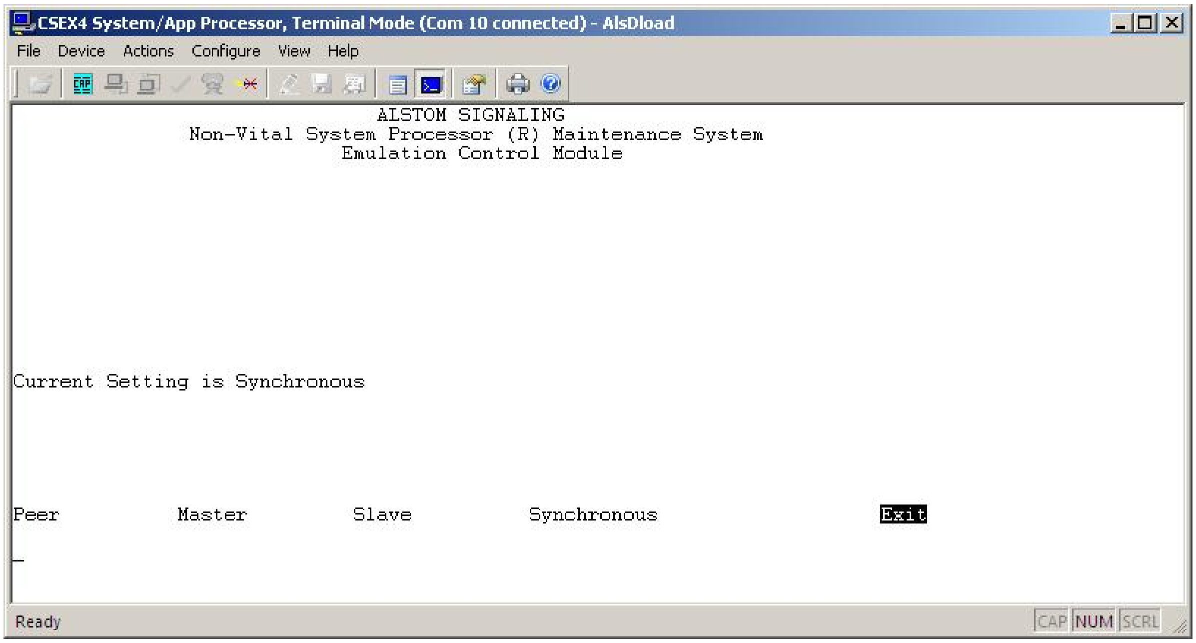


Figure 4–15. Local Screen

4.3.6.5 Timers Menu

The *Timers* menu choice allows the user to view and change the interval timers.

Note: The AD Time Interval is called the AE Time Out elsewhere in this manual.

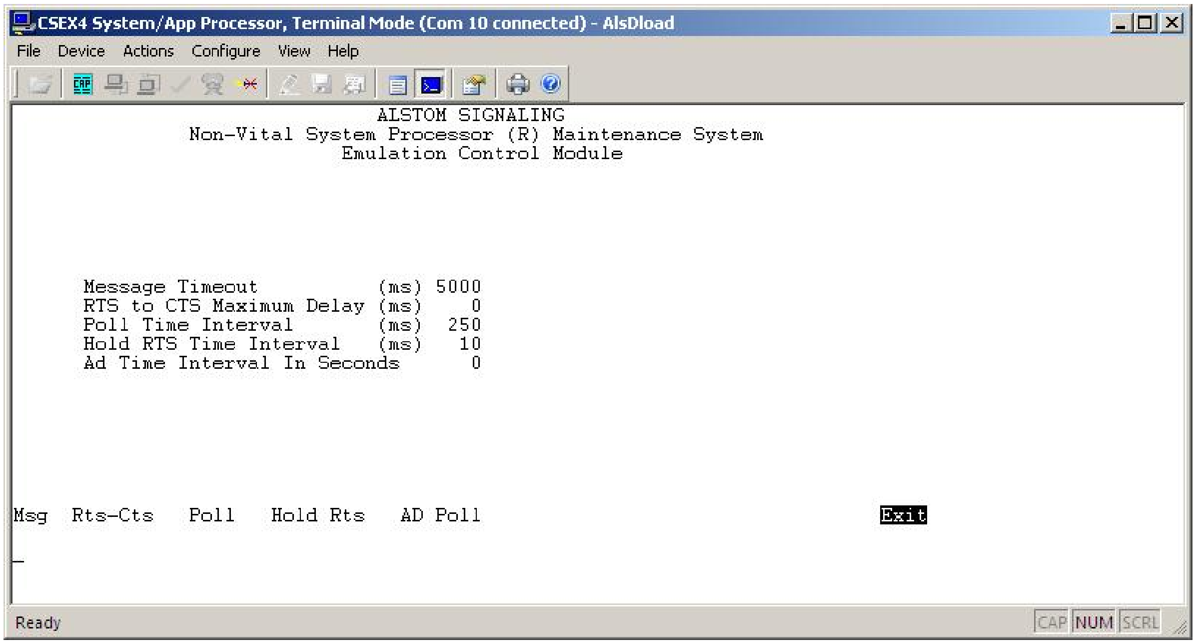


Figure 4–16. Timers Screen

4.3.6.6 Flags Menu

The *Flags* menu choice allows the user to view and change the settings of flags used by DT8 for the handling of the RTS and CTS control lines.

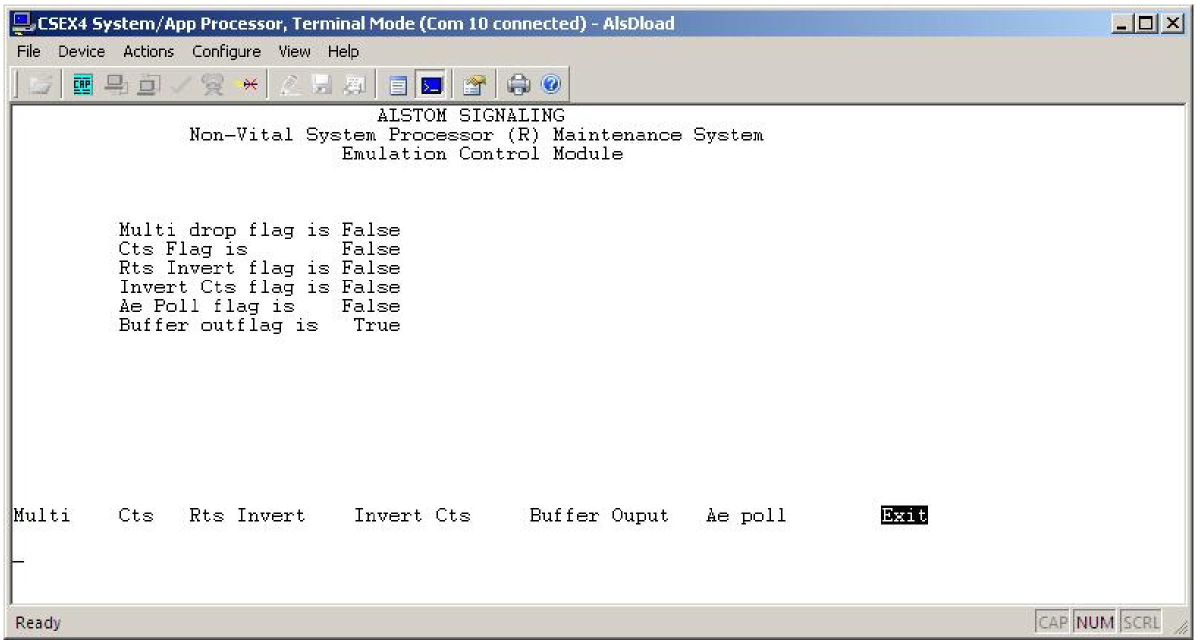


Figure 4–17. Flags Screen

4.3.7 Data Monitor Screen

The Data Monitor Screen is viewed by selecting *Monitor* from the Main Menu. Use the data monitor to observe message traffic on one or more code-line ports.

Selection Path:

- Main Menu
- Monitor
- Port (port selection from 1 to 5)

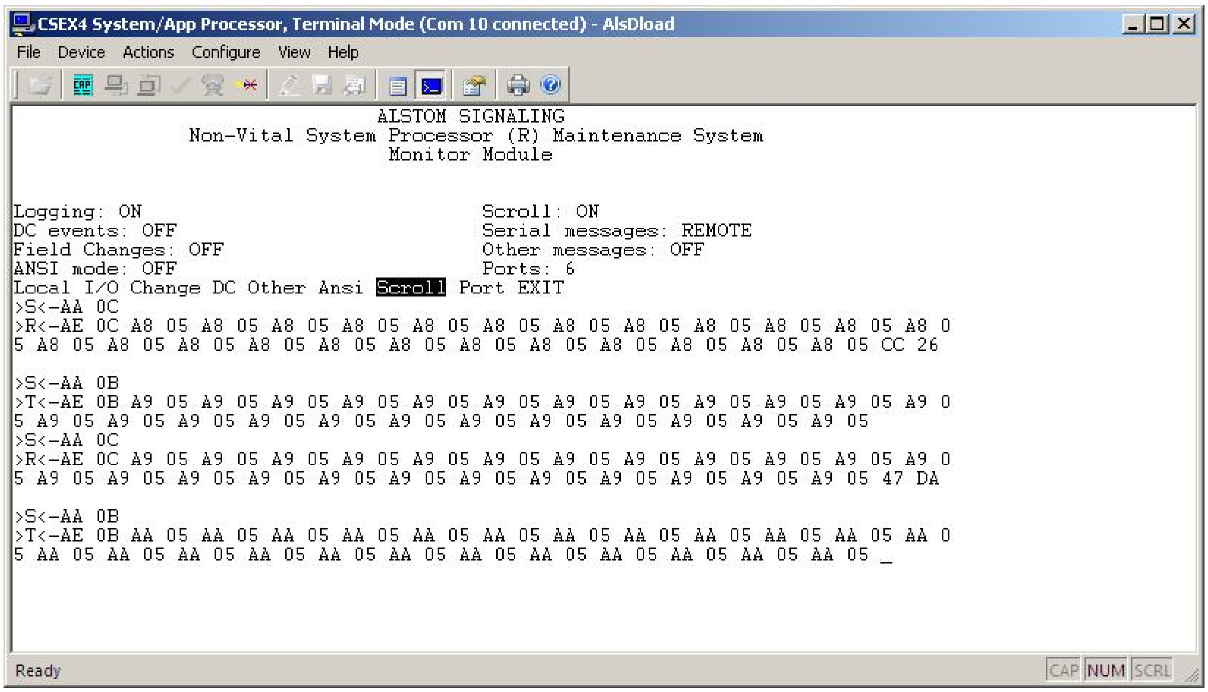


Figure 4–18. Data Monitor Screen

Data monitor operation is a background task. The displayed information can be preempted by other processor operations of a higher priority. As a result, it cannot be possible to report all code-line port message traffic in monitor mode. Split bytes, CRC checksums, and terminator bytes are not shown on this display.

Table 4–5 provides a description of the choices available in the Monitor Menu.

Table 4–5. Monitor Menu Choices

Option	Description
Local	Toggle the Local/Remote serial message flag. When in Local mode, only local messages are displayed.
I/O	Toggle the input/output status display.
Change	Toggle the field changes display. When on, field I/O are displayed in order of their occurrence.
DC	Display DC type events (not used in electronic code applications).
Other	Display communication error messages and other miscellaneous message types.
Ansi	Select ANSI mode in which the unit sends control codes to format the screen so that data appears in organized order. These codes cannot be used if a printer or other devices is connected to the port.
Scroll	Start or stop updating the screen so that message traffic can be monitored and displayed in real time.
Port	Select the number of the port (or ports), from 1 to 5, to monitor and display. Activity on multiple ports can be observed at the same time.
EXIT	Exit the data monitor. **Note:** Ctrl + ‘E’ should not be used to exit from the Data Monitor Screen.

Table 4–6 lists the Data Monitor Status Indicators displayed at the top of the screen.

Table 4–6. Data Monitor Status Indicators

Field	**Description**
Logging	On/off status of the data monitor; this is normally turned on.
Scroll	On/off status of the display; turn “Scroll” on for real-time display of message traffic.
DC Events	On/off status of the display of DC data; this is normally turned off.
Serial Messages	Serial message type (remote or local); this is normally set to remote.
Field Changes	Field changes display is turned on or off; this is normally turned off.
Other Messages	Other messages display is turned on or off; this must be on for error messages to be displayed.
ANSI Mode	ANSI display mode is turned on or off; this is normally turned off.
Ports	Reports message traffic for only the code-line ports displayed.

When “Scroll” is on, message traffic on the selected port(s) is displayed. A descriptor indicating the type of message received or transmitted precedes each message.

Table 4–7. Data Monitor Message Descriptors

Message Type	**Description**
>T<	Transmitted message (e.g., an indication).
>R<	Received message (e.g., a control).
>P<	Received poll message (Slave Mode only).
>S<	Supervisory or status message.
>-CFG-<	Received system configuration message.
ERROR	The received message contained an error.

Communication problems can be monitored by using this screen and are evidenced by the display of error messages.

Note: The Other Messages option must be turned on in order for error messages to be displayed by the data monitor.

4.4 PORT DIAGNOSTIC LEDS

Several LEDs on the front edge of CenTraCode II provide visual indication of serial port operation. This section describes the operation of these LEDs on serial ports that are configured to utilize the DT8 protocol.

A serial port LED is turned on to indicate the reception of a message, and another LED indicates that a message is transmitting. If the LEDs do not periodically change state (turn on and off), the following conditions are possible:

- the communication channel has failed
- the state is the same for each message
- the port used by the NVIP module is not being addressed
- there is a wiring or connection problem to the remote module
- the remote module is turned off

Serial Ports 1 and 2 have a group of four LEDs controlled by the CenTraCode II System Software and the protocol emulation that are used to indicate reception and transmission of messages. The other serial ports' LEDs are controlled by the CenTraCode II hardware and only indicate character input/output on a given port.

