SEMI E4-0298[E]
SEMI EQUIPMENT COMMUNICATIONS STANDARD 1 MESSAGE TRANSFER (SECS-I)

[E] This standard was editorially corrected in January 1999, so that the document conforms technically to the balloted copy. Corrections were made to Table A7-2.

SUPERSEDED

1 Introduction

1.1 *Revision History* — This is the first major revision since the original release of SECS-I in 1980. Very little of the original intent of SECS-I has been altered, although there are a few significant additions. The changes are summarized in Appendix X1. This specification has been developed in cooperation with the Japan Electronic Industry Development Association Committee 12 on Equipment Communications.

1.2 *Scope* — The SECS-I standard defines a communication interface suitable for the exchange of messages between semiconductor processing equipment and a host. Semiconductor processing equipment includes equipment intended for wafer manufacturing, wafer processing, process measuring, assembly and packaging. A host is a computer or network of computers which exchange information with the equipment to accomplish manufacturing. This standard includes the description of the physical connector, signal levels, data rate and logical protocols required to exchange messages between the host and equipment over a serial point-to-point data path. This standard does not define the data contained within a message. The meaning of messages must be determined through some message content standard such as SEMI Equipment Communications Standard E5 (SECS-II).

1.3 *Intent* — This standard provides a means for independent manufacturers to produce equipment and/or hosts which can be connected without requiring specific knowledge of each other.

1.3.1 *Layered Protocol* — The SECS-I protocol can be thought of as a layered protocol used for point-to-point communication. The levels within SECS-I are the physical link, block transfer protocol, and message protocol. (See Application Note A1.1.)

1.3.2 *Speed* — It is not the intent of this standard to meet the communication needs of all possible applications. For example, the speed of RS-232 may be insufficient to meet the needs of transferring mass amounts of data or programs in a short period of time, such as might be required by high speed functional test applications.

1.3.3 *Network Support* — The method by which blocks of data are routed to a piece of equipment or find their way back to the proper host application is not specified by SECS-I. In a network, the roles of host and equipment might be assumed by any party in the network. In this situation, one end of the communications link must assume the role of the equipment and the other the role of the host.

1.4 *Applicable Documents*

1.4.1 *Electronics Industries Association Standards*[1]

EIA RS-232-C — Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange

EIA RS-269-B — Synchronous Signaling Rates for Data Transmission

EIA RS-334 — Signal Quality at Interface Between Data Processing Terminal Equipment and Synchronous Communication Equipment for Serial Data Transmission

EIA RS-422 — Electrical Characteristics of Balanced Voltage Digital Interface Circuits

EIA RS-423 — Electrical Characteristics of Unbalanced Voltage Digital Interface Circuits

1.4.2 *European Computer Manufacturing Association*[2]

ECMA/TC24/82/18 — "Network Layer Principles," Final Draft (April, 1982)

1.4.3 *Japanese Industrial Standards Committees*[3]

---

1 EIA Engineering Department, Standards Sales Office, 2001 Eye Street, N.W., Washington, D.C. 20006

2 European Computer Manufacturing Association, 114 Rue du Rhone, 1204 Geneva, Switzerland

3 Japanese Standards Association, 1-24, Akasaka 4 Chome, Minato-ku, Tokyo 107, Japan

Documents Document Sharing Hub
Full document details: SEMI E4-0298E

JIS C 6361 — "The Interface between Data Circuit Terminating Equipment (DCE) and Data Terminal Equipment (DTE) (25-pin Interface)"

1.4.4 *International Organization for Standardization*[4]

ISO 2110-1980 — Data Communications, Interface Connectors and Pin Assignment

1.4.5 *SEMI Specifications*

SEMI E5 — SEMI Equipment Communications Standard 2 — Message Content (SECS-II)

SEMI E6 — SEMI Facilities Interface Specification Format

1.5 *Overview of SECS-I* — The SECS-I standard defines point-to-point communication of data utilizing a subset of the international standard known in the U.S.A. as EIA RS-232-C and in Japan as JIS C 6361 for the connector and voltage levels. The actual transmission consists of 8-bit bytes sent serially with one start and one stop bit. The communication is bidirectional and asynchronous, but flows in one direction at a time. The direction is established by special characters and a handshake, after which the data itself is sent. Data is sent in blocks of 254 bytes or less. Each block consists of a 10-byte header followed by data. A message is a complete unit of communication in one direction and consists of 1 to 32,767 blocks. Each block header contains information for identifying the block as part of a specific message. Messages are paired by a request and its reply which together are called a transaction.

