Designation: E 1381 – 95

An American National Standard

AMERICAN SOCIETY FOR TESTING AND MATERIALS
100 Barr Harbor Dr., West Conshohocken, PA 19428
Reprinted from the Annual Book of ASTM Standards. Copyright ASTM

Standard Specification for Low-Level Protocol to Transfer Messages Between Clinical Laboratory Instruments and Computer Systems[1]

This standard is issued under the fixed designation E 1381; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 This specification describes the electronic transmission of digital information between clinical laboratory instruments and computer systems. The clinical laboratory instruments under consideration are those that measure one or more parameters from one or more patient samples. Often they will be automated instruments that measure many parameters from many patient samples. The computer systems considered here are those that are configured to accept instrument results for further processing, storage, reporting, or manipulation. This instrument output may include patient results, quality control results, and other related information. Typically, the computer system will be a Laboratory Information Management System (LIMS).

1.2 The terminology of the Organization for International Standards (ISO) Reference Model for Open Systems Interconnection (OSI) is generally followed in describing the communications protocol and services. The electrical and mechanical connection between instrument and computer is described in the Physical Layer section. The methods for establishing communication, error detection, error recovery, and sending and receiving of messages are described in the Data Link Layer section. The data link layer interacts with higher layers in terms of sends and receives "messages," handles data link connection and release requests, and reports the data link status.

1.3 Specification E 1394 is concerned with message content in the interface between clinical instruments and computer systems. The major topics are found in the following sections:

2. Referenced Documents

2.1 *ASTM Standards:*

E 1394 Specification for Transferring Information Between Clinical Instruments and Computer Systems[2]

2.2 *ANSI Standards:*[3]

X3.4-1986 American National Standard Code for Information Systems—Coded Character Sets—7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)

X3.15-1976 American National Standard for Bit Sequencing of the American National Standard Code for Information Interchange in Serial-by-Bit Data Transmission

X3.16-1976 American National Standard Character Structure and Character Parity Sense for Serial-by-Bit Data Communication in the American National Standard Code for Information Interchange

2.3 *ISO Standard:*[3]

International Standard ISO 7498-1984(E), Information Processing Systems—Open Systems Interconnection—Basic Reference Model, International Organization for Standardization

2.4 *Other Document:*[4]

EIA-232-D-1986 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange

3. Terminology

3.1 *receiver*—the device that responds to the sender and accepts the message.

---

[1] This specification is under the jurisdiction of ASTM Committee E-31 on Healthcare Informatics and is the direct responsibility of Subcommittee E31.14 on Clinical Laboratory Instrument Interface.

Current edition approved Oct. 10, 1995. Published January 1996. Originally published as E 1381 – 91. Last previous edition E 1381 – 91.

[2] *Annual Book of ASTM Standards*, Vol 14.01.

[3] Available from American National Standards Institute, 11 W. 42nd St., 13th Floor, New York, NY 10036.

[4] Available from Electronics Industries Association, 2001 I Street, N.W., Washington, DC 20006.

3.2 *sender*—the device that has a message to send and initiates the transmission process.

3.3 The parts of a communication between instrument and computer are identified by the following terms. The parts are hierarchical and are listed in order of most encompassing first.

3.3.1 *session*—a total unit of communication activity, used in this standard to indicate the events starting with the establishment phase and ending with the termination phase, as described in subsequent sections.

3.3.2 *message*—a collection of related information on a single topic, used here to mean all the identity, tests, and comments sent at one time. When used with Specification E 1394, this term means a record as defined by Specification E 1394.

3.3.3 *frame*—a subdivision of a message, used to allow periodic communication housekeeping such as error checks and acknowledgements.

4. Significance and Use

4.1 Nearly all recent major clinical instruments have provision for connection to a computer system, and in nearly all laboratories that have implemented a LIMS, there is a need to connect the laboratory's high volume automated instruments to the LIMS so that results can be transferred automatically. To accomplish this connection, both the instrument and the computer must have compatible circuits and appropriate software, and there must be a proper cable to connect the two systems.