CSEX3 boards include a 2-character diagnostic display whose function is controlled by an on-board diagnostic switch. The function code shown on the diagnostic display must read “00” in order for the serial port LEDs to operate as described below. If the diagnostic display does not show “00”, the display and the LEDs are being used for a diagnostic function or the display is indicating the occurrence of a system error. A serial port LED is turned on to indicate the reception of a message, and another LED indicates message transmission. If the LEDs do not periodically change state (turn on and off), the communication channel has failed, the state is the same for each message, or this unit is not being addressed. A receive error LED is lit when an error is detected.

Table 4–8 summarizes the various communication LEDs controlled by CTC2 with DT8 protocol. In the discussion of each board type, an illustration of the board edge including LED labels is included.

Table 4–8. Communication LED Functions CSEX[1], CSEX2 and CSEX3 Ports 1 and 2 for Standard DT8 Protocol

LED	Description
Normal	This is always dark and is not used by the DT8 protocol.
Receive Error	This should be dark. It illuminates when the protocol receives an unrecognized message format. The LED stays illuminated until an Address OK or valid address message is received.
Invalid Address Received	This LED illuminates when the protocol receives a valid message, but does not match any assigned to this Port. It remains illuminated until an Address OK or valid address message is received.
Transmitting Response	This LED flashes, illuminating steadily during the time the serial communication port transmitter is attempting to communicate with the remote unit.
Valid Address Received	This LED illuminates when a complete and valid message is received and the address matches ones assigned to this Port. It remains illuminated until an Invalid Address is received or a Receiver Error is detected.

Table 4–9. Communication LED Functions CSEX[1], CSEX2 and CSEX3 Ports 3, 4, and 5 for Standard DT8 Protocol

LED	Description
Transmitting Response	This LED flashes, illuminating steadily during the time the serial communication port transmitter is attempting to communicate with the remote unit.
Receiving Response	This LED flashes, illuminating steadily during the time the serial communication port receiver is communicating with the remote unit.

Table 4–10. Communication LED Functions CSEX4 and NVSP Ports 1 and 2 for Standard DT8 Protocol

LED	Description
Transmitting Response	This LED flashes, illuminating steadily during the time the serial communication port transmitter is attempting to communicate with the remote unit.
Receiving Response	This LED flashes, illuminating steadily during the time the serial communication port receiver is communicating with the remote unit.

THIS PAGE INTENTIONALLY LEFT BLANK.

SECTION 5 – HARDWARE DESCRIPTION AND TROUBLESHOOTING GUIDE

5.1 GENERAL

This section describes the hardware platforms on which DT8 operates, and troubleshooting information.

This section shows layouts of the circuit boards with the positions of the LEDs (Light Emitting Diodes) identified along with other primary components. Diagnostic LEDs are shown and application switches are explained, as well as any user configurable hardware jumpers and switches on the supported hardware platforms. In the tables showing jumper settings, “N/C” means No Connection, in which case the jumper should be removed. Reference publications for each hardware platform are identified. Lastly, a troubleshooting guide is provided.

5.2 HARDWARE

5.2.1 Diagnostic LEDs

To evaluate the operation of the hardware, it is important to distinguish between indicators that verify normal operation and those that identify problems with communication. Several LEDs on the front edge of CenTraCode II indicate board operation.

Each board, except for CSEX[1], has an LED that is normally on during operation to indicate that the +5 V logic power supply is distributing power to the board.

A serial port LED is turned on to indicate the reception of a message, and another LED indicates that a message is transmitting. If the LEDs do not periodically change state (turn on and off), the following conditions are possible:

- the communication channel has failed
- the state is the same for each message
- the port used by the NVIP module is not being addressed
- there is a wiring or connection problem to the remote module
- the remote module is turned off

Serial Ports 1 and 2 have a group of four LEDs controlled by the CenTraCode II System Software and the protocol emulation that are used to indicate reception and transmission of messages. The other serial ports’ LEDs are controlled by the CenTraCode II hardware and only indicate character input/output on a given port.

CSEX3/CSEX4/NVSP boards include a 2-character diagnostic display whose function is controlled by an on-board diagnostic switch. The function code shown on the diagnostic display must read “00” in order for the serial port LEDs to operate as described below. If the diagnostic display does not show “00”, the display and the LEDs are being used for a diagnostic function or the display is indicating the occurrence of a system error. A serial port LED is turned on to indicate the reception of a message, and another LED indicates message transmission. If the LEDs do not periodically change state (turn on and off), the communication channel has failed, the state is the same for each message, or this unit is not being addressed. A receive error LED is lit when an error is detected.

Table 5–1 summarizes the various communication LEDs controlled by CTC2 with DT8protocol. In the discussion of each board type, an illustration of the board edge including LED labels is included. For a further explanation of LEDs used for troubleshooting CenTraCode II systems, refer to the following Alstom publications:

- P2086B VPI® Vital Processor Interlocking Control System Operation and Maintenance Manual (Volume 1)
- P2511B VPI® II Vital Processor Interlocking II Control System Operation and Maintenance Manual (Volumes 3, 4, and 5)
- P2521B iVPI™ integrated Vital Processor Interlocking II Control System Operation and Maintenance Manual (Volumes 3, 4, and 5)

Table 5–1. Communication LED Functions CSEX[1], CSEX2 and CSEX3
Ports 1 and 2 for Standard DT8 Protocol

LED	Description
Normal	This is always dark and is not used by the DT8 protocol.
Receive Error	This should be dark. It illuminates when the protocol receives an unrecognized message format. The LED stays illuminated until an Address OK or valid address message is received.
Invalid Address Received	This LED illuminates when the protocol receives a valid message, but does not match any assigned to this Port. It remains illuminated until an Address OK or valid address message is received.
Transmitting Response	This LED flashes, illuminating steadily during the time the serial communication port transmitter is attempting to communicate with the remote unit.
Valid Address Received	This LED illuminates when a complete and valid message is received and the address matches ones assigned to this port. It remains illuminated until an Invalid Address is received or a Receiver Error is detected.

Table 5–2. Communication LED Functions CSEX[1], CSEX2 and CSEX3 Ports 3, 4, and 5 for Standard DT8 Protocol

LED	Description
Transmitting Response	This LED flashes, illuminating steadily during the time the serial communication port transmitter is attempting to communicate with the remote unit.
Receiving Response	This LED flashes, illuminating steadily during the time the serial communication port receiver is communicating with the remote unit.

Table 5–3. Communication LED Functions CSEX4 and NVSP Ports 1 and 2 for Standard DT8 Protocol

LED	Description
Transmitting Response	This LED flashes, illuminating steadily during the time the serial communication port transmitter is attempting to communicate with the remote unit.
Receiving Response	This LED flashes, illuminating steadily during the time the serial communication port receiver is communicating with the remote unit.

5.2.2 Data Format

It is important to know the differences between what appears on an oscilloscope and information presented by a protocol analyzer or in CenTraCode II MAC port diagnostic screens. Figure 5–1 shows that a swapping of positions occurs between the least significant and most significant bits when viewed by an analyzer or on a diagnostic screen. It illustrates the format of a serial character containing 1 start bit, 8 data bits, 1 parity bit and 1 stop bit. The number of data bits in a character is user-configurable to be 7 or 8, the parity bit can be disabled and the number of stop bits can be set to 1 or 2.

Each serial character's data bits are logically grouped into two hexadecimal digits (0-9, A-F). In the following figure, LSB and MSB refer to the Least Significant and Most Significant Bits of a hexadecimal digit.

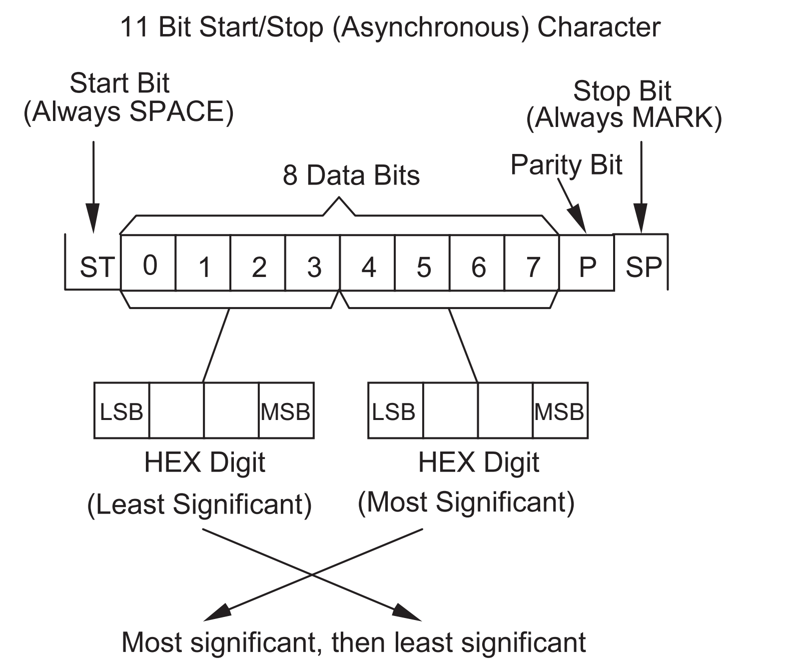


Figure 5–1. Transposition of Bytes in Transmitted Messages

5.2.3 CSEX Board

Refer to publication P2086B for details on the original CSEX Board hardware. Refer to the online help of the CAAPE software package for a detailed explanation of the generation and compilation of a non-vital application for the CSEX hardware platform.

- Table 5–4 describes the memory jumper placement for specific devices. CMOS versions of these memory devices are used where available, 120 ms or faster.
- Table 5–5 and Figure 5–2 show the serial port communication jumper settings. See Figure 5–3 and Figure 5–4 for the location of these jumpers, and of the on-board LEDs and other essential components.
- Table 5–6 describes the Watchdog jumper settings.

Table 5–4. CSEX Memory Jumpers

Device	Capacity	Type	Header	Jumpers
U36, U49	8K x 8	2764 EPROM	JU1	3-4, 7-8, 11-12
	16K x 8	27128 EPROM		1-2, 7-8, 11-12
	32K x 8	27256 EPROM		1-2, 5-6, 11-12
	64K x 8	27512 EPROM		1-2, 5-6, 9-10
U37, U50	8K x 8	2764 EPROM	JU2	3-4, 7-8, 11-12
	16K x 8	27128 EPROM		1-2, 7-8, 11-12
	32K x 8	27256 EPROM		1-2, 5-6, 11-12
	64K x 8	27512 EPROM		1-2, 5-6, 9-10

Table 5–5. CSEX Serial Port Communication Jumpers

Mode	JU3	JU4	JU5
RS-422	2-3, 4-5	No Connections	No Connections
RS-423/RS-232	1-2, 5-6	1-2, 3-4, 5-6, 7-8	1-2, 3-4, 5-6, 7-8