1.6 *Structure of Document* — This document is divided into sections which correspond to major aspects of the standard. The sections outline requirements as well as implications of the requirements. The standard may be implemented in a variety of ways, depending upon the computer environment where it is placed. Implementation is not part of the standard. Information which may be useful for implementation is included in the form of Application Notes.

2 Selected Definitions

2.1 The following brief definitions refer to sections providing further information.

ACK — "Correct Reception" handshake code. (See 5.2.)

application software — The software performing the specific task of the equipment or the host.

block — Header plus up to 244 bytes of data. (See 1.5, 6.7.)

block length — The number of bytes sent in the block transfer protocol. (See 5.6.)

block number — A 15-bit field in the header for numbering blocks in a message. (See 6.7.)

character — A byte sent on the SECS-I serial line. (See 4.1.)

checksum — A 16-bit number used to detect transmission errors. (See 5.7.)

communication failure — A failure in the communication link resulting from a failed send. (See Section 5.4.)

device ID — A 15-bit field in the header used to identify the equipment. (See 6.3.)

E-bit — A bit in the header identifying the last block of a message. (See 6.6.)

ENQ — "Request to Send" handshake code. (See 5.2.)

EOT — "Ready to Receive" handshake code. (See 5.2.)

equipment — The intelligent system which communicates with a host.

expected block — The block of a message which is expected by the message protocol. (See 7.4.4.)

header — A 10-byte data element used by the message and transaction protocols. (See 6.)

host — The intelligent system which communicates with the equipment.

length byte — The character used to establish the block length during transmission. (See 5.6.)

line control — A portion of the block transfer protocol. (See 5.8.2.)

master — The block transfer designation for the equipment. (See 5.5.)

message — A complete unit of communication. (See 7.)

4 ANSI, 1430 Broadway, New York, NY 10018

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E4-0298E

message ID — A 15-bit field in the header used in the process of message identification. (See 6.5, 7.3.1.)

multi-block message — A message sent in more than one block. (See 6.7, 7.2.2.)

NAK — "lncorrect Reception" handshake code. (See 5.2.)

open message — A multi-block message for which not all of the blocks have been received. (See 7.4.4.)

open transaction — A transaction in progress. (See 7.3.)

primary message — A message with an odd numbered message ID. Also the first message of a transaction. (See 6.5.)

primary/secondary attribute — The least significant bit of the lower message ID which indicates whether a block belongs to a primary or secondary message.

R-bit — A bit in the header signifying the direction of the message. (See 6.2.)

receiver — The end of the SECS-I link receiving a message. (See 5.8.4.)

reply — The particular secondary message corresponding to a primary message. (See 7.3.)

reply linking — The process of forming a transaction out of a primary and a secondary message. (See 7.3.1.)

retry count — The number of unsuccessful attempts to send a block in the block transfer protocol. (See 5.4.)

RTY — The retry limit or the number of times the block transfer protocol will attempt to retry sending a block before declaring a failed send. (See 5.4.)

sender — The end of the SECS-I link sending message. (See 5.8.3.)

secondary message — A message with an even numbered message ID. Also the second message of a transaction. (See 6.5.2.)

slave — The block transfer designation for the host (See 5.5.)

system bytes — A 4-byte field in the header used for message identification. (See 6.8.)

T1 — Receive inter-character timeout in the block transfer protocol. (See 5.3.1.)

T2 — Protocol timeout in the block transfer protocol. (See 5.3.2.)

T3 — Reply timeout in the message protocol. (See 5, 7.3.2)

T4 — Inter-block timeout in the message protocol. (See 7.4.3.)

transaction — A primary message and its associated secondary message, if any. (See 7.3.)

W-bit — A bit in the header signifying that a reply is expected. (See 6.4.)

3 Coupling

3.1 Coupling refers to the physical interface at the equipment. The host will provide compatible signals at this point. No restrictions are implied for any interface other than for equipment covered by this standard.

3.2 *Electrical Interface* — The connection will include a serial interface according to EIA Standard RS-232-C for interface Type E, full duplex communication, modified by the deletions, additions and exceptions described in this section.

3.2.1 *Connector* — The standard 25-pin connector described in ISO 2110-1980 will be used. A female connector will be mounted on the equipment and a male connector will be mounted on the cable from the host. The connector on the equipment will have female 4-40 threaded jack screw locks.

NOTE — Suitable 25-pin connectors known as Type "D" are similar to Amphenol MIN RAC 17 series with jack screw locks.

3.2.2 *Signal Pins* — Pins on the connector have functions as defined in Table 1. Pins 1, 2, 3, and 7 are required for all equipment complying with SECS-I. The two power supply pins, 18 and 25, are optional as indicated. Any other pins, if used, shall comply with the RS-232-C standard.