4.1.1 Without this standard specification, the interface between each different instrument and each different computer system is likely to be a different product. This increases the cost, the chances for compatibility problems, and the difficulty of specifying and designing a proper system. In addition, interfaces for every instrument-computer combination may not be available, forcing expensive and time-consuming custom development projects.

4.2 This standard specification defines the electrical parameters, cabling, data codes, transmission protocol, and error recovery for the information that passes between the instrument and the laboratory computer. It is expected that future products from instrument manufacturers and computer system developers, released after the publication of this specification, will conform to this specification, and that will lead to plug-together compatibility of clinical instruments and computer systems.

5. Physical Layer

5.1 *Overview*—The mechanical and electrical connection for serial binary data bit transmission between instrument and computer system is described in the physical layer. The topology is point-to-point, a direct connection between two devices.

5.2 *Electrical Characteristics*—The voltage and impedance levels for the generator and receiver circuits are as specified in the EIA-232-D-1986 standard.

5.2.1 *Signal Levels*:

5.2.1.1 For the data interchange circuits, a marking condition corresponds to a voltage more negative than minus three volts with respect to signal ground at the interface point. A spacing condition corresponds to a voltage more positive than plus three volts with respect to signal ground at the interface point.

5.2.1.2 Binary state ONE (1) corresponds to the marking condition; binary state ZERO (0) corresponds to the spacing condition.

5.2.1.3 The signal levels conform to the EIA-232-D-1986 standard.

5.2.2 *Character Structure*:

5.2.2.1 The method of data transmission is serial-by-bit start/stop. The order of the bits in a character is:

(1) One start bit, corresponding to a binary 0,

(2) The data bits of the character, least significant bit transmitted first,

(3) Parity bit,

(4) Stop bit(s), corresponding to a binary 1.

5.2.2.2 The time between the stop bit of one character and the start bit of the next character may be of any duration. The data interchange circuit is in the marking condition between characters.

5.2.2.3 Even parity corresponds to a parity bit chosen in such a way that there are an even number of ONE bits in the sequence of data bits and parity bit. Odd parity corresponds to an odd number of ONE bits when formed in the same way.

5.2.2.4 All devices must be capable of sending and receiving characters consisting of one start bit, eight data bits, no parity bit, and one stop bit.

5.2.2.5 The default character structure consists of one start bit, eight data bits, no parity bit, and one stop bit. Eight data bit character sets are allowed but not specified by this standard. Other character structures can be used for specialized applications, for example, seven data bits, odd, even, mark or space parity, or two stop bits.

5.2.2.6 The character bit sequencing, structure, and parity sense definitions conform to ANSI standards X3.15-1976 and X3.16-1976.

5.2.3 *Speed*:

5.2.3.1 The data transmission rate for instruments shall be at least one of these baud rates: 1200, 2400, 4800, or 9600 baud. The preferred rate is 9600 baud and should be the default setting of the instrument when more than one baud rate is available. The computer system must have the capability for all four baud rates.

5.2.3.2 Devices may optionally have the capability for other baud rates such as 300, 19 200, and 38 400 baud for use in specialized applications.

5.2.4 *Interface Connections*:

5.2.4.1 The conforming connection specified here defines the point of interconnection between the domain of the instrument and domain of the computer system. (See Fig. 1 and Fig. 2.) Within the domain of either device, any appropriate connection system may be used, preferably with suitable cable locking hardware.

5.2.4.2 The conforming connection utilizes a 25-position connector. The connector contact assignments are listed in Table 1. Connector contacts not listed are unused. The connector contact assignments conform to the EIA-232-D-1986 standard for the circuits that are used.