Table 5–6. CSEX Watchdog Jumper

Jumper	Watchdog Enabled	Watchdog Disabled
JU6	1-2	No Connections

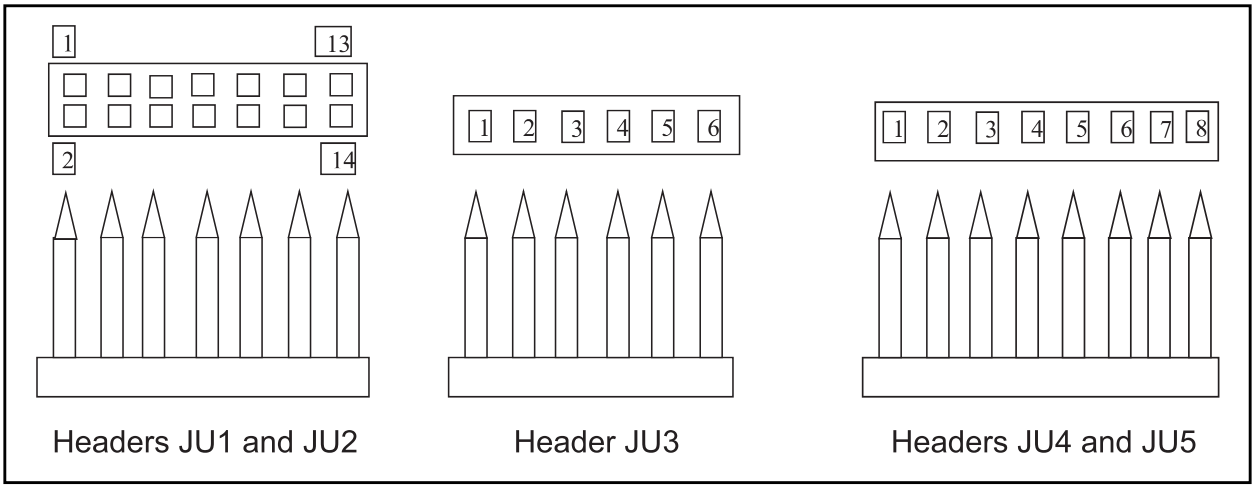


Figure 5–2. CSEX Memory Device and Serial Port Communication Selection Headers

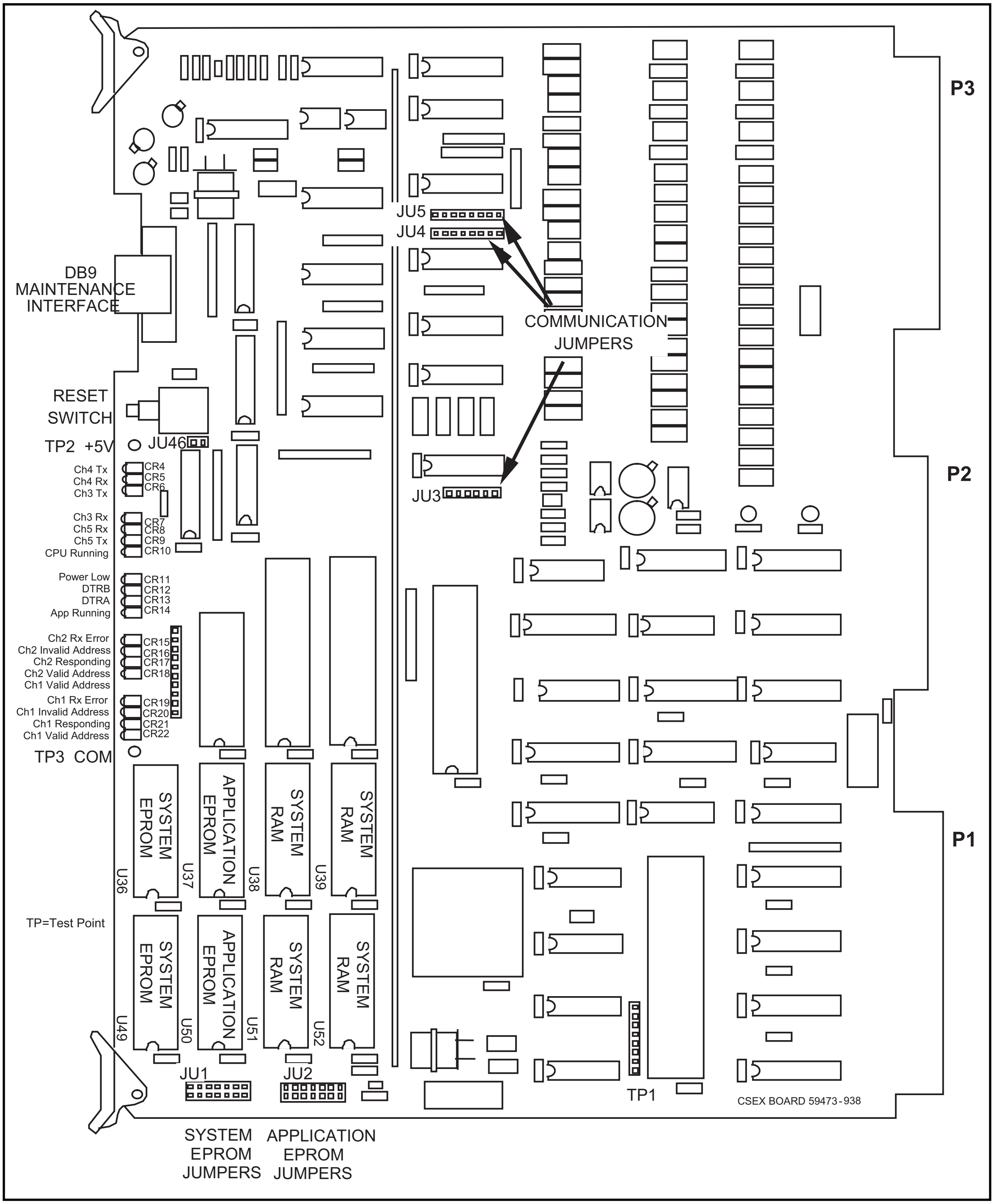


Figure 5–3. CSEX Board

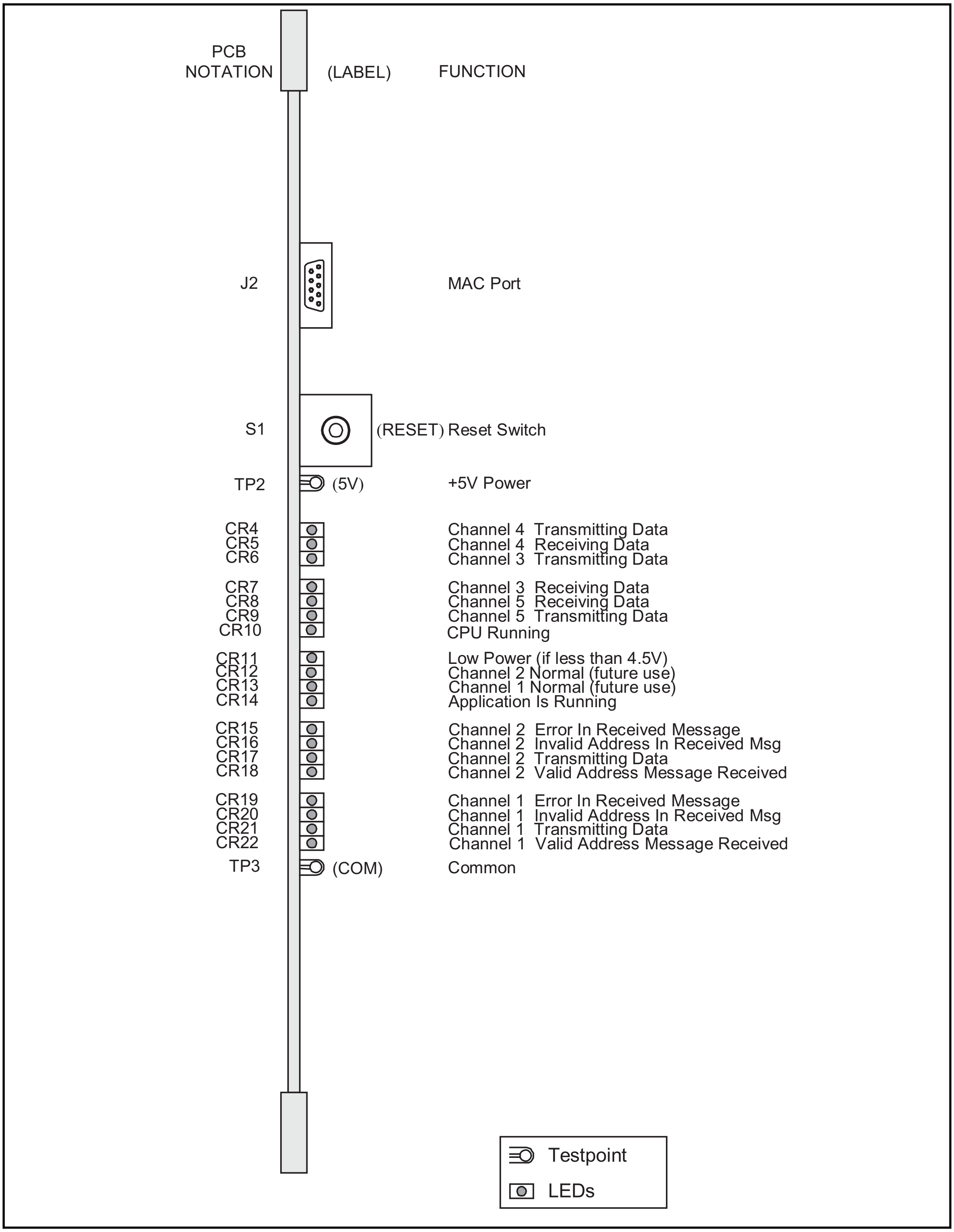


Figure 5–4. CSEX Board Edge

5.2.4 CSEX2 Board

Refer to Alstom publication P2086B for details on the CSEX2 Board P/N 31166-049-XX hardware. Refer to the online help of Alstom's CAAPE software package for a detailed explanation of the generation and compilation of a non-vital application for the CSEX2 hardware platform.

The CSEX2 MAC port in current loop mode can be connected to the P2 edge connector to provide a rear chassis connection. To do so, place jumpers TB2 and TB4 at pins 4-6 instead of as shown in Table 5–7.

Jumper TB5 is used to bring +5 Vcc to pin 1 of the MAC port connector, J2. This option is used to provide power to a Handheld Terminal (HHT).

Note: If an HHT is not used with the CSEX2 board, do not install Jumper TB5.

Table 5–7 defines the jumper settings for specific MAC port modes.

Table 5–8 and Table 5–9 define the jumper settings for Port 1 and Port 2 serial communications, and Table 5–10 shows how to set the jumpers for specific memory devices. CMOS versions of these memory devices are used where available, 150ns or faster. See Figure 5–5 and Figure 5–6 for the location of these jumpers, and of the on-board LEDs and other essential components.

Table 5–7. CSEX2 MAC Port Jumpers

MAC Port Mode	TB2	TB3	TB1	TB4	TB8
RS-232	2-4	1-2	1-2	2-4	1-2
Current Loop	3-5	2-3	2-3	3-5	2-3

Table 5–8. CSEX2 Serial Port 1 Communication Jumpers

Mode	TB6	TB24	TB25	TB26	TB27	TB28	TB29	TB30
RS-422	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2
RS-232	2-3	N/C	2-3	N/C	2-3	2-3	2-3	2-3

Table 5–9. CSEX2 Serial Port 2 Communication Jumpers

Mode	TB7	TB15	TB16	TB17	TB18	TB19	TB20	TB21	TB22	TB23
RS-422	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2
RS-232	2-3	N/C	2-3	N/C	3-4	N/C	3-4	N/C	3-4	3-4
DC	N/C	N/C	N/C	2-3	5-6	2-3	5-6	2-3	5-6	5-6

Place the CSEX2 battery board jumper at Position 1-2 prior to putting the board in service. Leave the jumper at Position 2-3 when storing the board.

Place the CSEX2 Watchdog Timer Enable jumper TB9 at Position 1-2 prior to putting the board into service.

Table 5–10. CSEX2 Memory Jumpers for Devices U36 and U37

Capacity	Type	TB10	TB11	TB12	TB13	TB14
32K x 8	27256 EPROM	1-2	1-2	2-3	2-3	2-3
64K x 8	27512 EPROM	1-2	2-3	2-3	2-3	1-2
128K x 8	27010 EPROM	N/C	2-3	2-3	1-2	1-2
256K x 8	27020 EPROM	2-3	2-3	1-2	1-2	1-2

