High Speed Message Services (HSMS) Technical Education Report

SEMATECH
Technology Transfer 95092974A-TR

High Speed Message Services (HSMS) Technical Education Report

Technology Transfer # **95092974A-TR**
SEMATECH
October 31, 1995

Abstract: This document reviews industry activity related to adoption of the recently passed High Speed Messages Service (HSMS) standards of Semiconductor Equipment and Materials International (SEMI). The HSMS standards are intended to provide for communication speeds higher than those allowed by the Semiconductor Equipment Communication Standard I (SECS-I) RS-232 standard. This report gives background on the development of the HSMS standard, an overview of its components, and current adoption activity in the United States and Japan. The proceedings of the 1995 SEMI Standards Technical Education Program on HSMS are included in this document .

Keywords: Computer Software, Interoperability Standards, Japan, Telecommunication Systems

Authors: Jeff Acklen, Jack Ghiselli, Thomas J. Dinnel

Approvals: Jeff Acklen, Author
Jim Keosian, Project Manager
Dan McGowan, Technical Information Transfer Team Leader

Table of Contents

List of Figures

List of Tables

Preface

For questions on this report or for further information, please contact:

Jeff Acklen
Equipment Integration Project
SEMATECH
2706 Montopolis Drive
Austin, TX 78741
Phone: (512) 356-3431
Fax: (512) 356-3575

1 EXECUTIVE SUMMARY

This document contains information on the recently passed High Speed Message Services (HSMS) standard (E37-95) of Semiconductor Equipment and Materials International (SEMI). The HSMS standard was developed to provide a means for independent manufacturers to produce high-speed communication implementations that can be connected and made to interoperate without requiring specific knowledge of one another. HSMS is intended as an alternative to the SEMI E4 Semiconductor Equipment Communication Standard 1 (SECS-1), which utilizes RS-232. This report gives an overview of the HSMS standard and provides information on current use and adoption within the worldwide semiconductor industry.

2 BACKGROUND

Within the semiconductor industry, the primary equipment communication standard used since 1980 has been the SEMI E4 SECS-I protocol, which is based on RS-232 technology. In the mid-1980s, SEMI sanctioned a task force to develop a higher-speed alternative to E4. The Network Interface Task Force (NITF) chose General Motors' Manufacturing Automation Protocol (MAP) as the protocol, then developed and published the SEMI E13 standard in 1990. Because of very limited use of the MAP protocol, however, E13 was not well accepted by the industry. Due to the advancement of network technology and its usage, a new task force was formed in 1992 to reexamine the issue. The High Speed Message Task force (HMTF) was formed to prepare for ballot a standard to address network communications with higher speed and throughput than E4.

3 HSMS OBJECTIVES

Because of low usage of E13, a primary objective of the HMTF was to develop standards that could operate on a wide choice of platforms, thereby promoting wider adoption of the standard. The HMTF selected Transmission Control Protocol/Internet Protocol (TCP/IP) as the protocol for high speed messages. The HMTF recognized that TCP/IP was being widely used and was available on most computer platforms. The task force included in the development process input from industry equipment suppliers, equipment integrators, and IC manufacture users (see Section 7 for a list of participants). The task force developed straw proposals and used prototype reference implementations to validate the proposals. The use of a prototype implementation was considered key to gaining understanding and adoption within the industry. Through the development and subsequent revision of a reference prototype implementation and after 19 task force meetings over three years, the HSMS E37 standard became official in July 1994.

Two subsidiary standards also were passed and became part of the HSMS standard: HSMS General Services, SEMI E37.2 (HSMS-GS), and HSMS Single Session Services, SEMI E37.1 (HSMS-SS). HSMS-GS focuses on use of HSMS in the cluster tool environment, while HSMS-SS focuses on the use of HSMS as an RS-232 replacement in point-to-point implementations. Although HSMS-SS was technically approved in July 1994, the HMTF decided to change part of the document to improve readability. These changes were approved in

February 1995 and can be obtained from SEMI by requesting document number 2392A, HSMS-SS E37.1-96.