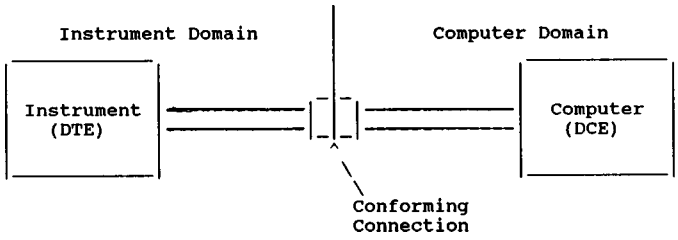


FIG. 1 Connector Strategy for Instrument Computer Connection—Cable Mounted

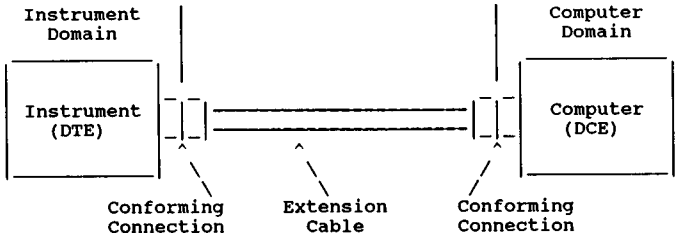


FIG. 2 Connector Strategy for Instrument Computer Connections—Chassis Mounted

TABLE 1 Connector Contact Assignments

Contact No.	EIA Circuit	Description	Direction	
			Instrument	Computer
1	...	Shield	...	No Connection
2	BA	Transmitted Data	Output	Input
3	BB	Received Data	Input	Output
7	AB	Signal Ground	...	...

5.2.4.3 Contact 1 is the shield connection, it connects to the instrument's (the DTE) frame. The shield connection is left open at the computer (the DCE) to avoid ground loops. There will be no connections on any other pins. All other pins will be open circuits.

5.3 *Mechanical Characteristics*:

5.3.1 *Connector*:

5.3.1.1 The conforming connector associated with the instrument is a commercial type DB-25P (subminiature D male) style connector. The conforming connector associated with the computer is a commercial type DB-25S (subminiature D female) style connector. The connector dimensions must correspond to those given in the EIA-232-D-1986 standard.

5.3.1.2 When the conforming connector of the instrument is cable mounted, it shall be configured with a locking device such as No. 4-40 or M-3 thread female screw locking hardware. When the conforming connector of the computer is cable mounted, it shall be configured with a locking device such as No. 4-40 or M-3 thread male screw locking hardware. (See Fig. 1.)

5.3.1.3 When the conforming connector of either device is chassis mounted, it shall be configured with devices such as No. 4-40 or M-3 thread female screw locking hardware. The mating cable connector shall use devices such as No. 4-40 or M-3 thread male screw locking hardware. (See Fig. 2.)

5.3.1.4 When the conforming connector of the instrument is cable mounted and the conforming connector of the computer is chassis mounted, then a change in the cable mounted locking hardware is necessary.

5.3.2 *Cable*—Any extension cables to connect the instrument to the computer require a female connector on one end to mate with the instrument and a male connector on the other end to mate with the computer. Detailed requirements of an interconnecting cable are undefined but good engineering practice should be followed in selecting the cable and connectors. Shielded cable and connectors may be necessary to suppress electromagnetic interface (EMI). Low capacitance cable may be necessary for long cable lengths or the higher data rates. Appropriate connector locking hardware should be used at the conforming connectors.

6. Data Link Layer

6.1 *Overview*—The data link layer has procedures for link connection and release, delimiting and synchronism, sequential control, error detection, and error recovery.

6.1.1 Link connection and release establish which system sends and which system receives information. Delimiting and synchronism provide for framing of the data and recognition of frames. Sequence control maintains the sequential order of information across the connection. Error detection senses transmission or format errors. Error recovery attempts to recover from detected errors by retransmitting defective frames or returning the link to a neutral state from otherwise unrecoverable errors.

6.1.2 The data link layer uses a character-oriented protocol to send messages between directly connected systems. (See ANSI X3.4-1986. Also, see Appendix X1 for the coding of the ASCI characters.) Some restrictions are placed on the characters which can appear in the message content.

6.1.3 The data link mode of operation is one-way transfer of information with alternate supervision. Information flows in one direction at a time. Replies occur after information is sent, never at the same time. It is a simplex stop-and-wait protocol.