Table 1. Signal Connections

Pin	*RS-232-C Circuit*	*Description*
1	AA	Shield
2	BA	Data from Equipment
3	BB	Data to Equipment
7	AB	Signal Ground
18	—	+12 to +15 volts (opt.)
25	—	-12 to -15 volts (opt.)

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E4-0298E

3.2.3 *Logic Levels* — For the signal pins 2 and 3, the logic 1 level will be a voltage less than -3 volts and the logic 0 level will be a voltage greater than +3 volts. Voltages will never exceed ± 25 volts. These values correspond to those specified by the RS-232-C standard.

3.2.4 *Power Supplies* — Pins 18 and 25 are optional power supplies for driving external isolation circuits. When provided, both shall be present and must be able to supply at least 50 mA. (See Application Note A2 for example use.)

3.3 *Data Rate* — The supported data rates on signal pins shall be 9600, 4800, 2400, 1200, and 300 baud. The same data rate shall apply for data sent to and from the equipment. The data rate shall be controlled to better than 0.5%. (See RS-269-B and RS-334.) Optional rates of 19,200 and 150 baud may be supplied if desired.

3.4 *Physical Medium* — The connection with the host may involve any medium that provides the required RS-232-C quality, signal levels and data rate at the equipment connector. The quality of signal should be such that the effective bit error rate is less than 1×10^{-6}. This rate can be achieved easily with hard-wired systems. The distance limits specified in RS-232-C apply only to systems using the wiring technique described in RS-232-C. Since any method may be used in SECS-I as long as RS-232-C signals are supplied at the connector, the distance and isolation is dependent upon the design of the physical medium which is external to the SECS-I standard. (See Application Note A2.)

4 Character Structure

4.1 *Characters* — Data will be transmitted or received in a serial bit stream of 10 bits per character at one of the specified data rates. The standard character has one start bit (0), 8 data bits and one stop bit (1). All bit transmissions are of the same duration. The 8 data bits are numbered from 1 to 8 in the order sent (see Figure 1). The timing between characters is asynchronous with respect to the data rate. The 8 data bits may be any arbitrary code. The eight data bits will hereafter be referred to as a byte.

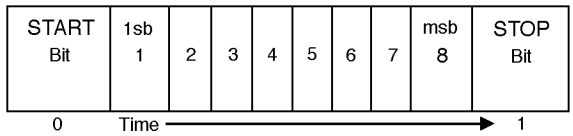

Figure 1
Character Structure

4.2 *Weighted Codes* — For bytes having weighted codes, bit one is the least significant and bit eight is the most significant. The most common weighted code is binary.

4.3 *Non-Weighted Codes* — For codes without numeric value such as ASCII, the bit numbers will be used as the entry into a standard code table for interpretation of the code. SECS-I performs no parity or other verification of the contents of individual bytes.

5 Block Transfer Protocol

5.1 The procedure used by the serial line to establish the direction of communication and provide the environment for passing message blocks is called the block transfer protocol. Most of the protocol is accomplished with a handshake of single bytes. When both ends of the line try to send at the same time, a condition known as line contention exists. The protocol resolves contention by forcing one end of the line, designated as the slave (always the host), to postpone its transmission and enter the receive mode. Retransmission of blocks is used to correct communication errors. The block transfer protocol is shown in flow chart form in Figure 2, and described below. Additional information is also contained in Application Notes A3 and A4.

5.2 *Handshake Bytes* — The four standard handshake codes used in the block transfer protocol are shown in Table 2. The three letter names, ENQ, EOT, ACK, and NAK correspond to the ASCII code having the same pattern.

Table 2. Handshake Codes

Name	*Code b8 b7.....b1*	*Function*
ENQ	00000101	Request to Send
EOT	00000100	Ready to Receive
ACK	00000110	Correct Reception
NAK	00010101	Incorrect Reception

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E4-0298E

5.3 *Timeout Parameters* — Timeouts are used to detect communications failures. A timeout occurs when the measured time between two events exceeds a pre-determined limit. Generally, the length of time that must pass before it can be assumed that an error has occurred depends upon the particular systems involved. The time required in one situation might be excessively long in another. Thus, the timeout values must be "tuned" to meet the application. In the block transfer protocol, there are two situations requiring timeout values. The two timeout values are called parameters T1 and T2.

5.3.1 *Inter-Character Timeout, T1* — The inter-character timeout, T1, limits the time between receipt of characters within a block after the length byte has been received and until the receipt of the second checksum byte.