4 HSMS OVERVIEW

4.1 HSMS Generic Services (E37-95)

The HSMS document provides the fundamental components for developing an HSMS-compliant communications interface. The document defines message exchange procedures for using the TCP/IP network protocol. Procedures are described for the following areas:

- Establishing a communications link between entities using a TCP/IP connection procedure
- Establishing and maintaining the protocol conventions necessary for exchanging SECS messages between entities
- Exchanging data using TCP/IP
- Recognizing error conditions
- Formally ending communications to confirm both parties no longer need the TCP/IP connection
- Logically breaking the communications link without any physical disconnect from the network medium
- Testing the communications link for connection integrity purposes
- Rejecting connection attempts from incompatible subsidiary standards

In addition, the document describes special considerations, such as network timeouts and handling multiple connections, which should be taken into account in a TCP/IP implementation. Information on documentation required in an HSMS implementation and example message exchange procedures using the Berkley Sockets Definition (BSD) interface and the Transport Layer Interface (TLI) are provided in the appendixes.

4.2 HSMS-SS

HSMS-SS is a subsidiary standard that provides a proposed subset of HSMS, including the minimum set of services required for use as a direct SECS-I replacement. The document defines a different state machine than the HSMS-GS document, with limited capabilities. Specifically, HSMS-SS imposes the following limitations:

- It eliminates the use of a number of HSMS-GS procedures which are intended to be used by implementations which support multiple TCP/IP connections.
- It limits other HSMS-GS procedures to simplify operation for the specific case of SECS-I replacement.

The document also explains what documentation is required in an HSMS-SS implementation and provides application notes on support of multiple hosts.

4.3 HSMS -GS

HSMS-GS also is a subsidiary standard to HSMS. HSMS-GS provides a proper subset of the main standard, including support for complex systems containing multiple, independently accessible subsystems, such as cluster tool or track systems. The document details extensions to the HSMS state machine in the form of additional state transition definitions and added state information. These additions provide capabilities that permit individual subentities of complex systems to be accessed separately during HSMS procedures. Specifically, HSMS-GS details the following additional capabilities:

- A Session Entity List consisting of a set of all session entities that are accessible via TCP/IP connection from an outside entity
- A Selected Entity List comprising a list of entities that are currently selected for access on a given TCP/IP connection
- A Selection Count that corresponds to the number of entity IDs currently selected

The document also explains what documentation is required in an HSMS-GS implementation and provides application notes on supporting both HSMS-GS and HSMS-SS simultaneously.

5 HSMS ADOPTION

5.1 Domestic Activity

With the complete passage of HSMS in February 1995, interest and adoption activity among U.S. equipment suppliers and their customers is just beginning. The first adopters were equipment suppliers in the metrology and inspection area, who have had problems with transferring large amount of measurement data through RS-232. The restrictions of RS-232 have driven some metrology and inspection suppliers to develop proprietary network solutions for delivering information via the equipment’s interface.

The use of HSMS by other types of equipment suppliers is driven largely by IC manufacturers’ communications requirements. These manufacturers’ interest in using HSMS is high because of the availability of TCP/IP support in most factory environments. A large plus for utilizing HSMS-compliant equipment within the factory is the flexibility of physical placement of equipment within the factory. With HSMS, there is no maximum length of cable restrictions, and moving equipment within the factory only requires reconfiguring a network address instead of supplying a dedicated RS-232 cable. IC manufacturers who are currently including HSMS in procurement specifications for future equipment purchases include Advanced Micro Devices, Inc. (AMD), International Business Machines Corporation (IBM), Motorola, Inc., and National Semiconductor Corp. (in back-end assembly/test equipment).

5.2 Japan Activity

In May 1995, SEMI Japan distributed 300 questionnaires among Japanese IC manufacturers and equipment suppliers. Respondents (mostly software engineers and software development staff) represented 17 IC and 18 equipment manufacturers. Interest in using HSMS was high for both IC manufacturers (≅65%) and equipment suppliers (≅70%). The survey reported that about 30% of the responding IC manufacturers and more than 50% of the responding equipment suppliers plan to implement HSMS. Both groups also plan to implement commercial HSMS solutions instead of developing in-house solutions.

There also have been active HSMS education efforts in Japan. A SEMI Japan Technical Education Program was held in 1994, and an HSMS interpretability test was held at Texas Instruments (TI) Japan in February 1995. More information or proceedings of these events can be obtained from SEMI Japan at the following address:

SEMI Japan
7F Kenwa Building
4-7-15 Kudan-minami
Chiyoda-ku , Tokyo 102 Japan
Phone: (81)(3)3222.5755
Internet:62261920@eln.attmail.com

6 HSMS SOFTWARE SUPPLIERS

The software suppliers listed below provide commercial products for implementing, testing, and integrating HSMS-compliant communication interfaces. Over the past year, SEMATECH has worked with both suppliers, and each has successfully supported Equipment Integration (EI) projects with integration software products and/or services. The EI Project welcomes input on other commercial supplier products used in the integration process.