6.1.4 At times, the two systems are actively operating to transfer information. The remainder of the time the data link is in a neutral state. See the state diagram in Annex A1.

6.1.5 There are three distinct phases in transferring information between instrument and computer system. In each phase, one system directs the operation and is responsible for continuity of the communication. The three phases assure the actions of sender and receiver are coordinated. The three phases are establishment, transfer, and termination.

6.2 *Establishment Phase (Link Connection)*:

6.2.1 The establishment phase determines the direction of information flow and prepares the receiver to accept information.

6.2.2 The sender notifies the receiver that information is available. The receiver responds that it is prepared to receive before information is transmitted.

6.2.3 A system which does not have information to send normally monitors the data link to detect the establishment phase. It acts as a receiver, waiting for the other system.

6.2.4 The system with information available initiates the establishment phase. After the sender determines the data link is in a neutral state, it transmits the **<ENQ>** transmission control character to the intended receiver. Sender will ignore all responses other than **<ACK>, <NAK>,** or **<ENQ>**.

6.2.5 Upon receiving the **<ENQ>**, the receiver prepares to receive information. All other characters are ignored. It replies with the **<ACK>** transmission control character signifying it is ready. With this sequence of events, the establishment phase ends and the transfer phase begins.

6.2.6 A receiver that cannot immediately receive information, replies with the **<NAK>** transmission control character. Upon receiving **<NAK>**, the sender must wait at least 10 s before transmitting another **<ENQ>**.

6.2.7 Systems not having the ability to receive information always respond to an **<ENQ>** by replying with a **<NAK>**. Systems not having the ability to send information never transmit an **<ENQ>**.

6.2.7.1 *Contention*—Should both systems simultaneously transmit an **<ENQ>**, the data link is in contention. The instrument system has priority to transmit information when contention occurs. Contention is resolved as follows:

(1) Upon receiving a reply of **<ENQ>** to its transmitted **<ENQ>**, the computer system must stop trying to transmit; it must prepare to receive. When the next **<ENQ>** is received, it replies with an **<ACK>** or **<NAK>** depending on its readiness to receive.

(2) Upon receiving a reply of **<ENQ>** to its transmitted **<ENQ>**, the instrument must wait at least 1 s before sending another **<ENQ>**.

6.3 *Transfer Phase*—During the transfer phase, the sender transmits messages to the receiver. The transfer phase continues until all messages are sent.

6.3.1 *Frames*—Messages are sent in frames, each frame contains a maximum of 247 characters (including frame overhead). Messages longer than 240 characters are divided between two or more frames.

6.3.1.1 Multiple messages are never combined in a single frame. Every message must begin in a new frame.

6.3.1.2 A frame is one of two types, an intermediate frame or an end frame. Intermediate frames terminate with the characters **<ETB>**, checksum, **<CR>** and **<LF>**. End frames terminate with the characters **<ETX>**, checksum, **<CR>** and **<LF>**. A message containing 240 characters or less is sent in a single end frame. Longer messages are sent in intermediate frames with the last part of the message sent in an end frame. The frame structure is illustrated as follows:

<STX> FN text **<ETB>** C1 C2 **<CR>** **<LF>** ← intermediate frame

<STX> FN text **<ETX>** C1 C2 **<CR>** **<LF>** ← end frame

where:

<STX>	=	Start of Text transmission control character
FN	=	single digit Frame Number **0** to **7**
text	=	Data Content of Message
<ETB>	=	End of Transmission Block transmission control character
<ETX>	=	End of Text transmission control character
C1	=	most significant character of checksum **0** to **9** and **A** to **F**
C2	=	least significant character of checksum **0** to **9** and **A** to **F**
<CR>	=	Carriage Return ASCII character
<LF>	=	Line Feed ASCII character

6.3.2 *Frame Number*—The frame number permits the receiver to distinguish between new and retransmitted frames. It is a single digit sent immediately after the **<STX>** character.