5.3.2 *Protocol Timeout, T2* — The protocol timeout, T2, limits the time between sending ENQ and receiving EOT, sending EOT and receiving the length byte, and sending the second checksum byte and receiving any character.

5.4 *Retry Limit, RTY* — The retry limit, RTY, is the maximum number of times the Block Transfer Protocol will attempt to retry sending a block before declaring a failed send. (See 5.8.2.)

5.5 *Master/Slave* — The master/slave parameter is used in the resolution of contention (see 5.8.2). The host is designated as the slave. The equipment is designated as the master. This convention is based upon the assumption that the equipment is less able to store messages than the host.

5.6 *Block Lengths* — The unsigned integer value of the first byte sent after receipt of EOT is the length of the block being sent. The length includes all the bytes sent after the length byte, excluding the 2 bytes of the checksum. The maximum block length allowed by SECS-I is 254 bytes, and the minimum is 10 bytes.

5.7 *Checksum* — The checksum is calculated as the numeric sum of the unsigned binary values of all the bytes after the length byte and before the checksum in a single block. The checksum is sent as 16 bits in two bytes following the last byte of the block data. The high order eight bits of the checksum will be sent first, followed by the low order eight bits. The checksum is used by the receiver to check for transmission errors. The receiver performs the same checksum calculation on the received header and data.

5.8 *Algorithm* — The operation of the block transfer protocol is best understood by following the logic flow in Figure 2. This flow chart depicts the operation of the five states of the protocol — Receive, Idle, Send Line Control, and Completion. The flow chart shown in Figure 2 is not meant to imply that a particular implementation is required under this standard. However, any SECS-I block transfer protocol implementation must include all the logic described in Figure 2. The same algorithm is executed on each end of the SECS-I communications link.

5.8.1 *Idle State* — Both ends of the communications link are assumed to start in the Idle state. There are two primary activities of the protocol signified by the two exits from the Idle state. These are:

A. SEND — a message block is to be sent.

B. RECEIVE — the other end of the communications link has a message block to send

5.8.2 *Line Control* — The line control section establishes the transmission direction, resolves contention, and handles retries. When an ENQ is received in the Idle state, the Line Control responds with an EOT if the Block Transfer Protocol is ready to receive. The Block Transfer Protocol then goes to the Receive state. If a message block is to be sent, then an ENQ is sent. If an EOT is received in response to the ENQ within the time limit T2, the Block Transfer Protocol goes to the Send state.

5.8.2.1 If the slave receives an ENQ in response to the ENQ, contention has occurred. The slave postpones the send of its block until it receives a block from the master. The slave prepares to receive the incoming block and sends an EOT. When the block transfer protocol returns to the Idle state, the postponed block Send may be sent as if it were a new send request. After a master sends an ENQ, it can ignore all characters except an EOT. After a slave sends at ENQ, it can ignore all characters except an ENQ or EOT.

5.8.2.2 When the time between sending ENQ and receiving EOT exceeds T2, or the time between sending the second checksum byte and receiving any character exceeds T2, or a non-ACK character is received within time T2 after sending the second checksum byte, the Line Control will increment the retry count ("tries" in the flowchart). If the retry count does not exceed the value of the RTY parameter, then the Block Transfer Protocol will retry sending the block beginning with ENQ. If the retry count does exceed the

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E4-0298E

value of the RTY parameter, then a failed send has occurred.

5.8.3 *Send* — Once a send state is established, the first byte sent is the length, N, of the data in the block. After N more bytes have been sent, the two bytes of the checksum are sent. The sender computes the checksum based on the data in the block and the block header, but not the length byte. When the sender receives the ACK before time T2, the block is deemed properly sent. However, if the sender receives a non-ACK character before time T2, or no character within time T2, it returns to the line control state for a possible retry. In the Send state, characters received prior to sending the last checksum byte can be ignored.

5.8.4 *Receive* — Once a receive state is established, the first byte received is the length byte, N. The receiver counts and saves the following N + 2 bytes. The last two bytes are the checksum. The receiver compares the two checksum bytes against its own computation of the checksum. In a good block, the computed and received checksum are the same.

5.8.5 *Receive Completion* — After a block is correctly received, an ACK character is sent and the message protocol is notified that a block has been received. If T2 is exceeded while waiting for the length character, or Tl is exceeded between characters being received, then an NAK is sent. If the length byte is invalid or if the received checksum does not agree with the computed checksum, the receiver continues to listen for characters to ensure that the sender is finished sending. This is detected when the inter-character time exceeds T1, at which point the receive is aborted and a NAK is sent. In any case when an NAK is sent, the receiver may discard any data it has received for the block. The receiver returns immediately to the Idle state after sending an ACK or NAK.

