SEMI E37-0298
HIGH-SPEED SECS MESSAGE SERVICES (HSMS) GENERIC SERVICES

1 Purpose

HSMS provides a means for independent manufacturers to produce implementations which can be connected and interoperate without requiring specific knowledge of one another.

HSMS is intended as an alternative to SEMI E4 (SECS-I) for applications where higher speed communication is needed or when a simple point-to-point topology is insufficient. SEMI E4 (SECS-I) can still be used in applications where these and other attributes of HSMS are not required.

HSMS is also intended as an alternative to SEMI E13 (SECS Message Services) for applications where TCP/IP is preferred over OSI.

It is intended that HSMS be supplemented by subsidiary standards which further specify details of its use or impose restrictions on its use in particular application domains.

2 Scope

High-Speed SECS Message Services (HSMS) defines a communication interface suitable for the exchange of messages between computers in a semiconductor factory.

3 Referenced Documents

3.1 *SEMI Standards*

SEMI E4 — SEMI Equipment Communication Standard 1 — Message Transport (SECS-I)

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

3.2 *IETF Documents*[1]

IETF RFC 791 — Internet Protocol

IETF RFC 792 — Internet Control Message Protocol

IETF RFC 793 — Transmission Control Protocol

IETF RFC 1120 — Requirements for Internet Hosts - Communication Layers

IETF RFC 1340 — Assigned Numbers. Note: This RFC supersedes RFC 820.

3.3 *POSIX Document*[2]

IEEE POSIX P1003.12 — Protocol Independent Interfaces (PII)

4 Terminology

API — Application Program Interface. In the case of TCP/IP, a set of programming conventions used by an application program to prepare for or invoke TCP/IP capabilities.

communication failure — a failure in the communication link resulting from a transition to the NOT CONNECTED state from the SELECTED state. (See Section 9.)

confirmed service (HSMS) — an HSMS service requested by sending a message from the initiator to the responding entity which requires that completion of the service be indicated by a response message from the responding entity to the initiator.

connection — a logical linkage established on a TCP/IP LAN between two entities for the purposes of exchanging messages.

control message — an HSMS message used for the management of HSMS sessions between two entities.

data message — an HSMS message used for communication of application-specific data within an HSMS session. A Data Message can be a Primary Message or a Reply Message.

entity — an application program associated with an endpoint of a TCP/IP connection.

header — a 10-byte data element preceding every HSMS message.

initiator (HSMS) — the entity requesting an HSMS service. The initiator requests the service by sending an appropriate HSMS message.

IP Address — Internet Protocol Address. A logical address which uniquely identifies a particular attachment to a TCP/IP network.

local entity — relative to a particular end point of a connection, the local entity is that entity associated with that endpoint.

1 The IETF documents can be obtained from The Network Information Center, Network Solutions, 14700 Park Meadow Drive, Suite 200, Chantilly, VA 22021 USA

2 POSIX documents can be obtained from Institute of Electrical and Electronic Engineers (IEEE), 345 East 47th Street, New York, NY 10017 USA

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E37-0298

local entity-specific — general qualifier to any procedure, option, issue, or other implementation matter which is not a subject of this standard and left to the discretion of the individual supplier.

message — a complete unit of communication in one direction. An HSMS Message consists of the Message Length, Message Header, and the Message Text. An HSMS Message can be a Data Message or a Control Message.

message length — a 4-byte unsigned integer field specifying the length of a message in bytes.

open transaction — a transaction in progress.

port — an endpoint of a TCP/IP connection whose complete network address is specified by an IP Address and TCP/IP Port number.

port number — (or TCP port number). The address of a port within an attachment to a TCP/IP network which can serve as an endpoint of a TCP/IP connection.

primary message — an HSMS Data Message with an odd numbered Function. Also, the first message of a data transaction.

published port — a TCP/IP IP Address and Port number associated with a particular entity (server) which that entity intends to use for receiving TCP/IP connection requests. An entity's published port must be known by remote entities intending to initiate connections.