LPA Software, Inc.
30 Kimball Avenue South
South Burlington, VT 05403
(716) 248-9600
Products:
GEMVS (compliance testing)
PROTOCOL (HSMS driver support)

GW Associates
1183 Bordeaux Drive
Sunnyvale, CA 94089
(408) 745-1844
Products:
SECSIM/PRO (HSMS communications support)
SDR170 (HSMS communications driver)

7 SEMI TECHNICAL EDUCATION PROGRAM (STEP)

The following presentations where made during an HSMS SEMI Technical Education Program (STEP) at SEMICON/West 1995. Presented at STEP were HSMS implementation experiences from implementors in the equipment industry and the factory automation area. In addition, an interoperability demonstration was held to show actual HSMS implementations by multiple suppliers and prove that such implementations could interoperate. Participants in the demonstration included the following:

- Bruce Technologies International, Inc., with an equipment implementation of HSMS-SS
- GW Associates with HSMS-SS driver software and a SECS/HSMS simulator
- LPA Software with a SECS/HSMS simulator

- Realtime Performance with an equipment implementation of HSMS-GS
- Techware with an equipment control system implementation of HSMS-GS
- Universal Instruments with an equipment implementation of HSMS-SS

APPENDIX A

SEMI E-37
High Speed Message Services
SEMICON West 1995
July 10th, 1995

PROBLEM: The transmission of data using the SECS-I protocol (E-4) over RS-232 is too slow for data intensive applications and does not provide for local area network (LAN) access.

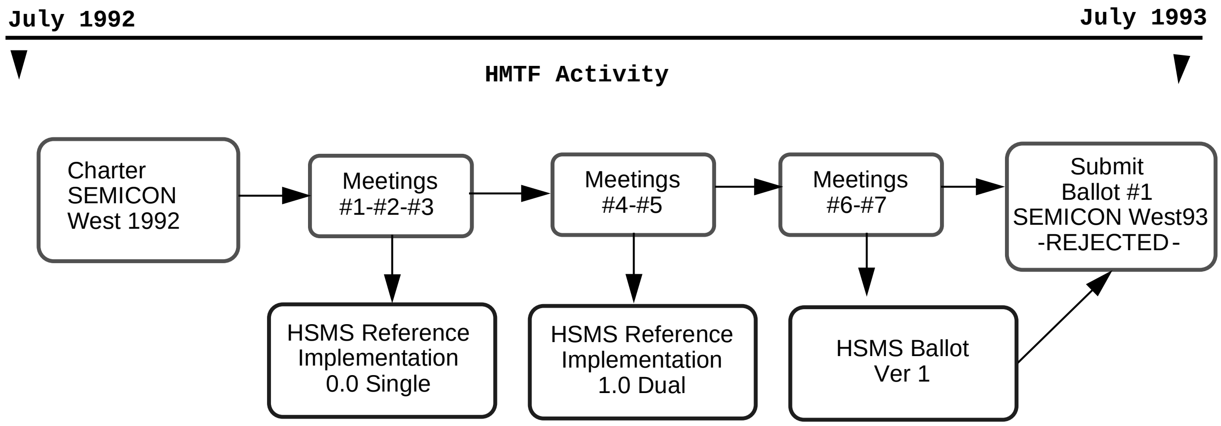
CHARTER: This task force will prepare and propose for ballot a SEMI standard for the transport of messages (including SECS-II and other formats) via network, to achieve faster transmission speed than currently available using SECS-I with RS-232.

KEY OBJECTIVES:

- Higher speed than RS-232
- Low cost
- High reliability
- Wide platform choice

Reach a solution quickly

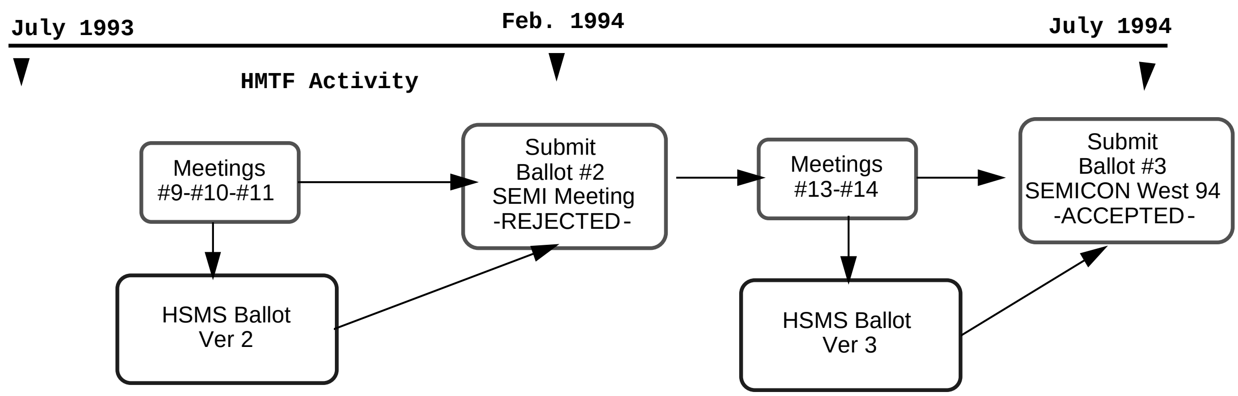
High Speed Message Services
92/93 Activities



Highlights:

- HMTF task force forms and agrees on charter
- TCP/IP chosen as LAN of choice
- Two TCP/IP reference implementations were developed
- HSMS ballot revision #1 balloted and rejected

High Speed Message Services
93/94 Activities

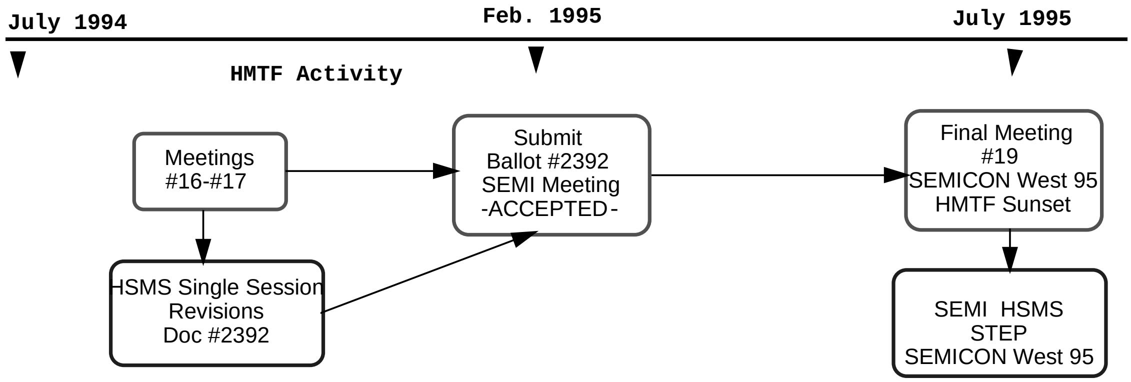


Highlights:

- Original ballot split into three ballots
  HSMS generic services (2213) - PASSES SC West 94
  HSMS single session services (2294) - FAILS SC West 94
  HSMS general session services (2295) - PASSES SC West 94

- HSMS single session reference implementation revised and tested

High Speed Message Services 94/95 Activities



Highlights:

- HSMS single session services (2392) - PASSED SEMI Meetings Feb. 95, New Orleans.

- HSMS SEMI Technical Education Program (STEP) scheduled for SC West 1995.

- HMTF task force concluded at SEMICON West 95

High Speed Message Services
SEMI E37-95

Task Force Participants:

- Jack Ghiselli
  GW Associates, Inc.
- Chris Brandson
  Techware, Inc.
- Paul Thordarson
  Brooks Automation
- Jeff Acklen
  SEMATECH
- Yoav Agmon
  Realtime Performance
- Bart Cox
  Realtime Performance
- Paul Meyer
  IBM Corp
- Jim Tamulonis
  IBM Corp
- Ray Paul
  KLA Instruments
- David Reis
  Tencor Instruments
- Ray Ellis
  Motorola
- Hansruedi Haenni
  ESEC SA
- John DeBolt
  PROMIS Systems
- Tom Baum
  Texas Instruments

APPENDIX B
THE HSMS-GS PROTOCOL

by
Paul Thoardsen
Brooks Automation
41 Wellman Street
Lowell, MA 01851

One wire, one standard ...