6.3.2.1 The frame number is an ASCII digit ranging from **0** to **7**. The frame number begins at **1** with the first frame of the Transfer phase. The frame number is incremented by one for every new frame transmitted. After **7**, the frame number rolls over to **0**, and continues in this fashion.

6.3.3 *Checksum*—The checksum permits the receiver to detect a defective frame. The checksum is encoded as two characters which are sent after the **<ETB>** or **<ETX>** character. The checksum is computed by adding the binary values of the characters, keeping the least significant eight bits of the result.

6.3.3.1 The checksum is initialized to zero with the **<STX>** character. The first character used in computing the checksum is the frame number. Each character in the message text is added to the checksum (modulo 256). The computation for the checksum does not include **<STX>**, the checksum characters, or the trailing **<CR>** and **<LF>**.

6.3.3.2 The checksum is an integer represented by eight bits, it can be considered as two groups of four bits. The groups of four bits are converted to the ASCII characters of the hexadecimal representation. The two ASCII characters are transmitted as the checksum, with the most significant character first.

6.3.3.3 For example, a checksum of 122 can be represented as 01111010 in binary or **7A** in hexadecimal. The checksum is transmitted as the ASCII character **7** followed by the character **A**.

6.3.4 *Acknowledgments*—After a frame is sent, the sender stops transmitting until a reply is received.

6.3.4.1 The receiver replies to each frame. When it is ready to receive the next frame, it transmits one of three replies to acknowledge the last frame. This reply must be transmitted within the timeout period specified in 6.5.2.

6.3.4.2 A reply of **<ACK>** signifies the last frame was

received successfully and the receiver is prepared to receive another frame. The sender must increment the frame number and either send a new frame or terminate.

6.3.4.3 A reply of **<NAK>** signifies the last frame was not successfully received and the receiver is prepared to receive the frame again.

6.3.4.4 A reply of **<EOT>** signifies the last frame was received successfully, the receiver is prepared to receive another frame, but is a request to the sender to stop transmitting. (See the following section on receiver interrupts.)

6.3.5 *Receiver Interrupts*—The receiver interrupt is a means for the receiver to request the sender to stop transmitting messages as soon as possible.

6.3.5.1 During the transfer phase, if the receiver responds to a frame with an **<EOT>** in place of the usual **<ACK>**, the sender must interpret this reply as a receiver interrupt request. The **<EOT>** is a positive acknowledgment of the end frame, signifies the receiver is prepared to receive next frame, and is a request to the sender to stop transmitting.

6.3.5.2 The sender does not have to stop transmitting after receiving the receiver interrupt request. If the sender chooses to ignore the **<EOT>**, the receiver must re-request the interrupt for the request to remain valid.

6.3.5.3 If the sender chooses to honor the receiver interrupt request, it must first enter the termination phase to return the data link to the neutral state. This gives the receiver an opportunity to enter the establishment phase and become the sender. The original sender must not enter the establishment phase for at least 15 s or until the receiver has sent a message and returned the data link to the neutral state.

6.4 *Termination Phase (Link Release)*—The termination phase returns the data link to the clear or neutral state. The sender notifies the receiver that all messages have been sent.

6.4.1 The sender transmits the **<EOT>** transmission control character and then regards the data link to be in a neutral state. Upon receiving **<EOT>**, the receiver also regards the data link to be in the neutral state.

6.5 *Error Recovery*—Methods are described which enable both sender and receiver to recover, in an orderly way, from errors in data transmission.

6.5.1 *Defective Frames*—A receiver checks every frame to guarantee it is valid. A reply of **<NAK>** is transmitted for invalid frames. Upon receiving the **<NAK>**, the sender retransmits the last frame with the same frame number. In this way, transmission errors are detected and automatically corrected.

6.5.1.1 Any characters occurring before the **<STX>** or **<EOT>** or after the end of the block character (the **<ETB>** or **<ETX>**) are ignored by the receiver when checking the frame. A frame should be rejected because:

(1) Any character errors are detected (parity error, framing error, etc.),

(2) The frame checksum does not match the checksum computed on the received frame,

(3) The frame number is not the same as the last accepted frame or one number higher (modulo 8).