receiver — the HSMS Entity receiving a message.

remote entity — relative to a particular endpoint of a connection, the remote entity is the entity associated with the opposite endpoint of the connection.

reply — an HSMS Data Message with an even-numbered function. Also, the appropriate response to a Primary HSMS Data Message.

responding entity (HSMS) — the provider of an HSMS service. The responding entity receives a message from an initiator requesting the service. In the event of a confirmed service, the responding entity indicates completion of the requested service by sending an appropriate HSMS response message to the initiator of the request. In an unconfirmed service, the responding entity does not send a response message.

session — a relationship established between two entities for the purpose of exchanging HSMS messages.

session entity — an entity participating in an HSMS session.

session ID — a 16-bit unsigned integer which identifies a particular session between particular session entities.

stream (TCP/IP) — a sequence of bytes presented at one end of a TCP/IP connection for delivery to the other end. TCP/IP guarantees that the delivered sequence of bytes matches the presented stream. HSMS subdivides a stream into blocks of contiguous bytes - messages.

T3 — reply timeout in the HSMS protocol.

T5 — connect Separation Timeout in the HSMS protocol used to prevent excessive TCP/IP connect activity by providing a minimum time between the breaking, by an entity, of a TCP/IP connection or a failed attempt to establish one, and the attempt, by that same entity, to initiate a new TCP/IP connection.

T6 — control Timeout in the HSMS protocol which defines the maximum time an HSMS control transaction can remain open before a communications failure is considered to have occurred. A transaction is considered open from the time the initiator sends the required request message until the response message is received.

T7 — connection Idle Timeout in the HSMS protocol which defines the maximum amount of time which may transpire between the formation of a TCP/IP connection and the use of that connection for HSMS communications before a communications failure is considered to have occurred.

T8 — network Intercharacter Timeout in the HSMS protocol which defines the maximum amount of time which may transpire between the receipt of any two successive bytes of a complete HSMS message before a communications failure is considered to have occurred.

TCP/IP — Transmission Control Protocol/Internet Protocol. A method of communications which provides reliable, connection-oriented message exchange between computers within a network.

TLI — Transport Level Interface. One particular API provided by certain implementations of TCP/IP which provides a transport protocol and operating system independent definition of the use of any Transport Level protocol.

transaction — a Primary Message and its associated Reply message, if required. Also, an HSMS Control Message of the request (.req) type, and its response Control Message (.rsp), if required.

unconfirmed service (HSMS) — an HSMS service requested by sending a message from the initiator to the responding entity which requires no indication of completion from the responding entity.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E37-0298

5 HSMS Overview and State Diagram

High-Speed SECS Message Services (HSMS) defines a communication interface suitable for the exchange of messages between computers in a semiconductor factory using a TCP/IP environment. HSMS uses TCP/IP stream support, which provides reliable two way simultaneous transmision of streams of contiguous bytes. It can be used as a replacement for SECS-I communication as well as other more advanced communications environments.

The procedure for HSMS communications parallels the more familiar SECS-I communications it replaces. The following steps are followed for any communications (HSMS or otherwise):

1. Obtain a communications link between two entities. In SECS-I, this is the RS232 wire physically connecting host and equipment. In HSMS, the link is a TCP/IP connection obtained by the standard TCP/IP connect procedure. Note that the abstract term "entity" is used instead of "host" or "equipment." This is because, while HSMS is used for SECS-I replacement, it has more general applications as well. In a SECS-I replacement application, the "host" is an "entity" and the "equipment" is an "entity."

2. Establish the application protocol conventions to be used for exchanging data messages between two entities. For SECS-I, this step is implicit in the fact that semiconductor equipment is physically connected on the two ends of the wire: the protocol is SECS-II.

   In the case of HSMS, the communications link is a dynamically established TCP/IP connection on a physical link which may be shared with many other TCP/IP connections using protocols other than HSMS or connections using non TCP/IP protocols. HSMS adds a message exchange (called the Select procedure) which is used to confirm to both entities that the particular TCP/IP connection is to be used exlusively for HSMS communications.