5.8.6 *Send Completion* — After the second checksum byte is sent, if an ACK is received within time T2, the message protocol is notified that the send was successful. If a failed send occurs, the message protocol is informed of the failure. Application-level software must take the possibility of a send failure into account.

6 Header Structure

6.1 The operation of all communications functions above the block transfer protocol is linked to information contained in a 10-byte data element called the header. The header is always the first 10 bytes of every block sent by the block transfer protocol. The information in the header is also used by the message protocol (see 7). The fixed format of the header is described in this section. The general header structure is shown in Figure 3.

NOTE — The header also contains information required by SECS-II.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E4-0298E

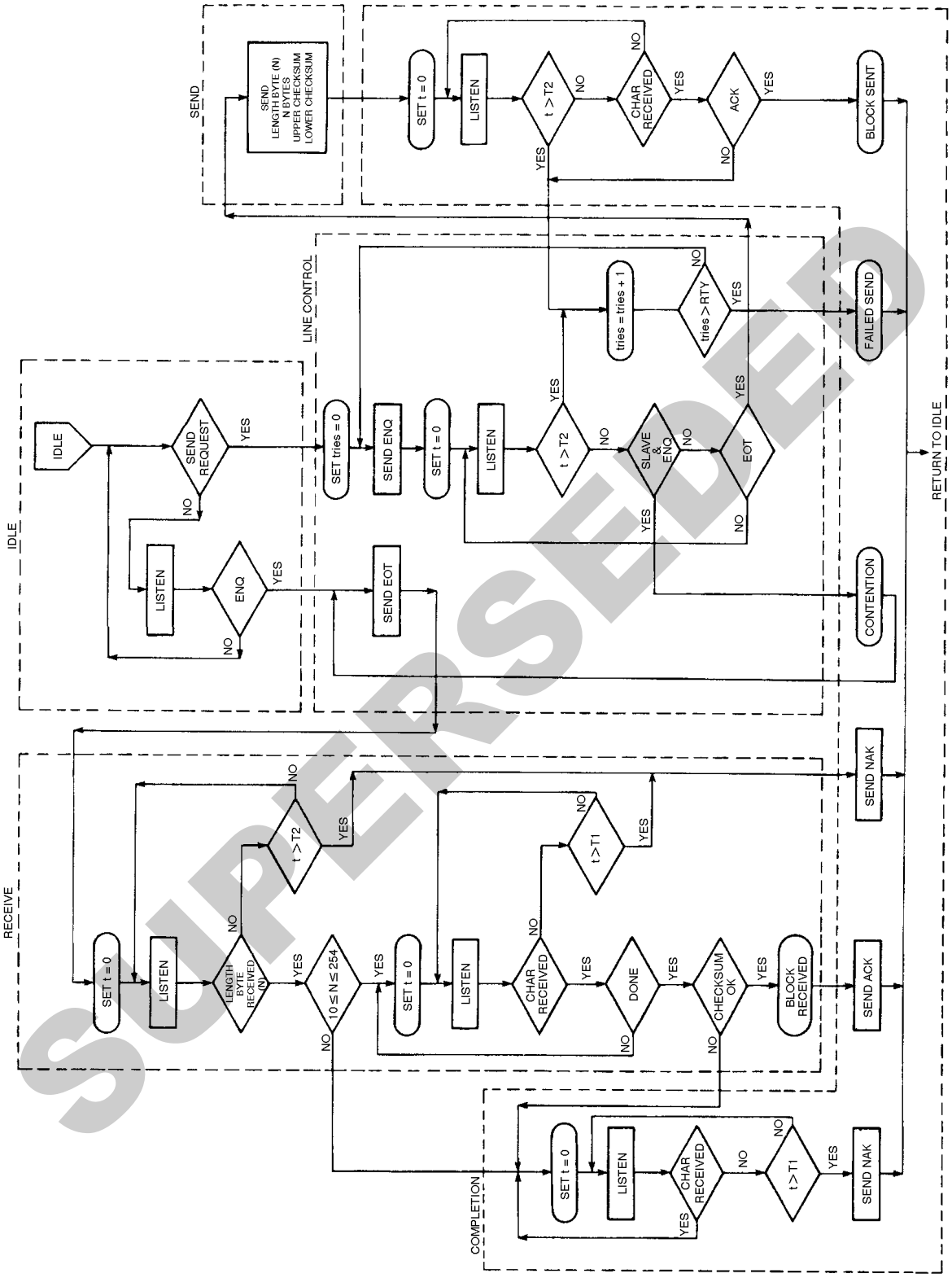

Figure 2

Block Transfer Protocol

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E4-0298E