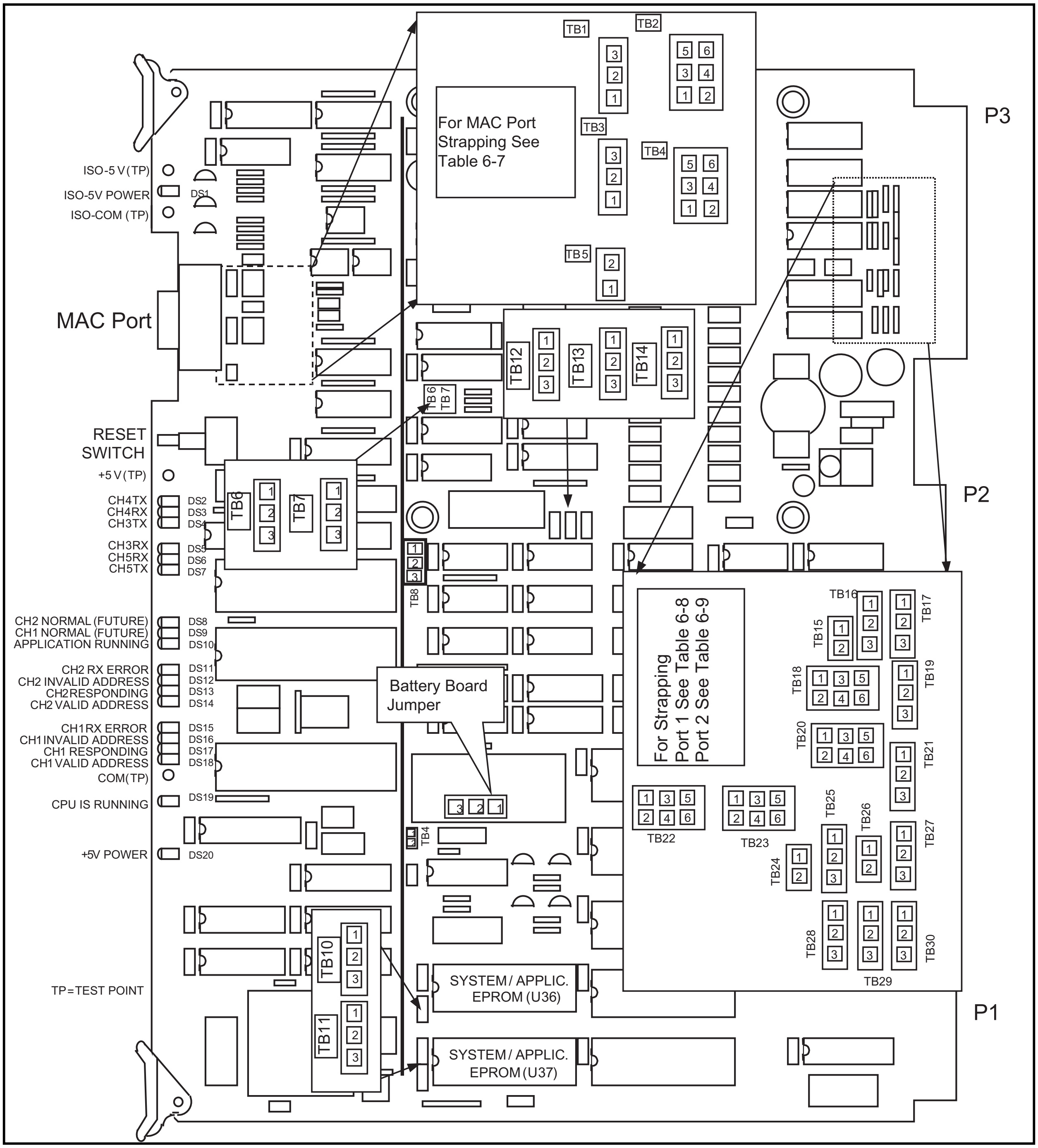


Figure 5–5. CSEX2 Board

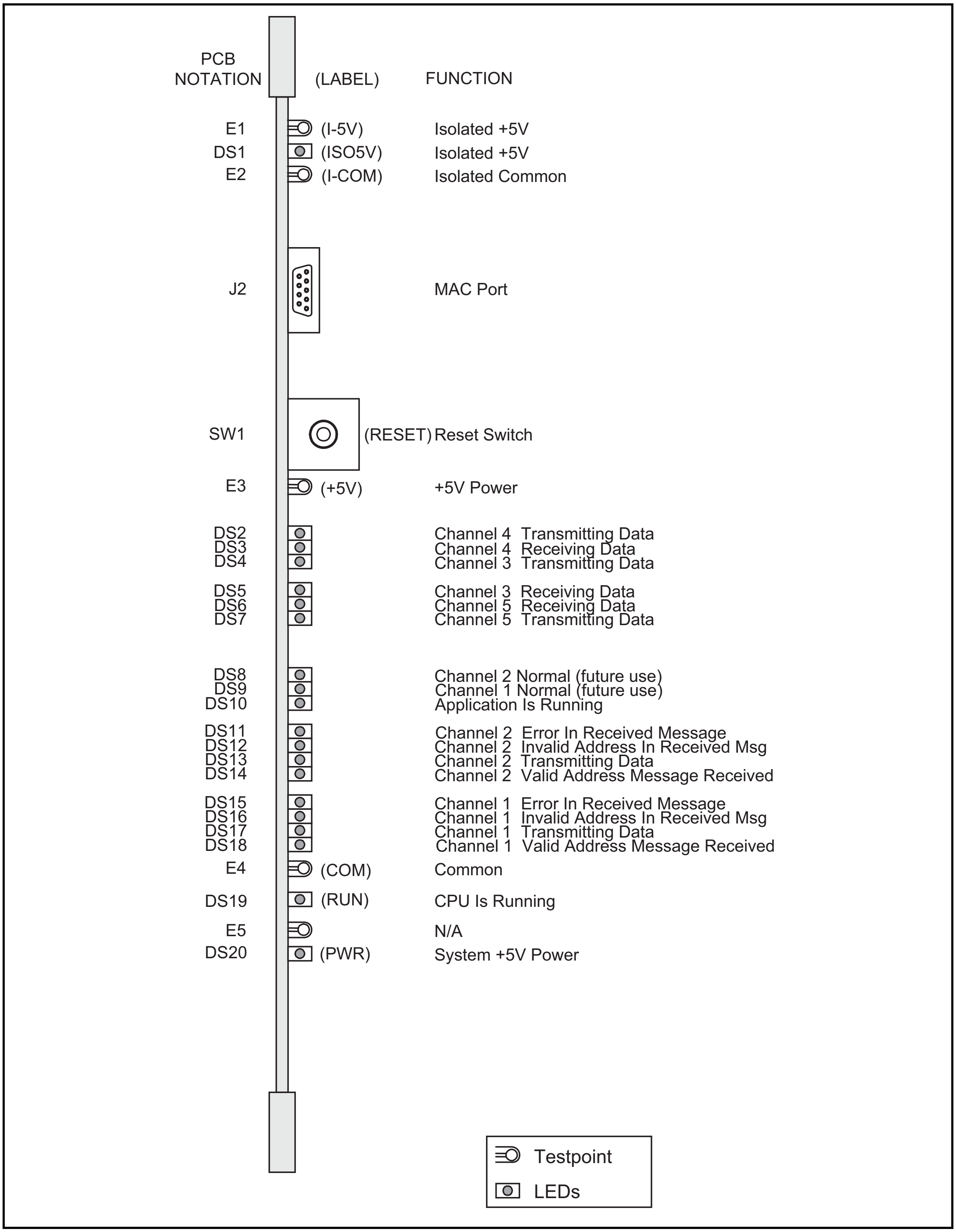


Figure 5–6. CSEX2 Board Edge

5.2.5 CSEX3 Board

Refer to Alstom publication P2086B for details on the CSEX3 Board P/N 31166-175-XX hardware. Refer to the online help of Alstom’s CAAPE software package for a detailed explanation of the generation and compilation of a non-vital application for the CSEX3 hardware platform.

Table 5–11 through Table 5–21 define the various jumpers and switches available on the CSEX3 board. Refer to Figure 5–7 and Figure 5–8 for the location of these jumpers and switches, and of the on-board LEDs and other essential components. CMOS versions of the Flash memory devices (29F040) are used where available, 70ns or faster.

Table 5–11. CSEX3 MAC Port Power Selection Jumper

JP1	Function
3-4	MAC port power disconnected (normal operation)
1-2	MAC port power connected (HHT use only)

Table 5–12. CSEX3 MAC Port Receive Data Source Selection Switch

SW1	Receive Data Source
“F”	Front of board (DB-9 connector J3)
“B”	Backplane connection

Table 5–13. CSEX3 Watchdog Selection Jumper

JP2	Function
3-5	Normal operation
1-3	Disable watchdog reset (emulator use only)

Table 5–14. CSEX3 Battery Selection Jumper

JP3	Function
6-8	Battery disconnected (storage)
7-8	Battery connected (normal operation)

Table 5–15. CSEX3 Application Switches

Serial Port	Switches
1	SW9, APP8 – APP1
2	SW10, APP16 – APP9

The operating mode for Serial Ports 1 and 2 is set in hardware using SW7 and SW4, respectively, as shown in Table 5–16 and Table 5–17. For Serial Port 2, Table 5–18 details additional switches that are required. In addition to these hardware switches, specify the operating mode in the non-vital application, as explained in Section 2.5.

Table 5–16. CSEX3 Serial Port 1 Communication Mode Switch

SW7	Mode of Operation
All Off	RS-422/485
All On	RS-232

Table 5–17. CSEX3 Serial Port 2 Communication Mode Switch

SW4	Mode of Operation
All Off	RS-422/485
All On	RS-232

Table 5–18. CSEX3 Serial Port 2 DC Code-Line Selection Switch

SW2, SW3, SW5, SW6	Mode of Operation
All positioned towards front of board	RS-232/422/485
All positioned towards back of board	DC Code-line

Table 5–19. CSEX3 Flash Enable Jumper

JP3	Mode of Operation
1-3	Flash Write Always Disabled
3-4	Flash Write Always Enabled
2-4	Flash Write Enabled by TP5 Only

Table 5–20. CSEX3 36-Pin P3 Auxiliary Board Communication Selection Switch SW1 Settings

Switch Position	Function
Towards P4 connector (DCD-B)	Normal operation
Towards P3 connector (RXCLK)	Provides external receive synchronous clock signal (e.g., for ARES protocol)

Table 5–21. CSEX3 36-Pin P3 Auxiliary Board CSEX vs. CSEX2/3 Switch SW2 Settings

Switch Position	Function
Towards P4 connector (CSEX2/3)	CSEX2/3 compatibility mode (normal operation)
Towards P3 connector (CSEX[1])	CSEX compatibility mode

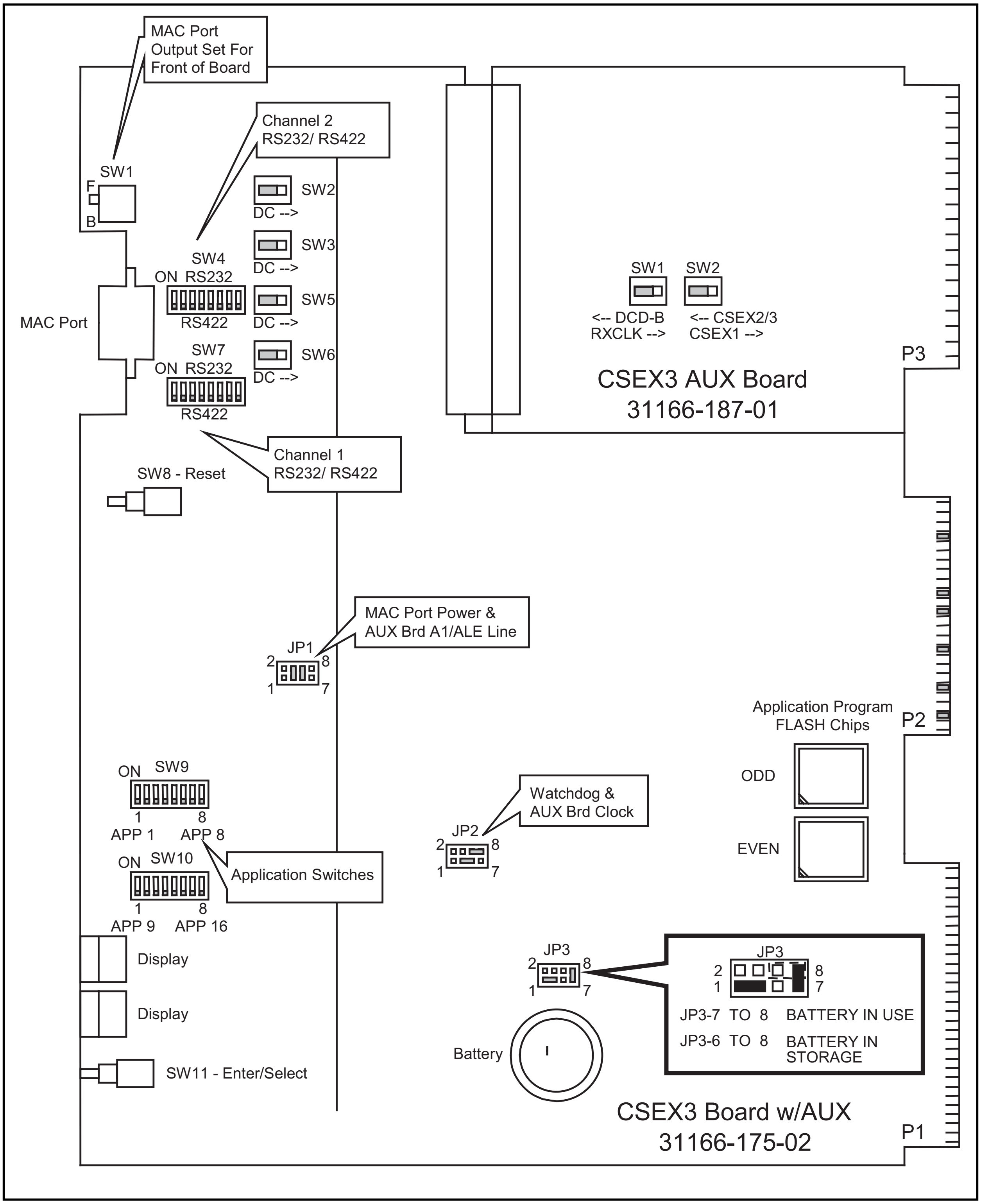


Figure 5–7. CSEX3 Board

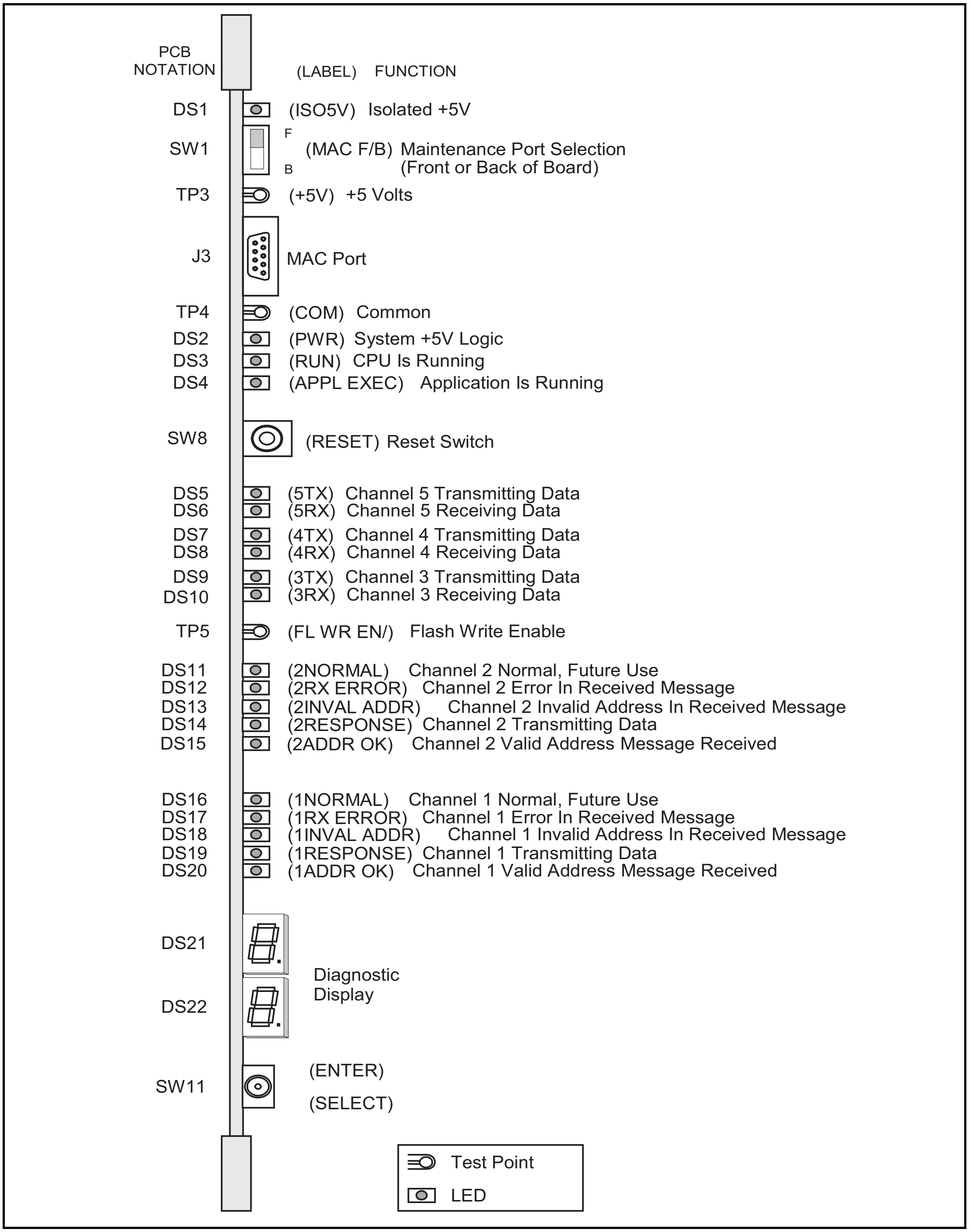


Figure 5–8. CSEX3 Board Edge

5.2.6 CSEX4 Board

Refer to Alstom publication P2511B for details on the CSEX3 Board P/N 31166-417-XX hardware. Refer to the online help of Alstom's CAAPE software package for a detailed explanation of the generation and compilation of a non-vital application for the CSEX4 hardware platform.

Table 5–22 through Table 5–28 show the jumper assignments for the CSEX4 Board. All possible functions have a jumper installed even though the jumper may not make an electrical connection. This is done to ensure that there is the correct number of jumpers (six) on the board at all times. See Figure 5–9 for the location of the CSEX4 Board jumpers.