3. Exchange Data. This is the normal intended purpose of the communications link. In both SECS-I and HSMS, the procedure is to exchange SECS-II encoded messages for the control of semiconductor equipment and/or processes. Data exchange normally continues until one or both of the entities are taken off-line for equipment-specific purposes, such as maintenance.

4. Formally end communications. In SECS-I, there is no formal requirement here; the equipment to be taken off-line stops communicating.

   In HSMS, a message exchange (either the "bilateral" Deselect procedure or the "unilateral" separate procedure) is used for both parties to confirm that the TCP/IP connection is no longer needed for HSMS communications.

5. Break the communications link. In SECS-I, this is done by physically unplugging the host or equipment from the communications cable, which only occurs during repair or physical reconfiguration of the factory network environment.

   In HSMS, since it uses the dynamic connection environment of TCP/IP, the TCP/IP connection is logically broken via a release or a disconnect procedure without any physical disconnect from the network medium.

Two additional procedures, of a diagnostic nature, are supported in HSMS, which are generally not required by a simple SECS-I link or a SECS-I direct replacement. These follow:

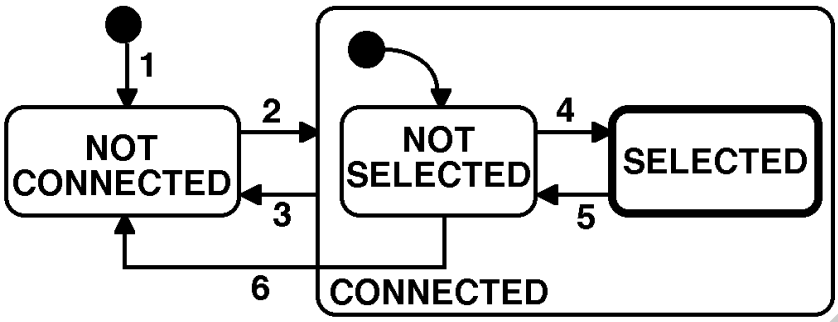
1. Linktest. This procedure provides a simple confirmation of connection integrity.

2. Reject. Because HSMS is intended to be extended to protocols other than just SECS-II (by means of subsidiary standards), it is possible that two entities can be connected (due to a configuration error) which use incompatible subsidiary standards. Also, during initial implementation, incorrect message types may be sent, or they may be sent out of order due to software bugs. The reject procedure is used to indicate such an occurrence.

5.1 *HSMS Connection State Diagram* — The HSMS state machine is illustrated in the diagram below. The behavior described in this diagram defines the basic requirements of HSMS: subsidiary standards may further extend these or other states.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E37-0298



5.2 *State Descriptions*

5.2.1 *NOT CONNECTED* — The entity is ready to listen for or initiate TCP/IP connections but either has not yet established any connections or all previously established TCP/IP connections have been terminated.

5.2.2 *CONNECTED* — A TCP/IP connection has been established. This state has two substates, NOT SELECTED and SELECTED.

5.2.2.1 *NOT SELECTED* — A substate of CONNECTED in which no HSMS session has been established or any previously established HSMS session has ended.

5.2.2.2 *SELECTED* — A substate of CONNECTED in which at least one HSMS session has been established. This is the normal "operating" state of HSMS: data messages may be exchanged in this state. It is highlighted by a heavy outline in the state diagram.

5.3 *State Transition Table*

Table 1

#	*Current State*	*Trigger*	*New State*	*Actions*	*Comment*
1	...	Local entity-specific preparation for TCP/IP communication.	NOT CONNECTED	Local entity-specific	Action depends on connection procedure to be used: active or passive.
2	NOTCONNECTED	A TCP/IP connection is established for HSMS communication.	CONNECTED - NOT SELECTED	Local entity-specific	none
3	CONNECTED	Breaking of TCP connection.	NOT CONNECTED	Local entity-specific	See Section 6.4.
4	NOT SELECTED	Successful completion of HSMS Select Procedure.	SELECTED	Local entity-specific	HSMS communication is now fully established: data message exchange is permitted.
5	SELECTED	Successful completion of HSMS Deselect or Separate.	NOT SELECTED	Local entity-specific	This transition normally indicates the end of HSMS communication and so an entity would immediately proceed to break the TCP/IP connection (transition 3 above).
6	NOT SELECTED	T7 Connection Timeout.	NOT CORRECTED	Local entity-specific	per Section 9.2.2