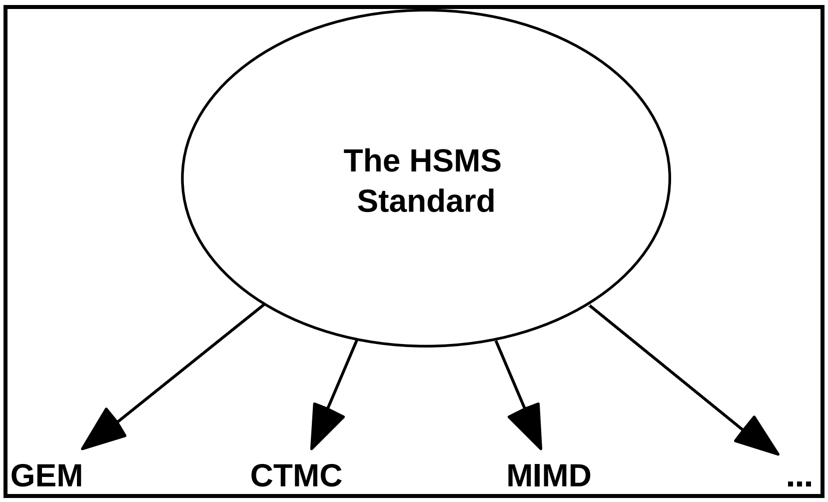


Figure 1 HSMS: The Initial Vision

One wire, one standard (mostly...)

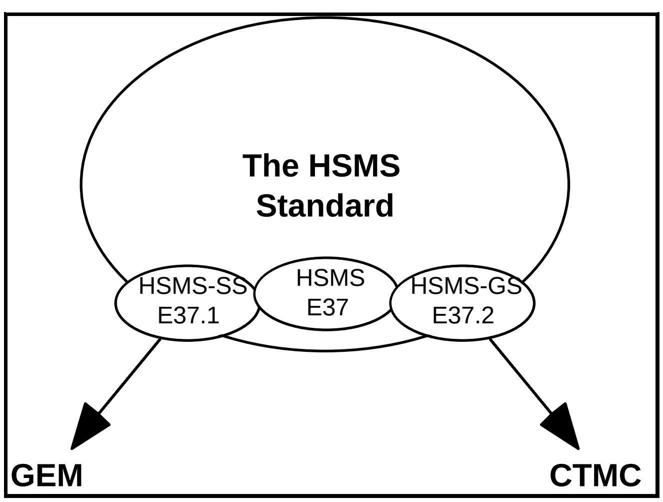


Figure 2 HSMS: The Resulting Vision

And now, for something completely different: a standard with three.

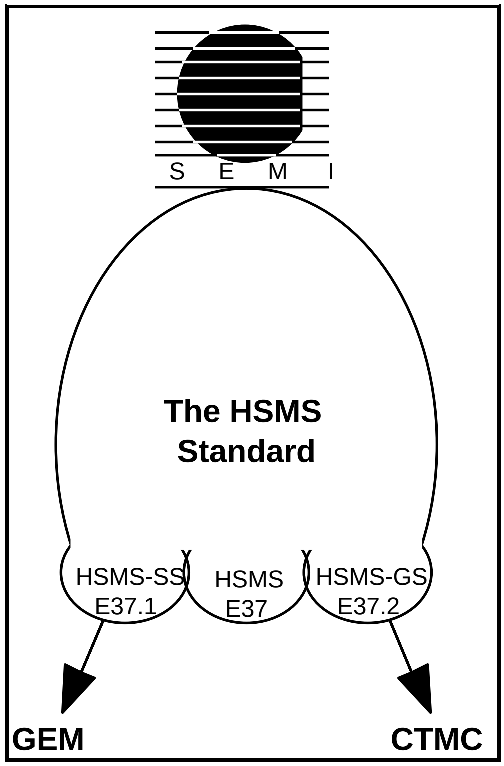


Figure 3 HSMS: The User's Confusion

Overview of HSMS-GS

- Motivation for HSMS-GS
  - Issues addressed
  - Features that address them
- Using HSMS-GS
  - Basic operating procedures
- Extending HSMS-GS
  - Examples to show extensibility with backwards compatibility

Motivation for HSMS-GS: Background

- Desired Features for HSMS (HMTF Charter)
  - Provide higher bandwidth on modern network
  - Support range of semiconductor applications (Generic Equipment Model [GEM], Cluster Tool Machine Communications [CTMC], etc.)
  - Support SECS-II and Other Formats
  - Leverage Widely Available Communications Software