Table 5–22. CSEX4 Board Communication Processor PROMJet Selection Jumper

TB	Function
TB1	Communication Processor PROMJet Header (no jumpers)
TB2 1-2	Communication Processor PROMJet connection points

Table 5–23. CSEX4 Board Communication Processor Flash Write Selection Jumper

TB3	Function
1-2	Communication Processor Flash PROM write enabled
2-3	Communication Processor Flash PROM write disabled

Table 5–24. CSEX4 Board Battery Selection Jumper

TB4	Function
1-2	Battery disconnected (use this position for shipping and storage, or if no battery is installed during operation)
2-3	Battery connected (do not use this position if no battery is installed)

Table 5–25. CSEX4 Board Main Processor Watchdog Selection Jumper

TB5	Function
1-2	Main Processor enable watchdog reset (normal operation) , jumper installed
2-3	Main Processor disable watchdog reset (for emulator use only)

Table 5–26. CSEX4 Board Main Processor PROMJet Selection Jumpers

TB	Function
TB6	Main Processor PROMJet Header (no jumpers)
TB7 1-2	Main Processor PROMJet connection points

Table 5–27. CSEX4 Board ASIC Selection Jumpers

TB	Function
TB8	ASIC Test Header (no jumpers)
TB9	ASIC Programming Header (no jumpers)

Table 5–28. CSEX4 Board Main Processor Write Enable Jumper

TB10	Function
1-2	Main Processor Flash Write always disabled
2-3	Main Processor Flash Write always enabled

Serial Ports 1 and 2 can receive and transmit the EIA232, EIA422 and EIA485 standards. Serial Port 2 can be alternately configured as a DC code line interface. Both ports can be configured independently, as shown in Table 5–29 and Table 5–30. Serial Ports 3 can receive and transmit asynchronous RS-422 and RS-485. SeeFigure 5–9 for the location of the CSEX4 switches.

-

Table 5–29. CSEX4 Board Channel 1 Communication Standard Selection Switch Setting

Standard	SW7 Position
EIA422/485	all off
EIA232	all on

Table 5–30. CSEX4 Board Channel 2 Communication Standard Selection Switch Settings

Standard	SW6 Position
EIA422/485	all off
EIA232	all on

Serial Port 4 receives and transmits the EIA232 standard, and connects diagnostic equipment to the CSEX4 Board. Switch SW1 determines whether the MAC ports RXD signal is accessible through the DB-9 connector or through P2 and P3 on the backplane (for a permanent diagnostic connection). A status LED provides the USB connection status. Table 5–32 describes the switch settings to select between front and back MAC port access. Table 5–31 describes the MAC RS-232 port connections, while Table 5–33 describes the MAC USB port connections.

Table 5–31. CSEX4 Board MAC (Maintenance Access) RS-232 Port Connector Pin Assignments

Pin	Function
J3-1	-
J3-2	RXD: receive data
J3-3	TXD: transmit data
J3-4	-
J3-5	ISOCOM: isolated common
J3-6	-
J3-7	-
J3-8	-
J3-9	

Table 5–32. CSEX3 Board MAC Port RXD Source Selection Switch Settings

SW1 Setting	Source
"F"	Front of board (DB-9 connector)
"B"	Backplane connection

Table 5–33. CSEX4 Board MAC (Maintenance Access) USB Port Connector Pin Assignments

Pin	Function
J4-1	VBUS
J4-2	D-
J4-3	D+
J4-4	COM
J4-5	Shield
J4-6	Shield

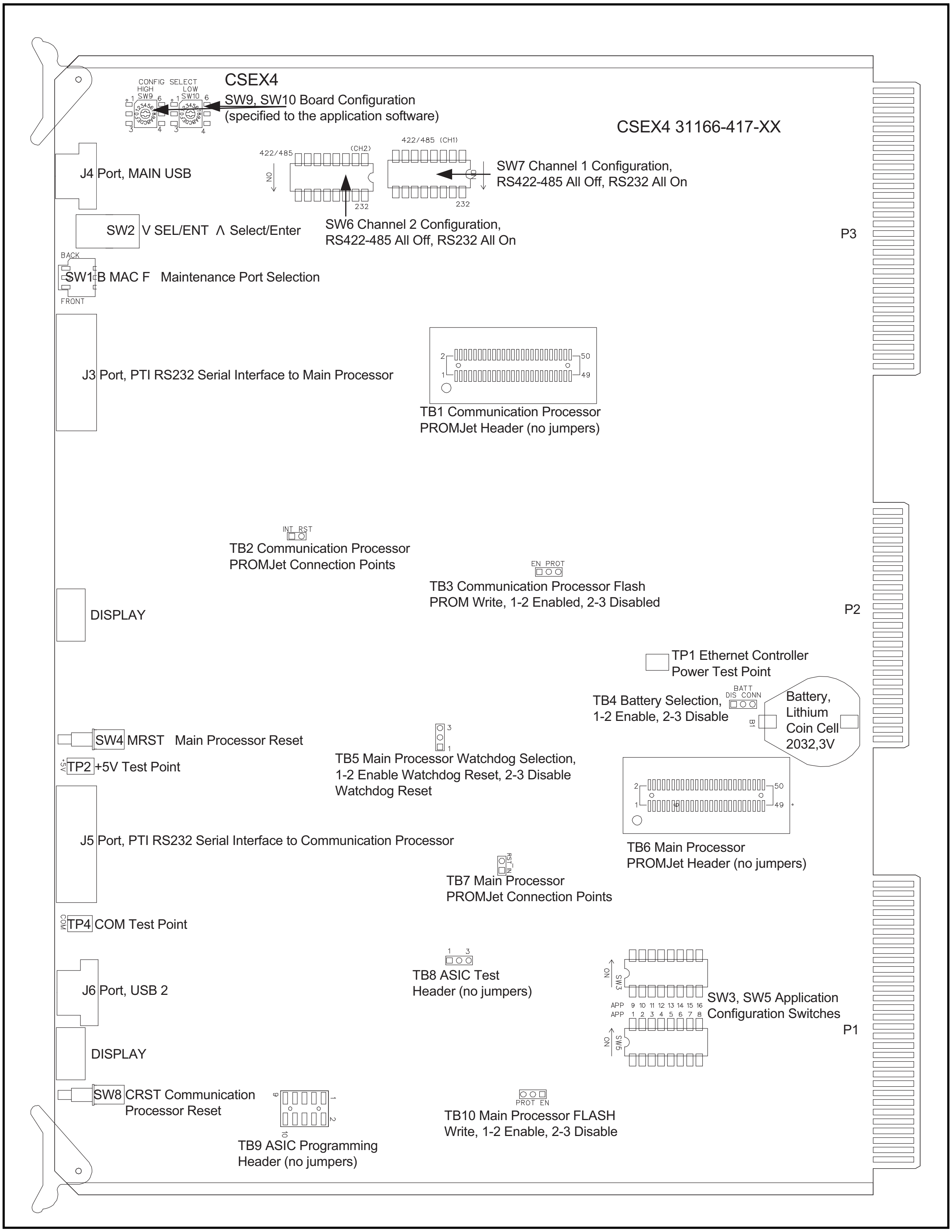


Figure 5–9. CSEX4 Board

CSEX4 BOARD
31166-417-XX

NORMAL INDICATION		PCB NOTATION	LABEL	FUNCTION
		J4	USB 1	(MAIN USB, Main Processor USB Port)
		SW2	V SEL/ENT Λ	(Display Select Button, Down for Select and Up for Enter, Center Position is OFF)
		SW1	B MAC (F / B)	(Maintenance Port Selection Front or Back of Board)
		J3	MAIN SERIAL	(PT1 RS232 Serial Interface to Main Processor)
				COM and Status LEDs
OFF, ON	OFF, ON	DS1	MTX MRX	(MAC Serial Port: Transmitting Data / Receiving Data)
OFF, ON	OFF, ON	DS2	CTX CRX	(Communication Processor: Serial Port Transmitting Data / Receiving Data)
ON	OFF	DS3	1OK 1INV	(Serial Port 1: Valid Addressed Message Received / Invalid [Unknown] Address in Received Message)
OFF	ON	DS4	1RSP 1ERR	(Serial Port 1: Error In Received Message / Transmitting Data)
OFF	OFF	DS5	2NRM 1NRM	(Not Used / Not Used [For Future Use])
ON	OFF	DS6	2OK 2INV	(Serial Port 2: Valid Addressed Message Received / Invalid [Unknown] Address in Received Message)
OFF	ON	DS7	2RSP 2ERR	(Serial Port 2: Error In Received Message / Transmitting Data)
		DS9	MAIN DSPY	(Main Processor 7-Segment Diagnostic Display)
				Network Status LEDs
OFF, ON	OFF, ON	DS15	CUSB MUSB	(Communication USB Port Active, Main USB Port Active)
OFF, ON	OFF, ON	DS8	3TX 3RX	(Serial Port 3: Transmitting Data / Receiving Data)
OFF, ON	OFF, ON	DS12	1LN 1LK	(LAN 1 Active / Link 1 Active)
OFF, ON	OFF, ON	DS13	2LN 2LK	(LAN 2 Active / Link 2 Active)
OFF	OFF, ON	DS10	MRST MCYC	(Main Processor Reset / Main Processor Running)
		SW4	MRST	(Reset 1, Main Processor Reset Button)
		TP2	+5V COMM	(+5 Volts Test Point)
		J5	COMM	(PT2 RS232 Serial Interface to Communication Processor)
		TP4	SERIAL COM	(Common Test Point)
OFF	OFF, ON	DS14	CRST CYCC	(Communication Processor Reset / Communication Processor Running)
		J6	COMM USB	(USB 2, Communication Processor USB Port)
		DS16	COMM DSPY	(Communication Processor 7-Segment Diagnostic Display)
		SW8	CRST	(Reset 2, Communication Processor Reset Button)

Figure 5–10. CSEX4 Board Edge

5.2.7 NVSP Board

Refer to Alstom publication P2521B for details on the NVSP Board P/N 31166-428-XX hardware. Refer to the on-line help of Alstom’s CAAPE software package for a detailed explanation of the generation and compilation of a non-vital application for the NVSP hardware platform.

Table 5–34 shows the jumper assignments for the NVSP Board. All possible functions have a jumper installed even though the jumper may not make an electrical connection. This is done to ensure that there is the correct number of jumpers the board at all times. See Figure 5–11 for a board layout drawing identifying the NVSP jumper locations.

Table 5–34. NVSP Board Jumpers

JP1	Function
TB1 (no jumper)	Communication Processor PROMJet Header
TB2 (no jumper)	Communication Processor PROMJet connection points
TB3-1 to TB3-2	Communication Processor Flash PROM write enabled
TB3-2 to TB3-3	Communication Processor Flash PROM write disabled
TB4-1 to TB4-2	Backup battery disconnected (use this position for shipping & storage, or if no battery is installed during operation)
TB4-2 to TB4-3	Backup battery connected
TB5-1 to TB5-2	NVSP Processor watchdog- normal operation
TB5-2 to TB5-3	NVSP Processor watchdog- disable watchdog reset (for emulator use only)
TB6 (no jumper)	NVSP Processor PROMJet Header
TB7 (no jumper)	NVSP Processor PROMJet connection points
TB8 (no jumper)	ASIC Test Header
TB9 (no jumper)	ASIC Programming Header
TB10-1 to TB10-2	NVSP Processor Flash PROM write enabled
TB10-2 to TB10-3	NVSP Processor Flash PROM write disabled

A port is assigned RS422/485 or RS232.

Table 5–35. NVSP Board Channel 1 Communication Standard Selection Switch Settings

Standard	SW7 Position
RS422/485	all off
RS232	all on

Table 5–36. NVSP Board Channel 2 Communication Standard Selection Switch Settings

Standard	SW6 Position
RS422/485	all off
RS232	all on

Serial Port 4 (the MAC port) uses an EAI-232 transceiver to transmit and receive. An alternative connection for Port 4 is via the USB connection J6.

Table 5–37. NVSP Board MAC EIA-232 Port Description

Pin	Function
J3-1	–
J3-2	RXD: receive data
J3-3	TXD: transmit data
J3-4	–
J3-5	COM: isolated common
J3-6	–
J3-7	–
J3-8	–
J3-9	–