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E37-0298

6 Use of TCP/IP

6.1 *TCP/IP API* — The specification of a required TCP Application Program Interface (API) for use in implementations is outside the scope of HSMS. A given HSMS implementation may use any TCP/IP API — sockets, TLI (Transport Layer Interface), etc. — appropriate to the intended hardware and software platform, as long as it provides interoperable TCP/IP streams protocol on the network.

The appendix contains examples of the TCP/IP procedures referenced in this standard and sample scenarios using both the TLI (POSIX standard 1003.12) and the popular BSD socket model for TCP/IP communication.

6.2 *TCP/IP Network Addressing Conventions*

6.2.1 *IP Addresses* — Each physical TCP/IP connection to a given Local Area Network (LAN) must have a unique IP Address. IP Addresses must be assignable at installation time, and an HSMS implementation cannot select a fixed IP Address. A typical IP Address is 192.9.200.1.

IP imposes restrictions on these numbers which are outside the scope of the HSMS protocol. Consult Section 2.3 of RFC 791, Internet Protocol (IP) in Section 3.

6.2.2 *TCP Port Numbers* — A TCP Port Number can be considered as an extension of the IP Address.

HSMS implementations should allow configuring TCP Port to the full range of the TCP/IP implementation used. A typical TCP Port Number is 5000.

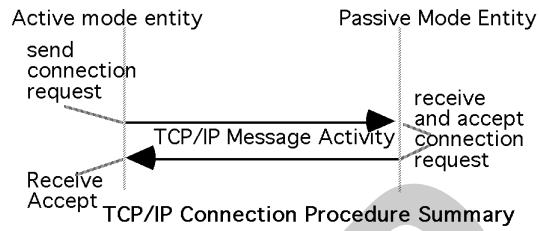
Conventions have been established for selecting TCP Port Numbers which are outside the scope of the HSMS protocol. Consult RFC 793, Transmission Control Protocol (TCP) in Section 3.

6.3 *Establishing a TCP/IP Connection*

6.3.1 *Connect Modes* — The procedures for establishing a TCP/IP connection are defined in RFC 793. However, not all the procedures defined by RFC 793 are supported by commonly available APIs. In particular, while RFC 793 permits both entities to initiate the connection simultaneously, this feature is rarely supported in available

APIs. Therefore, HSMS restricts an entity to one of the following modes:

- Passive Mode. The Passive mode is used when the local entity listens for and accepts a connect procedure initiated by the Remote Entity.
- Active Mode. The Active mode is used when the connect procedure is initiated by the Local Entity.



The appendix provides an example of how an entity may operate alternately in the active and passive modes to achieve greater flexibility in establishing communications.

6.3.2 *Passive Mode Connect Procedure* — The procedure followed by the Passive Local Entity is defined in RFC 793. It is summarized as follows:

1. Obtain a connection endpoint and bind it to a published port.
2. Listen for an incoming connect request to the published port from a remote entity.
3. Upon receipt of a connect request, acknowledge it and indicate acceptance of the connection. At this point, the connect procedure has completed successfully, and the CONNECTED state is entered (Section 5).

These procedures are carried out through the API of the local entity's implementation of TCP/IP. The appendices provide the API-specific procedures for the above steps using both TLI and BSD.

NOTE: A failure may occur during the above steps. The reason for failure may be local entity-specific or may be due to a lack of any connect request after a local entity-specific timeout. The action to be taken (for example: return to step 1 to retry) is a local entity-specific issue.

NOTE: See Section 9, Special Considerations, for issues relating to multiple connection requests to the same passive mode entity.