- Conflicting Context
  - Need to support legacy systems (SECS-I-based)
  - Need to support emerging standards (Object Support Services [OSS], Material Movement Management Services [MMMS], etc.)
  - The consensus process produces significant compromise
- Other Context
  - Non-acceptance of SECS Message Services (SMS) (MAP-based)
  - No clear “preferred operating system”

Resulting HSMS-GS “Charter”

- Aligned with HSMS-SS as much as possible
  - Define a common base (HSMS)
  - Depart from it as little as possible
  - Could be used as a SECS-I replacement:
Single implementation for all applications should be possible
- Provide maximum flexibility within HSMS framework
  - Flexible entity addressing
  - Flexible application topology
- Support true extensibility
  - Recognize that requirements continue to evolve
  - Support migration to newer implementations with backwards compatible interoperability

HSMS-GS Design Principles and Features

- Divide and conquer: separate problems get separate solutions
  - Reliable end-to-end message delivery (TCP Connection Maintenance)
  - Semiconductor specific message content (HSMS Session Maintenance)
- Hooks for extensibility
  - “Spend a byte, save a standard”: SType and PType bytes
  - Procedures for dealing with unsupported features: the reject procedure
  - Limiting the scope of changes: Result of divide and conquer (e.g., new network affects very little)

Table 1 Operation of HSMS-GS

BSD API Calls	TLI API Calls	Network Activity	TLI API Calls	BSD API Calls
prepare to initiate a connection request			prepare to receive a connection request	

skt = socket(...);	tep = t_open(...);
	t_bind(tep, ...);
			tep = t_open(...);
			t_bind(tep,...);
				skt = socket(...);
				bind(skt,...);
				listen(skt,...);

initiate a connection request and wait for response			receive a connection request, accept it and send response	
connect(skt,...);	t_connect(tep,...); t_rcvconnect(tep,...);	TCP/IP Connect Req Msg(s) → ← TCP/IP Accept Msg(s)	t_listen(tep,...); t_accept(tep,...);	accept(skt,...);
Initiate an HSMS Select procedure: send request and receive response			Respond to HSMS select procedure: receive request and send response.	
write(skt,hdr,14); read(skt,hdr,14);	t_snd(tep,hdr,14,0); t_rcv(tep,hdr,14,...);	Select.req Message → ← Select.rsp Message	t_rcv(tep,hdr,14,...); t_snd(tep,hdr,14,0);	read(skt,hdr,14); write(skt,hdr,14);
send an HSMS data message as length bytes and header followed by Text			Receive an HSMS data message as length bytes and header followed by text.	

hdr->Len = length;
write(skt,hdr,14);
write(skt,Text,...);
	hdr->Len = length;
	t_snd(tep,hdr,14,0);
	t_snd(tep,Text,...);
		HSMS Data Message (hdr) →
		HSMS Data Message (text) →
			t_rcv(tep,hdr,14,...;
			t_rcv(tep,Text,...);
				read(skt,hdr,14);
				read(skt,Text,...);

Table 2 Deselect Procedure and Disconnect

BSD API Calls	TLI API Calls	Network Activity	TLI API Calls	BSD API Calls
Send the Deselect.req and receive Deselect.rsp			Receive the Deselect.req and send the Deselect.rsp.	
write(skt,hdr,14);	t_snd(tep,hdr,14,0);	Deselect.req Message →	t_rcv(tep,hdr,14,...);	read(skt,hdr,14);
read(skt,hdr,14);	t_rcv(tep,hdr,14,...);	← Deselect.rsp Message	t_snd(tep,hdr,14,0);	write(skt,hdr,14);
Disconnect the TCP/IP Connection			Respond to Disconnect of connection	
shutdown(skt,2); close(skt);	t_snddis(tep,...); t_close(tep);	← TCP/IP Disconnect Msg(s) →	t_rcvdis(tep,...);	close(skt);

Example Extensions

- Disclaimer: do not panic!
  - These are NOT proposed standards
  - They are for illustrative purposes only
- Demonstrates backwards compatibility
  - Examples can improve link operation but are optional in nature
- Specific examples
  - Automatic data compression
  - Recovery from failed link