Table 5–38. NVSP Board MAC USB Port Description

Pin	Function
J4-1	VBUS
J4-2	D-
J4-3	D+
J4-4	COM
J4-5	Shield
J4-6	Shield

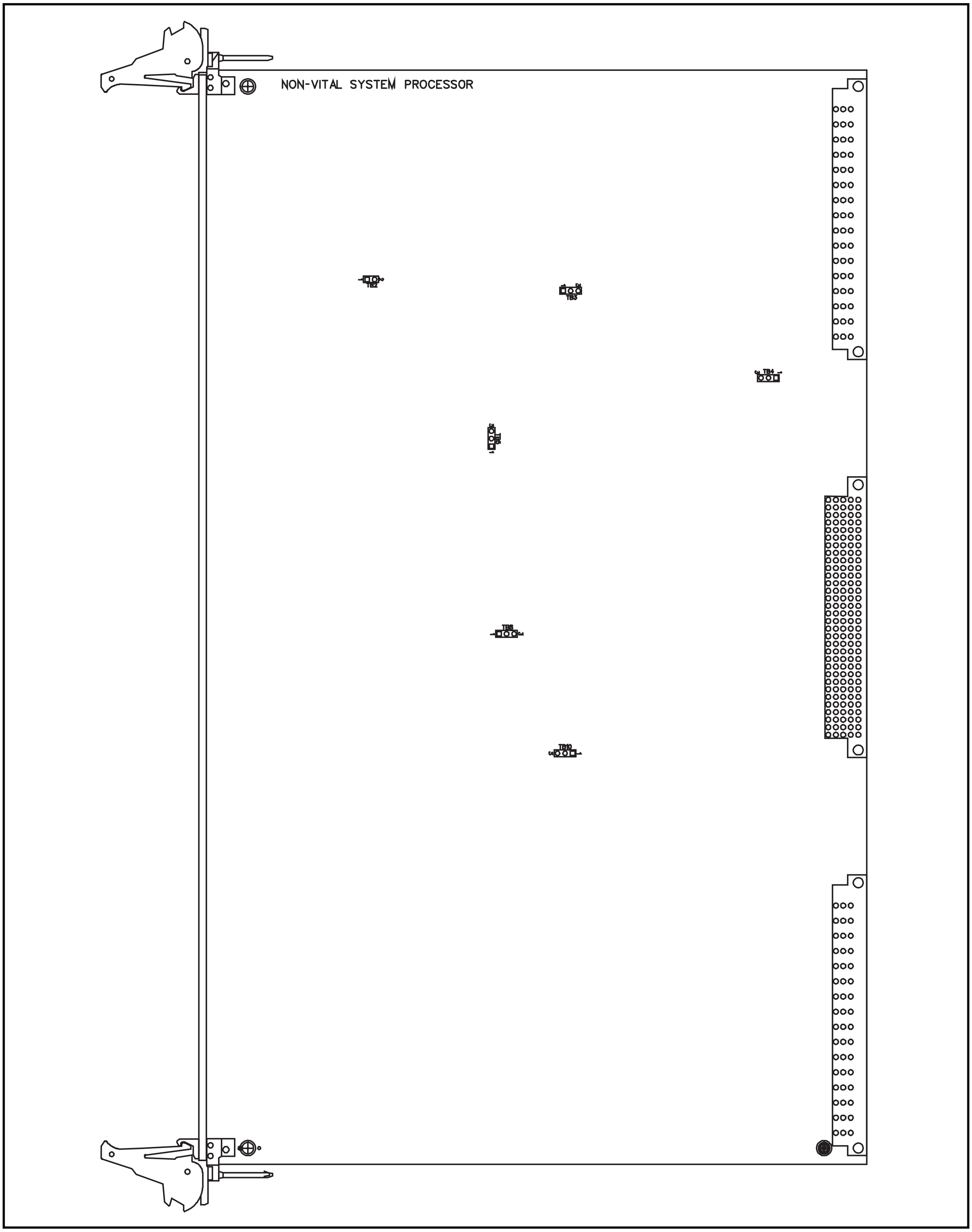


Figure 5–11. NVSP Board

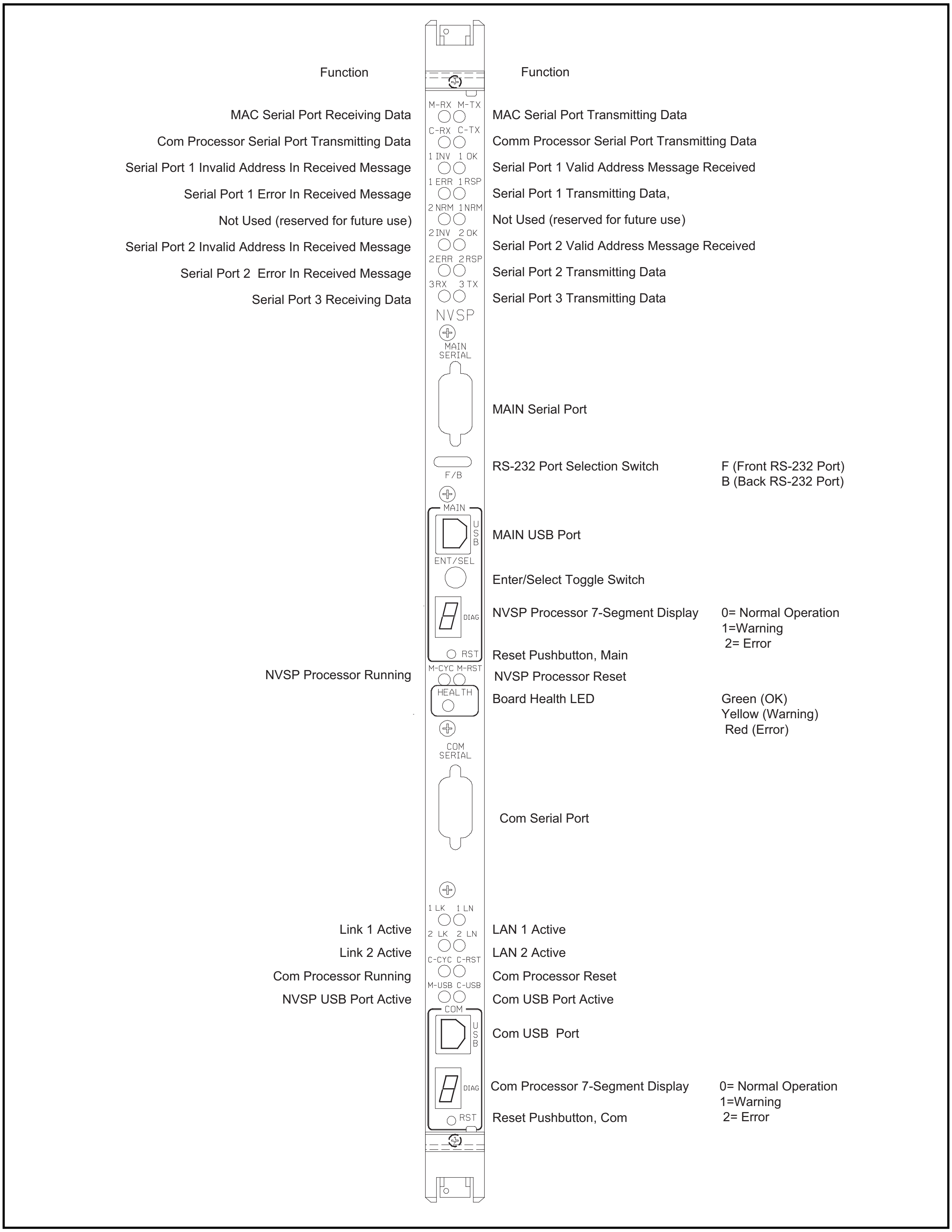


Figure 5–12. NVSP Board Edge

5.3 TROUBLESHOOTING GUIDE

Installation requirements for the DT8 protocol on the CenTraCode II system depend primarily upon the specific hardware being used on the code-line. The code-line is usually interfaced with either Serial Port 1 or 2, since other CenTraCode II ports do not support the modem control lines RTS and CTS. Along the same lines, the RTS and CTS signals can be required for handshaking with the modem when operating in Slave Mode. In the event of problems with the code-line, verify all wiring to the code unit according to the location's wiring diagrams.

It can be useful to connect a protocol analyzer (for example, the Hewlett-Packard HP4957A) to the digital interface of the Data Communications Equipment (DCE) and obtain a recording of control and indication messages. Examine the recording to determine the integrity of the messages, and to insure that the appropriate communication parameters are being used (for example Quiet Poll Interval and Hold RTS Timer).

Table 5–39 is a general troubleshooting guide for the CenTraCode II systems.

Table 5–39. CenTraCode II Troubleshooting Guide

Observation	Possible Cause
All LEDs are off.	Improper +5V (±0.25V) system power.
Status of LEDs does not change.	Code-line interface or code-line problem. Communication port interface problem.
A serial port's "Valid Address" LED does not light.	Improper CAA application configuration – the station address is not correct.
The "Application Is Running" LED does not light: • CSEX: CR14 • CSEX2: DS10 • CSEX3: DS4 • CSEX4: DS10 (Main Processor) • NVSP: • M-CYC (Main /NVSP Processor)	This LED should be on to indicate that the application logic (the site-specific program) is running. If this LED does not stay on, the non-vital application code is not operational. Verify proper programming of the EPROMs or FLASH.

Table 5–39. CenTraCode II Troubleshooting Guide *(Cont.)*

Observation	Possible Cause
The "CPU Is Running" LED flashes on and off: • CSEX: CR10 • CSEX2: DS19 • CSEX3: DS3 • CSEX4: DS14 (Communication Processor) • NVSP: • C-CYC (Communication Processor)	This LED should remain on to indicate that the embedded System Software is running correctly. If this LED does not stay on, the operating software is not running correctly. Verify proper programming of the EPROMs or FLASH.
The receiver does not respond to a request from the remote communication unit.	RTS/CTS is not connected or is not functioning as programmed.
Controls or displays do not function as expected.	CAA application logic programming error or incorrect wiring.
The Maintenance Access (MAC) port does not operate properly.	• Check the error code shown on the 2-digit diagnostic display (if applicable). See the troubleshooting section of the appropriate system manual (P2086B for VPI, P2511 for VPI II, P2521B for iVPI). • Check • the wiring of the MAC port cable. • Check the MAC port for communication activity. • Simultaneously press • Ctrl+E at the VT100 terminal to view the Main Menu. • Check the baud rate of the VT100 terminal (typically is 9600 baud).

Review the list of CenTraCode system errors that can be displayed at the MAC port diagnostic menu by the self-checking mechanism built into the CenTraCode System Software.

APPENDIX A – CENTRACODE® II SAMPLE CAAPE INPUT FILE

A.1 GENERAL

This section contains an input file to illustrate the inclusion of DT8 in a CenTraCode® II non-vital application.

This appendix contains two samples:

- A sample Serial Communications Section, typically found in a non-vital application's CSS file, to illustrate the use of CAAPE records for the inclusion of the DT8 protocol in an application
- A sample Network Serial Communications Section, typically found in a non-vital application's NSS file, to illustrate the use of CAAPE records for the inclusion of a NVSoE port running the DT8 protocol in an application

A.2 CSEX, CSEX2, CSEX3 SAMPLE INPUT FILE

```
**********************************************************************
*                                                                    *
* FILE: SAMPLE.CSS                                                   *
* COMMUNICATIONS INPUT FILE                                          *
* DataTrain VIII EMULATION                                           *
*                                                                    *
**********************************************************************

SERIAL COMMUNICATIONS SECTION

SERIAL PORT 1 = TYPE (DT8 PEER), UNLATCHED CONTROLS

CONFIGURATION FILE = PEER_DT8.LPC
DEFAULT BAUD RATE = 9600
DATA FORMAT = 8,1,N

DESTINATION = CTCID
SOURCE = MAIN
* NOTE: FOR THE FOLLOWING MESSAGE TO BE COMPILED, THE DESTINATION NAME
* MUST MATCH THE BOARD ID NAME SPECIFIED IN THE APPLICATION’S CSI FILE.

MESSAGE = ADDRESS (00000001), LENGTH (16)

    1 = WORD1BIT1
    2 = WORD1BIT2
    3 = WORD1BIT3
    4 = WORD1BIT4
    5 = WORD1BIT5
    6 = WORD1BIT6
    7 = WORD1BIT7
    8 = WORD1BIT8
    9 = WORD2BIT1
   10 = WORD2BIT2
   11 = WORD2BIT3
   12 = WORD2BIT4
   13 = WORD2BIT5
   14 = WORD2BIT6
   15 = WORD2BIT7
   16 = WORD2BIT8

DESTINATION = MAIN
SOURCE = CTCID
* NOTE: FOR THE FOLLOWING MESSAGE TO BE COMPILED, THE SOURCE NAME
* MUST MATCH THE BOARD ID NAME SPECIFIED IN THE APPLICATION’S CSI FILE.
```

```
MESSAGE = ADDRESS (00000001), LENGTH (24)

    1 = WORD1BIT1K
    2 = WORD1BIT2K
    3 = WORD1BIT3K
    4 = WORD1BIT4K
    5 = WORD1BIT5K
    6 = WORD1BIT6K
    7 = WORD1BIT7K
    8 = WORD1BIT8K
    9 = WORD2BIT1K
   10 = WORD2BIT2K
   11 = WORD2BIT3K
   12 = WORD2BIT4K
   13 = WORD2BIT5K
   14 = WORD2BIT6K
   15 = WORD2BIT7K
   16 = WORD2BIT8K
   17 = WORD3BIT1K
   18 = WORD3BIT2K
   19 = WORD3BIT3K
   20 = WORD3BIT4K
   21 = WORD3BIT5K
   22 = WORD3BIT6K
   23 = WORD3BIT7K
   24 = WORD3BIT8K

SPECIAL CONTROL = LENGTH (24)
* USED TO PASS FLAGS BETWEEN THE PROTOCOL AND THE NON-VITAL APPLICATION
* NOTE: FLAGS SET BY THE PROTOCOL MUST BE CLEARED BY NON-VITAL LOGIC STATEMENT

    1 = BIT_MAP_IN
    2 = CHANGE_IN
    3 = ACK_IN
    4 = POLL_IN
    5 = MAP_REQ_IN
    6 = BIT_MAP_OUT
    7 = CHANGE_OUT
    8 = ACK_OUT
    9 = POLL_OUT
   10 = MAP_REQ_OUT
   11 = STATION_ALIVE
   12 = POLL_ENABLE
   13 = TEXT_IN
   14 = TEXT_OUT
   15 = SEND_MAP
   16 = PERMZERO
   17 = SEND_TIME_REQ
```

```
18 = TIME_REQ_IN
19 = SEND_TIME
20 = TIME_IN
21 = PERMZERO
22 = PERMZERO
23 = PERMZERO
24 = PERMZERO
```