6.5.1.2 Upon receiving a **<NAK>** or any character except an **<ACK>** or **<EOT>** (a **<NAK>** condition), the sender increments a retransmit counter and retransmits the frame. If this counter shows a single frame was sent and not accepted six times, the sender must abort this message by proceeding to the termination phase. An abort should be extremely rare, but it provides a mechanism to escape from a condition where the transfer phase cannot continue.

6.5.2 *Timeouts*—The sender and receiver both use timers to detect loss of coordination between them. The timers provide a method for recovery if the communication line or the other device fails to respond.

6.5.2.1 During the establishment phase, the sender sets a timer when transmitting the **<ENQ>**. If a reply of an **<ACK>**, **<NAK>**, or **<ENQ>** is not received within 15 s, a timeout occurs. After a timeout, the sender enters the termination phase.

6.5.2.2 During the establishment phase, if the computer (as receiver) detects contention, it sets a timer. If an **<ENQ>** is not received within 20 s, a timeout occurs. After a timeout, the receiver regards the line to be in the neutral state.

6.5.2.3 During the transfer phase, the sender sets a timer when transmitting the last character of a frame. If a reply is not received within 15 s, a timeout occurs. After a timeout, the sender aborts the message transfer by proceeding to the termination phase. As with excessive retransmissions of defective frames, the message must be remembered so it can be completely repeated.

6.5.2.4 During the transfer phase, the receiver sets a timer when first entering the transfer phase or when replying to a frame. If a frame or **<EOT>** is not received within 30 s, a timeout occurs. After a timeout, the receiver discards the last incomplete message and regards the line to be in the neutral state.

6.5.2.5 A receiver must reply to a frame within 15 s or the sender will timeout. A receiver can delay its reply for up to 15 s to process the frame or to otherwise go busy. Longer delays cause the sender to abort the message.

6.5.2.6 Receivers that cannot process messages fast enough to keep up with a sender may cause message buffer overflows in the sender. A sender can normally store at least one complete message. Storage space for more than one outgoing message is desirable but optional.

6.6 *Restricted Message Characters*—The data link protocol is designed for sending character based message text. Restrictions are placed on which characters may appear in the message text. The restrictions make it simpler for senders and receivers to recognize replies and frame delimiters. Additional characters are restricted to avoid interfering with software controls for devices such as multiplexers.

6.6.1 A **<LF>** character is not permitted to appear in the message text; it can appear only as the last character of a frame.