6.3.3 *Active Mode Connect Procedure* — The procedure followed by the Active Local Entity is defined in RFC 793. It is summarized as follows:

1. Obtain a connection endpoint.
2. Initiate a connection to the published port of a passive mode remote entity.
3. Wait for the receipt of the acknowledge and the acceptance of the connect request from the remote entity. Receipt of the acceptance from the remote

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E37-0298

entity indicates successful completion of the connect procedure, and the CONNECTED state is entered (Section 5).

These procedures are carried out through the API of the local entity's implementation of TCP/IP. The appendix provides the API-specific procedures for the above steps using both TLI and BSD.

NOTE: A failure may occur during the above steps. The reason for failure may be local entity-specific or may be due to a lack of any accept message after a local entity-specific timeout. The action to be taken is a local entity-specific issue. If, however, the local entity intends to retry the connection, it should do so subject to the T5 connect separation timeout (see "Special Considerations").

6.4 *Terminating a TCP/IP Connection* — Connection termination is the logical inverse of Connection establishment. From the Local Entity's perspective, a TCP/IP connection may be broken at any time. However, HSMS only permits termination of the connection when the connection is in the NOT SELECTED substate of the CONNECTED state.

The procedures for termination of a connection are defined in RFC 793. Either entity may initiate termination of the connection. The NOT CONNECTED state is entered, indicating the end of HSMS communications. The appendix illustrates the procedures for both release and disconnect using the TLI and BSD APIs.

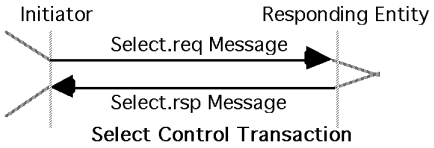
7 HSMS Message Exchange Procedures

HSMS defines the procedures for all message exchange between entities across the TCP/IP connection established according to the procedures in the previous section. As explained in the overview, once the connection is established, the two entities establish HSMS communications with the Select procedure. Then data messages may be exchanged in either direction at any time. When the entities wish to end HSMS communications, the Deselect or Separate procedure is used to end HSMS communications.

7.1 *Sending and Receiving HSMS Messages* — All HSMS procedures involve the exchange of HSMS messages. These messages are sent and received as TCP/IP streams using the previously established TCP/IP connection at standard priority. In particular, the use of "Urgent" data is not supported under HSMS (see RFC 793 for more information on send and receive procedures).

The appendix gives examples of sending and receiving HSMS messages using both TLI and BSD socket APIs.

7.2 *Select Procedure* — The Select procedure is used to establish HSMS communications on a TCP/IP connection using the Select.req and Select.rsp messages in a control transaction.



Although HSMS permits Select at any time in the CONNECTED state, subsidiary standards may further require the connection to be in the NOT SELECTED substate (see "Special Considerations").

7.2.1 *Initiator Procedure* — The procedure followed by the initiator is as follows.

1. The initiator of the select procedure sends the Select.req message to the responding entity.
2. If the initiator receives a Select.rsp with a Select Status of 0, The HSMS Select procedure completes successfully and the SELECTED state is entered (see Section 5).
3. If the initiator receives a Select.rsp with a non-zero Select Status, the Select completes unsuccessfully (no state transitions).
4. If the T6 timeout expires in the initiator before receipt of a Select.rsp, it is considered a communications failure (see "Special Considerations").

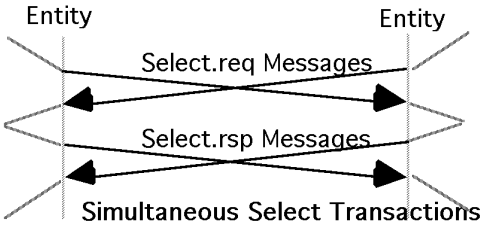
7.2.2 *Responding Entity Procedure* — The procedure followed by the responding entity is as follows.