A.3 CSEX4/NVSP SAMPLE CAAPE INPUT FILE

```
**********************************************************************
*                                                                    *
* FILE: SAMPLE.NSS                                                   *
* COMMUNICATIONS INPUT FILE                                          *
* DataTrain VIII EMULATION                                           *
*                                                                    *
**********************************************************************
NETWORK SERIAL COMMUNICATIONS SECTION
NETWORK PORT 1 = TYPE (DT8 PEER), UNLATCHED CONTROLS
CONFIGURATION FILE = PEER_DT8.LPC
DESTINATION = CTCID:1
SOURCE = MAIN:3
* NOTE: FOR THE FOLLOWING MESSAGE TO BE COMPILED, THE DESTINATION NAME
* MUST MATCH THE BOARD ID NAME SPECIFIED IN THE APPLICATION’S CSI FILE.
MESSAGE = ADDRESS (00000001), LENGTH (16)
    1 = WORD1BIT1
    2 = WORD1BIT2
    3 = WORD1BIT3
    4 = WORD1BIT4
    5 = WORD1BIT5
    6 = WORD1BIT6
    7 = WORD1BIT7
    8 = WORD1BIT8
    9 = WORD2BIT1
   10 = WORD2BIT2
   11 = WORD2BIT3
   12 = WORD2BIT4
   13 = WORD2BIT5
   14 = WORD2BIT6
   15 = WORD2BIT7
   16 = WORD2BIT8

DESTINATION = MAIN:3
SOURCE = CTCID:1
* NOTE: FOR THE FOLLOWING MESSAGE TO BE COMPILED, THE SOURCE NAME
* MUST MATCH THE BOARD ID NAME SPECIFIED IN THE APPLICATION’S CSI FILE.
MESSAGE = ADDRESS (00000001), LENGTH (24)
    1 = WORD1BIT1K
    2 = WORD1BIT2K
    3 = WORD1BIT3K
    4 = WORD1BIT4K
    5 = WORD1BIT5K
    6 = WORD1BIT6K
    7 = WORD1BIT7K
```

```
    8 = WORD1BIT8K
    9 = WORD2BIT1K
   10 = WORD2BIT2K
   11 = WORD2BIT3K
   12 = WORD2BIT4K
   13 = WORD2BIT5K
   14 = WORD2BIT6K
   15 = WORD2BIT7K
   16 = WORD2BIT8K
   17 = WORD3BIT1K
   18 = WORD3BIT2K
   19 = WORD3BIT3K
   20 = WORD3BIT4K
   21 = WORD3BIT5K
   22 = WORD3BIT6K
   23 = WORD3BIT7K
   24 = WORD3BIT8K

SPECIAL CONTROL = LENGTH (24)
* USED TO PASS FLAGS BETWEEN THE PROTOCOL AND THE NON-VITAL APPLICATION
* NOTE: FLAGS SET BY THE PROTOCOL MUST BE CLEARED BY NON-VITAL LOGIC STATEMENT
    1 = BIT_MAP_IN
    2 = CHANGE_IN
    3 = ACK_IN
    4 = POLL_IN
    5 = MAP_REQ_IN
    6 = BIT_MAP_OUT
    7 = CHANGE_OUT
    8 = ACK_OUT
    9 = POLL_OUT
   10 = MAP_REQ_OUT
   11 = STATION_ALIVE
   12 = POLL_ENABLE
   13 = TEXT_IN
   14 = TEXT_OUT
   15 = SEND_MAP
   16 = PERMZERO
   17 = SEND_TIME_REQ
   18 = TIME_REQ_IN
   19 = SEND_TIME
   20 = TIME_IN
   21 = PERMZERO
   22 = PERMZERO
   23 = PERMZERO
   24 = PERMZERO
```

APPENDIX B – Non-Vital Ethernet Network Communication Background And Setup

B.1 NVSOE NETWORKING

The CSEX4 non-vital processor board in VPI and the NVSP non-vital processor board in iVPI are capable of communicating non-vital binary data with other devices. Several physical interfaces (RS-232, RS-422, RS-485, and Ethernet) and a number of communication protocols (including Alstom DataTrain VIII – DT8, Ansaldo Genisys, and Modicon Modbus®, among others) are supported. In particular, each CSEX4 or NVSP board can support a number of non-vital serial over Ethernet (NVSoE) connections. Each NVSoE connection can be configured to use one of several protocols (e.g., DT8, Genisys, or Modbus). However, regardless of the selected protocol, NVSoE communication employs a TCP/IP client/server arrangement. An overview of TCP/IP client/server networking is presented below, followed by the details of setting up NVSoE communication in CAAPE.

B.1.1 TCP/IP Client/Server Networking and NVSoE

A given non-vital processor board permits a number of virtual point-to-point NVSoE connections to be defined. At present, up to 10 NVSoE connections, represented by NVSoE nodes numbered 1 through 10, are permitted for each CSEX4 or NVSP board in a system.

An NVSoE connection is a TCP/IP client/server connection: one end of the connection is set up as a TCP/IP server, and the other end is set up as a client. A TCP/IP server provides services to one or more TCP/IP clients. The client initiates a transaction by communicating with the server. Consequently, the client must know how to get in touch with the server, but the server does not generally know how to contact a particular client prior to receiving a message from the client. In TCP/IP terms, the client contacts the server at a particular IP address using a particular TCP/IP port number[1] that is associated with the service that the client wishes to access on the server. In its message to the server, the client provides a TCP/IP port number to which the server can provide a reply. In practice, a TCP/IP port number is assigned to the server and made known to the client, but the client automatically chooses an unused port number and uses it to establish a connection with the server; the client's port number is not predefined and is referred to as an *ephemeral port number*. Some port numbers are well known and registered with a governing body[2]: for example, a web browser (an HTTP client) communicates with an HTTP server at port 80, and an email client communicates with an SMTP server at port 25. Some ranges of unregistered port numbers may be employed for private use (for NVSoE communication, for example) or on an ad-hoc basis.

1. The TCP/IP port number is not to be confused with the CSEX4/NVSP RS-232, RS-422, RS485 hardware ports or the virtual NVSoE nodes 1 through 10.
2. Internet Assigned Numbers Authority – IANA (http://www.iana.org/).

This understanding of how a TCP/IP client/server connection works is useful in understanding how to set up an NVSoE connection. The NVSoE setup will be described in the context of CAAPE, the VPI/iVPI application development tool.

A number of items are required to set up the network settings for an NVSoE node on a CSEX4/NVSP board:

1. the IP addresses associated with the local Ethernet devices on the CSEX4/NVSP board, and whether or not redundancy is enabled;
2. which physical Ethernet devices on the CSEX4/NVSP board will be used for the NVSoE node: enet1, enet2, or both enet1 and enet2 if redundancy is required;
3. whether the NVSoE node is being set up as the client end or the server end of the TCP/IP connection;
   – if the NVSoE node is being set up as a TCP/IP server, a port number for the server is also required (if it is a client, a specific port number is explicitly disallowed – see the above discussion of TCP/IP client/server operation);
4. the IP address (or addresses, if redundancy is enabled) associated with the remote end of the connection for the NVSoE node;
   – if the NVSoE node is being set up as a TCP/IP client, thus implying that the remote end of the connection is a server, then a port number is also required for the remote server connection.

The IP addresses for the local Ethernet devices on a CSEX4/NVSP board are the same for all NVSoE nodes running on the board. Other configuration items are set independently for each NVSoE node: which of the board’s Ethernet devices is used, whether the NVSoE node is client or a server, and remote IP addresses are all set independently.

Following is a discussion of each of these NVSoE setup items including where the information is entered in CAAPE and what options are available.

B.1.2 Local IP Address(es)

On the Network tab of the CSEX4/NVSP board window in CAAPE (Figure B–2), either or both of the two available Ethernet devices can be enabled via Enable checkboxes for each device. If an Ethernet device is enabled via its checkbox, an IP address and subnet mask is entered by clicking the Properties… button associated with each Ethernet device. This establishes the local IP addresses associated with this particular board. The addresses chosen for a particular board must be unique within their subnets. Note that these local IP addresses apply to all NVSoE nodes running on this CSEX4/NVSP board.

If redundancy is desired to be available for NVSoE connections, the **Redundancy** checkbox must be checked. This allows both Ethernet devices to be used to provide a redundant NVSoE connection with a remote device: a redundant NVSoE connection sends the same message over both Ethernet connections, with the receiving end discarding all duplicate messages. This way, if one of the two Ethernet networks is disrupted, NVSoE communication will continue on the other Ethernet network. With the **Redundancy** checkbox checked, each NVSoE node can be set up to use or not use redundancy independently of the other NVSoE nodes.

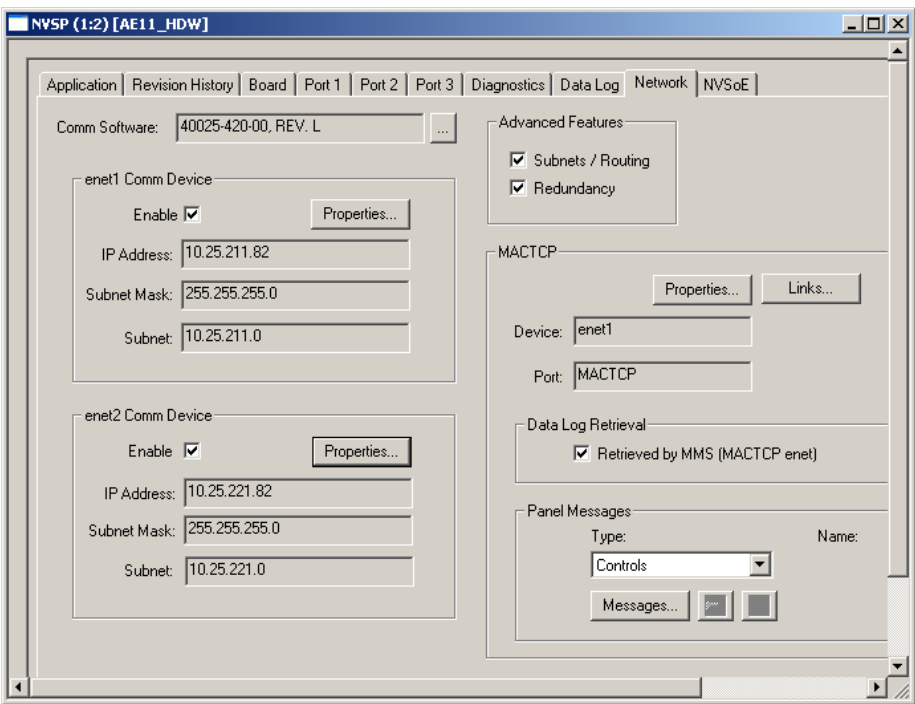


Figure B–1. CSEX4/NVSP Network Tab

B.1.3 NVSoE Setup

The NVSoE tab of the CSEX4/NVSP board window in CAAPE (Figure B–2) permits up to 10 NVSoE connections to be configured. In particular, the NVSoE tab permits selecting the Ethernet device(s) for each enabled NVSoE connection and permits selecting whether the local end of the connection is a client or a server.

On the NVSoE tab, the **NVSoE:** pulldown (Figure B–2) allows an NVSoE node, numbered 1 through 10, to be selected for configuration. After selecting a node number for configuration, the **Properties:** button is clicked to bring up the NVSoE Properties dialog (Figure B–3) that permits setting up the networking properties for the node.

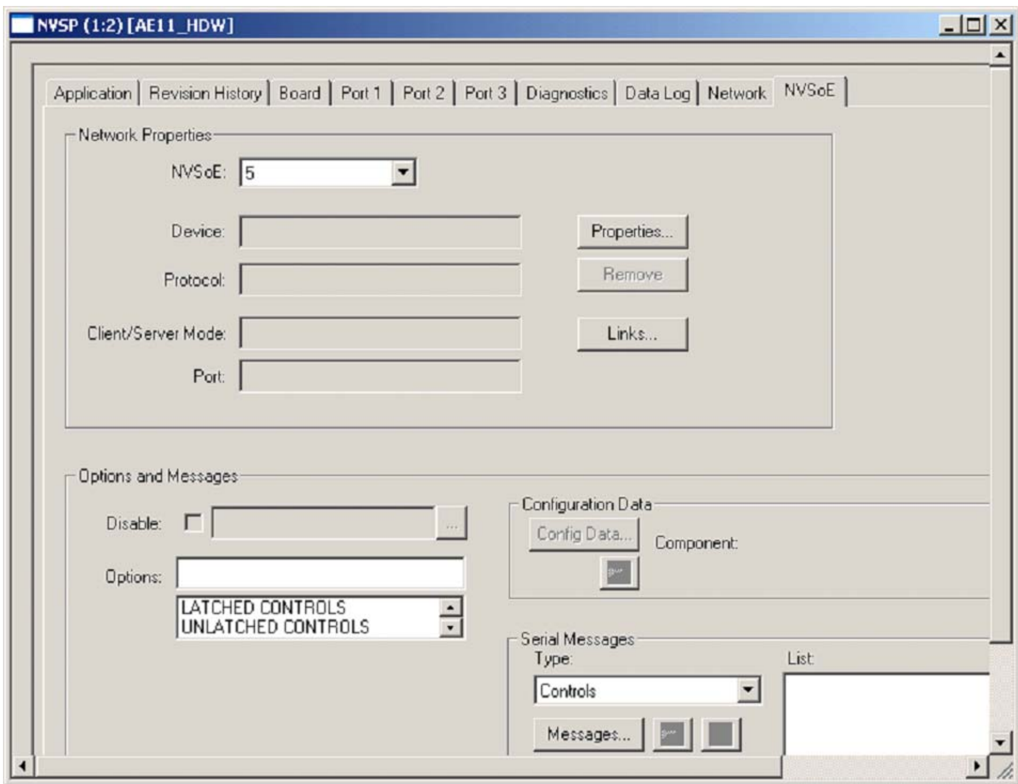


Figure B–2. CSEX4/NVSP NVSoE Tab

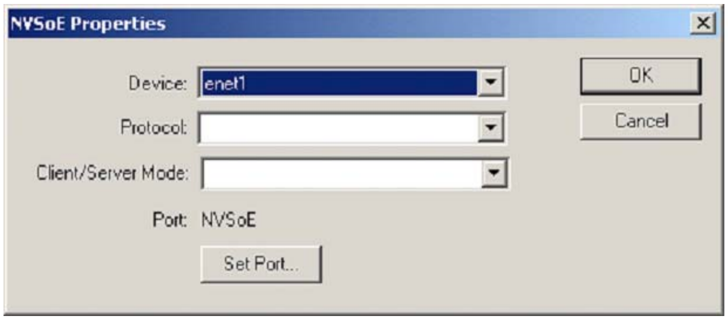


Figure B–3. NVSoE Properties Dialog

In the NVSoE Properties dialog (Figure B–3), the **Device:** pulldown permits selection of enet1, enet2, or Redundant, depending on how the local Ethernet devices have been configured on the Network tab (this corresponds to setup item 2 above). The **Protocol:** pulldown permits the selection of a protocol to use for this NVSoE node: DT8 Sync or Genisys, for example. The **Client/Server Mode:** pulldown permits the local end of this particular NVSoE connection to be a TCP/IP client or a server (this corresponds to setup item 3 above). If Server is selected from the **Client/Server Mode:** pulldown, then a TCP/IP port number is set up by clicking on the **Set Port…** button to bring up the Network Port dialog (Figure B–4) (this corresponds to the bulleted item following setup item 3).