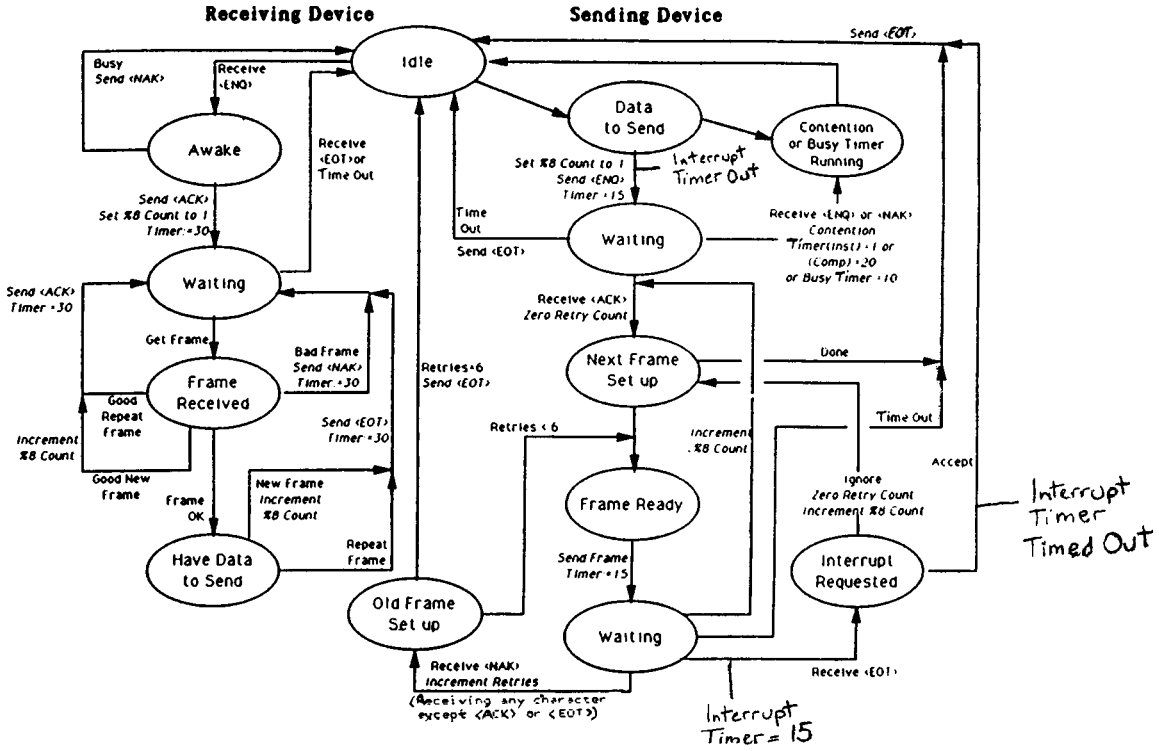
6.6.2 None of the ten transmission control characters, the **<LF>** format effector control character, or four device control characters may appear in message text. The restricted characters are: **<SOH>**, **<STX>**, **<ETX>**, **<EOT>**, **<ENQ>**, **<ACK>**, **<DLE>**, **<NAK>**, **<SYN>**, **<ETB>**, **<LF>**, **<DC1>**, **<DC2>**, **<DC3>**, and **<DC4>**.

ANNEX

(Mandatory Information)

A1. STATE DIAGRAM

A1.1 The state diagram is given in Fig. A1.1.



NOTE 1—"%8" represents modulo 8.
NOTE 2— " = " represents assignment of a value. "Timer: = 15" resets the timer to 15 s as used here.
NOTE 3—Arrow associated normal text denotes a condition; arrow associated italicized text denotes action taken.

FIG. A1.1 State Diagram

APPENDIX

(Nonmandatory Information)

X1. SEVEN-BIT ASCII CODE CHARTS

X1.1 Character codes are given in Fig. X1.1 and Fig. X1.2.

dec CHR	ASCII Character															
000	NUL	016	DLE	032	SP	048	0	064	@	080	P	096	`	112	p
001	SOH	017	DC1	033	!	049	1	065	A	081	Q	097	a	113	q
002	STX	018	DC2	034	"	050	2	066	B	082	R	098	b	114	r
003	ETX	019	DC3	035	#	051	3	067	C	083	S	099	c	115	s
004	EOT	020	DC4	036	$	052	4	068	D	084	T	100	d	116	t
005	ENQ	021	NAK	037	%	053	5	069	E	085	U	101	e	117	u
006	ACK	022	SYN	038	&	054	6	070	F	086	V	102	f	118	v
007	BEL	023	ETB	039	'	055	7	071	G	087	W	103	g	119	w
008	BS	024	CAN	040	(	056	8	072	H	088	X	104	h	120	x
009	HT	025	EM	041	)	057	9	073	I	089	Y	105	i	121	y
010	LF	026	SUB	042	*	058	:	074	J	090	Z	106	j	122	z
011	VT	027	ESC	043	+	059	;	075	K	091	[	107	k	123	{
012	FF	028	FS	044	,	060	<	076	L	092	\	108	l	124	|
013	CR	029	GS	045	-	061	=	077	M	093	]	109	m	125	}
014	SO	030	RS	046	.	062	>	078	N	094	^	110	n	126	~
015	SI	031	US	047	/	063	?	079	O	095	_	111	o	127	DEL

FIG. X1.1 Decimal Character Code

hex CHR	ASCII Character															
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	|
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	0	7F	DEL

FIG. X1.2 Hexadecimal Character Code