1. The responding entity receives the Select.req.
2. If the responding entity is able to accept the select, it transmits the Select.rsp with a Select Status of 0. The HSMS Select Procedure for the responding entity is successfully completed, and the SELECTED state is entered (see Section 5).
3. If the responding entity is unable to permit the select, it transmits the Select.rsp with a non-zero Select Status. The HSMS Select Procedure for the responding entity completes unsuccessfully (no state transitions).

7.2.3 *Simultaneous Select Procedures* — If the subsidiary standards do not restrict the use of the Select, it is possible that both entities simultaneously initiate Select Procedures with identical SessionID's. In such a case, each entity will accept the other entity's select request by responding with a Select.rsp.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E37-0298

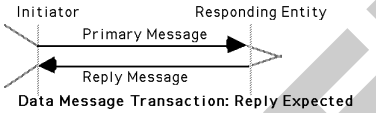


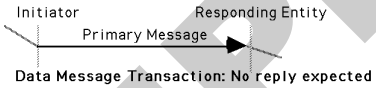
7.3 *Data Procedure* — HSMS data messages may be initiated by either entity as long as the connection is in the SELECTED state. Receipt of a data message when not in the SELECTED state will result in a reject procedure (see Section 7.7).

Data messages may be further defined as part of a data transaction as either a "Primary" or "Reply" data message. In a data transaction, the initiator of the transaction sends a primary message to the responding entity. If the Primary message indicates that a reply is expected, a Reply message is sent by the responding entity in response to the Primary.

The following types of Data Transactions are supported:

1. Primary Message with reply expected and the associated Reply Message.
2. Primary Message with no reply expected.

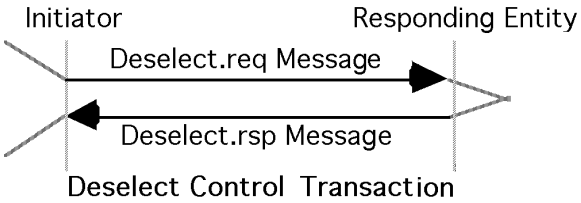




The specific procedures for these transactions are determined by the application layer and are subject to other standards (for example, E5 and E30 for GEM equipment using SECS-II encoded messages).

The applicable upper layer standard is identified by the message type. The type is determined from the specific format defined in Section 8. The normal type for HSMS messages is SECS-II text. Also refer to "Special Considerations" concerning the T3 Reply Timeout.

7.4 *Deselect Procedure* — The Deselect procedure is used to provide a graceful end to HSMS communication for an entity prior to breaking the TCP/IP connection. HSMS requires that the connection be in the SELECTED state. The procedure is as follows.



7.4.1 *Initiator Procedure*

1. The initiator of the Deselect procedure sends the Deselect.req message to the responding entity.
2. If the initiator receives a Deselect.rsp with a Deselect Status of 0, its Deselect procedure terminates successfully. The NOT SELECTED state is entered (see Section 5).
3. If the initiator receives a Deselect.rsp with a non-zero Deselect Status, its Deselect procedure terminates unsuccessfully. No state change occurs.
4. If the T6 timeout expires in the initiator before receipt of a Deselect.rsp, it is considered a communications failure (see "Special Considerations").

7.4.2 *Responding Entity Procedure*

1. The responding entity receives the Deselect.req message.
2. If the responding entity is in the SELECTED state, and if it is able to permit the Deselect, it responds using the Deselect.rsp with a zero response code. The responding entity's Deselect procedure completes successfully. The NOT SELECTED state is entered (see Section 5).
3. If the responding entity is unable to permit the Deselect, either because it is not in the SELECTED state or because local conditions do not permit the Deselect, it responds using the Deselect.rsp with a non-zero response code. The responding entity's Deselect procedure terminates unsuccessfully. No state change occurs.

7.4.3 *Simultaneous Deselect Procedures* — If the subsidiary standards do not restrict the use of the Deselect, it is possible that both entities simultaneously initiate Deselect Procedures with identical SessionID's. In such a case, each entity will accept the other entity's Deselect request by responding with the deselect.rsp.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E37-0298