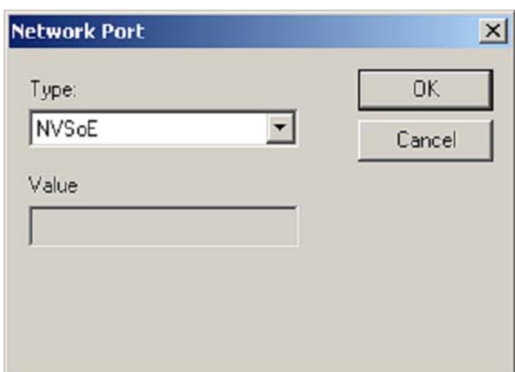


Figure B–4. Network Port Dialog

In the Network Port dialog (Figure B–4), the **Type:** pulldown permits selection of NVSoE, NVSoE Legacy, or Manual port configuration. If NVSoE is chosen, then the network port number for the local port is chosen from a predefined list by the application compiler (the CAA); if Manual is chosen, then the port number can be manually entered. NVSoE Legacy is similar to NVSoE in determining how port numbers are selected, but is not recommended for use in new applications[3].

3. NVSoE Legacy is no longer used and may be removed from future versions of CAAPE.

If Manual is chosen, it is recommended to use a high port number to avoid existing registered port numbers: something in the 50000s would be okay (this is the range that is used for predefined ports if NVSoE is chosen as the **Type:** in the Network Port dialog).

Note that if the NVSoE node is being set up as a client, then it is *required* that the Network Port **Type:** be set to NVSoE. This permits an ephemeral port to be selected for the client during operation as required by the client/server protocol. If a manual port number is entered for an NVSoE client node, *this will prevent the NVSoE connection from being established*.

B.1.4 Links… Setup

On the NVSoE tab (Figure B–2), the **Links:** button is clicked to bring up the Links dialog. The Network tab on the Links dialog (Figure B–5) permits viewing and setting up the IP addresses and ports (if required) for the remote device associated with each NVSoE node (this corresponds to setup item 4 above). When an NVSoE node has been defined by setting its properties as described above, then a line is added to the list of links shown on the Network tab in the Links dialog with missing information for the remote end of the connection. The Name column identifies the system, board, and NVSoE node to which the particular row applies. Right-clicking on a row permits the remote information to be added (**Add…**), edited (**Edit…**), or removed (**Remove**). **Add…**, **Edit…**, or double-clicking on a row brings up the Remote Connection Data dialog (Figure B–6) to permit entering data for the remote end of the connection.

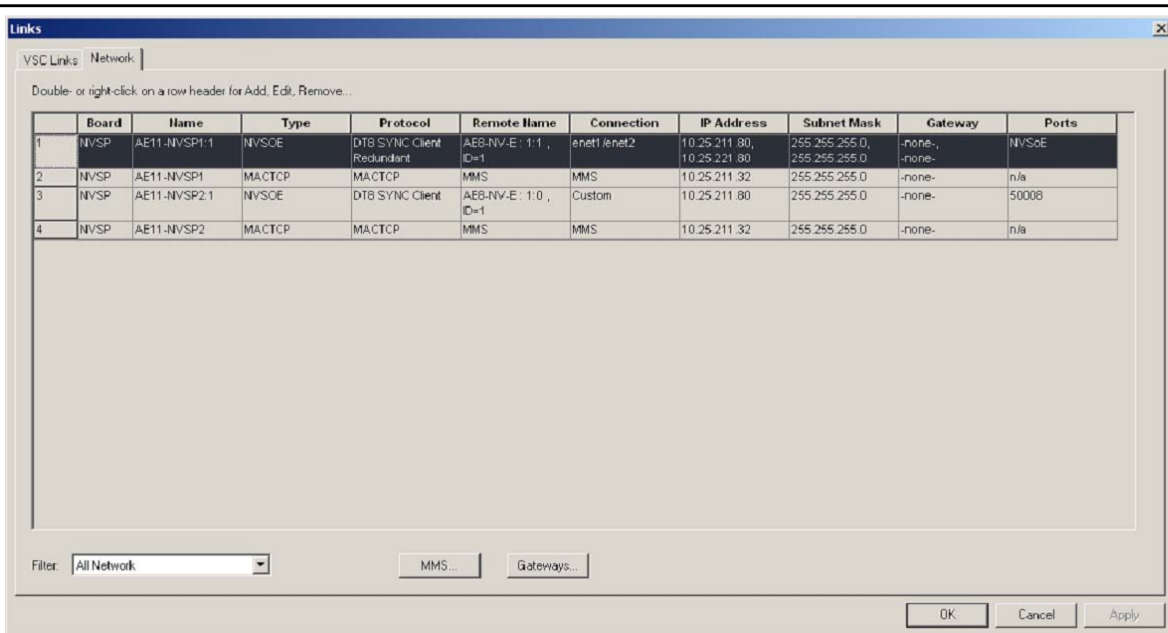


Figure B–5. Links Dialog, Network Tab

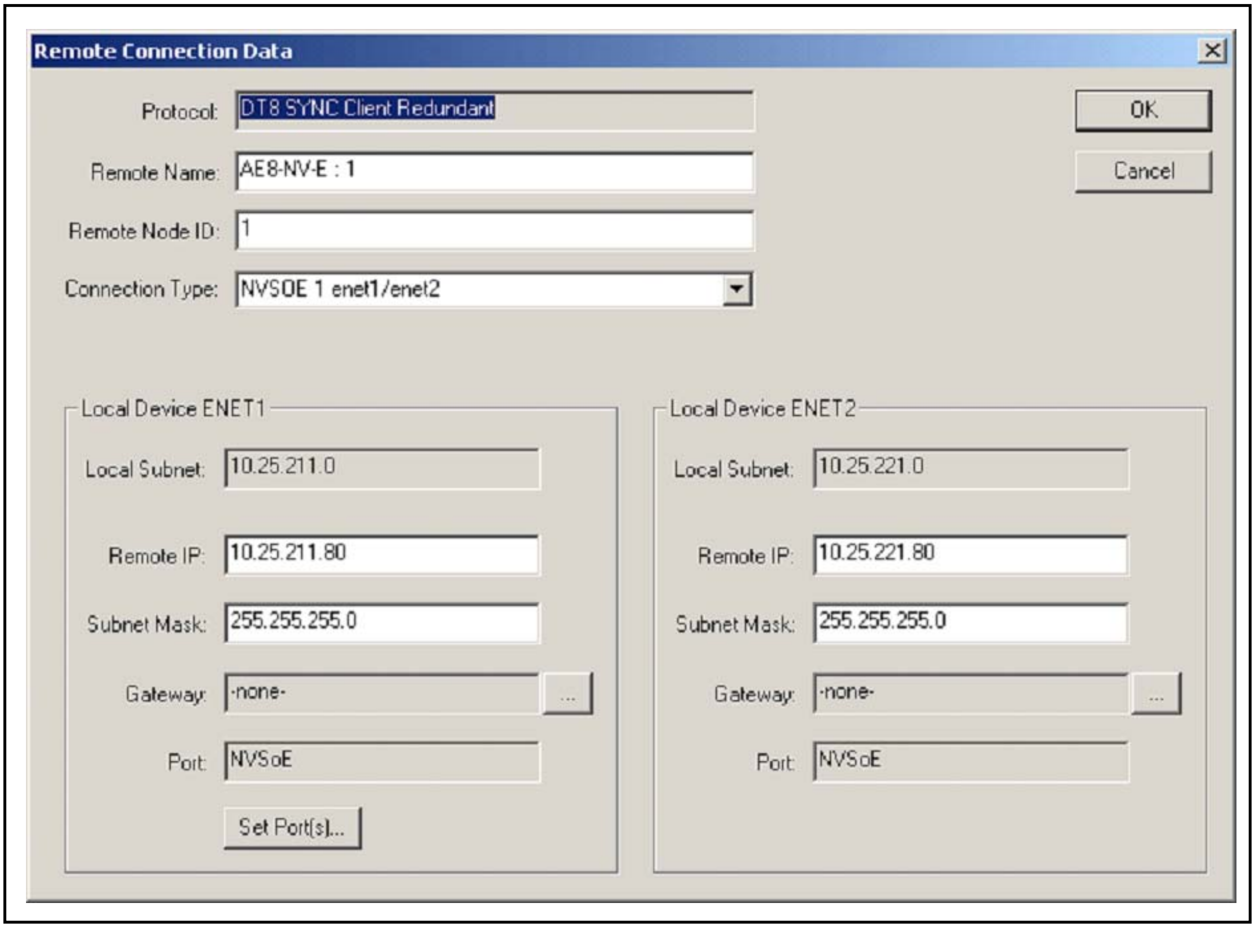


Figure B–6. Remote Connection Data Dialog

In the Remote Connection Data dialog (Figure B–6), the **Remote Name:** box is provided for entering a descriptive name for the remote node. The **Remote Node ID:** is used for certain situations in which multiple remote connections are being made to a particular local NVSoE node; this can be left at 1 if there is only a single remote device associated with a local NVSoE node. The **Connection Type:** relates to setting the remote TCP/IP port number; this will be explained in more detail later. The button (**…**) to the right of the **Gateway:** field permits a network gateway device to be identified for this connection; if the remote end of the connection is not in the same subnet as the local end of the connection (i.e., the subnet set in the Network tab – Figure B–1), then the IP address of the gateway that connects the local subnet to the larger network must be identified by using this button and the associated dialog (the name of the selected gateway is shown in the **Gateway:** field).

Depending on the local Ethernet devices that a particular NVSoE node is configured to use (i.e., enet1, enet2, or Redundant in the **Device:** setting in the NVSoE tab in Figure B–2), the Remote Connection Data dialog (Figure B–6) will provide places to enter corresponding IP settings for the remote devices that the selected local Ethernet devices will be connected to. For example, if NVSoE node 5 is set up to run over enet2, then the Remote Connection Data dialog provides a place to enter the **Remote IP:**, **Subnet Mask:**, and **Port:** of the remote device that **Local Device ENET2** will be communicating with. In the Remote Connection Data dialog shown in Figure B–6, the selected NVSoE node has redundancy enabled, so remote IP addresses, masks, gateways, and ports can be entered for both **Local Device ENET1** and **Local Device ENET2**.

The **Set Port(s)…** button brings up the Network Port dialog (Figure B–4) that permits setting the port numbers for the remote device that this NVSoE node will communicate with. If the local NVSoE node for this link is set up as a TCP/IP server, then this button is disabled: the server determines the port number of the NVSoE client when it receives a message from the client. If the local NVSoE node for this link is set up as a client, then this button and the associated dialog box allow the port number of the remote server to be entered (this corresponds to the bulleted item following setup item 4). In the Network Port dialog, if **Type:** is set to NVSoE, then a predefined port number for the remote server will be used based on the **Connection Type:** setting on the Remote Connection Data dialog (Figure B–6). In this case, it is required to know the NVSoE node number (1 through 10) of the server on the remote unit, as well as the Ethernet device on the remote CSEX4/NVSP (enet1, enet2, or enet1/enet2 if redundancy is enabled) used by the remote NVSoE node; it is also required that the remote server have its port set to **Type:** NVSoE so that the correct predetermined ports are used on both ends of the connection. If **Connection Type:** is set to Custom, then it is required to set **Type:** Manual in the Network Port dialog, and manually enter the port(s) associated with the remote server; additionally, **Type:** Manual appears to override the **Connection Type:** setting even if it is not set to Custom.

B.2 TCP/IP CLIENT/SERVER AND PROTOCOL

When an NVSoE connection is arranged between two CSEX4 or NVSP boards, CAAPE is used to configure one of the boards as a TCP/IP server and the other as a TCP/IP client. When the systems containing the two boards are powered up, the board configured as a client will immediately attempt to connect to the server *even if there is no request from the higher-level protocol* (e.g., DT8, Genisys, or Modbus). In this way a TCP/IP client/server connection is established that is independent of protocol. Consequently, the protocol that employs this connection does know (or need to know) whether it is using the TCP/IP client or server end of the connection: from the higher-level protocol point-of-view, the TCP/IP connection simply appears to be a high-speed serial connection. Because of this independence between the higher-level protocol (DT8, Genisys, Modbus) and the TCP/IP client/server connection, either end of the TCP/IP client/server connection can be used for master or slave (or client or server) of the higher-level protocol. For example, a DT8 Master could be on the TCP/IP server end of an NVSoE connection with DT8 Slave on the TCP/IP client end of the connection, even though it might be expected that DT8 Master should be a TCP/IP client because it initiates a data transaction.

If a TCP/IP connection is being established between a CSEX4/NVSP board and equipment from another vendor, the independence between protocol and TCP/IP client/server no longer applies. In this case, the initiating end of the connection should be configured as a TCP/IP client and the responding end of the connection should be configured as a TCP/IP server. For example, when configuring an Ethernet connection between CSEX4/NVSP and another vendor's device using Modbus TCP protocol, then the end of the connection that is configured to use Modbus TCP Client must also be set up as a TCP/IP client, and the end of the connection that is configured to use Modbus TCP Server must also be set up as a TCP/IP server.

The **Protocol:** dropdown on the NVSoE Properties dialog (Figure B–3) provides the following choices:

- DT8
- DT8 Code
- DT8 Code Master
- DT8 Code Slave
- DT8 Master
- DT8 Peer
- DT8 Slave
- DT8 Sync
- Genisys
- Modbus Slave
- Modbus TCP Client
- Modbus TCP Server

Details for each of these protocols are provided by the user manuals for the protocols:

- P2346E, DataTrain VIII Code System for the Alstom CenTraCode II (CTC2) System User's Manual (*this manual*)
- P2346F, CenTraCode II-s and II-v Communication Systems Emulation Genisys Code System
- P2346AA, Modbus TCP Server Protocol Emulation for the Alstom CenTraCode II (CTC2) System User's Manual

INDEX

INDEX

Need help?
Contact Customer Service:

Alstom Signaling Inc.
1025 John Street
West Henrietta, NY 14586
USA

1-800-717-4477
www.alstomsignalingsolutions.com