Example 1: Data compression

- Concept: save network bandwidth through compression technology

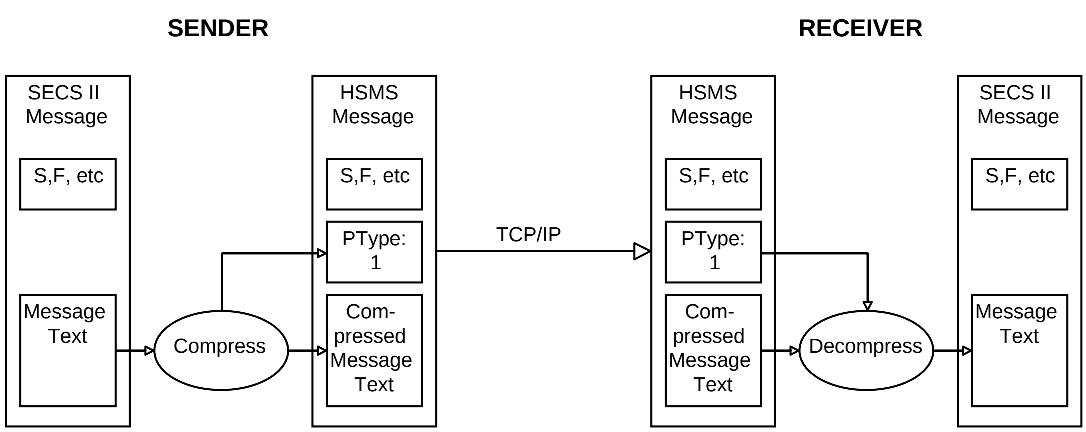


Figure 4 Data Compression

Example 1: Data compression - Interoperability Issues

- Procedure on implementation not supporting data compression
  - Standard behavior is sufficient:
    Simply use reject on PType = 1.
- Procedure on implementation supporting data compression
  - Must save initial message in uncompressed form prior to sending
  - If reject of PType = 1 received, must resend uncompressed text and use Uncompressed text for all subsequent messages

Example 2: Recovery from Failed Connection

- Concept: maintain context information which can be used on restarted connection
  - Save context under "Negotiated ID"
  - If connection broken and reestablished, recover the context using the ID
- Implementation: new "SType" messages
  - Confirmed service to establish ID
  - Confirmed service to reestablish context under previously established ID
  - Possible resend of messages that were in transit at time of failure
    Note: context must contain these messages

Example 2: Recovery - Interoperability Issues

- Procedure on implementations not supporting recovery
  - Standard behavior is sufficient: reject unsupported STypes
- Procedure on implementations supporting recovery
  - Initial attempt to establish ID may be rejected
  - If rejected, simply do not establish context
  - Recovery will not be possible but otherwise connection operates normally
  - Recovery and other extensions (e.g., compression)
  - Data compression and recovery are independent and can freely mix and match

Summary

- HSMS-GS realizes full potential of HSMS at minimum cost
  - Flexible addressing
  - Flexible network topology
- Minimizes cost of future enhancements
  - Structured to simplify changes of network
  - Support for backwards compatible extensions

APPENDIX C
THE HSMS-SS PROTOCOL

by
Jack Ghiselli, President
GW Associates, Inc.
1183 Bordeaux Dr., Suite 27
Sunnyvale, CA 94089

A HISTORY OF HSMS-SS

Although HSMS-SS was published in May 1995, it has been under development since 1992. HSMS-SS provides a higher-speed alternative to the trusty SECS-I protocol . SECS-I, published in 1980, was the first SEMI communication standard, and over the past 15 years has become the almost universal protocol of choice within the semiconductor industry. Almost all major types of semiconductor equipment provide communication interfaces are based on SECS-I. However, a major complaint about SECS-I was its relatively slow speed.

In 1986, NITF began work to develop a higher-speed protocol designed around General Motors' MAP, based on Token Bus (IEEE 802.4) technology. In 1986, semiconductor companies widely endorsed MAP, so the NITF adopted it, successfully designed a protocol, and published SEMI Standard E13, SECS Message Services, in 1990. Unfortunately, by that time MAP had widely fallen into disfavor. As a result, SEMI Standard E13, although published, has seen almost no use in production factories. One problem was that the Token Bus and open systems interconnect (OSI) Stack software of MAP was not widely available commercially.

In 1992, the HMTF tried again. At this time, there was wide consensus that the protocol should be based on TCP/IP, because it was widely available on many computers and operating systems. At first, HSMS was envisioned as a solution for metrology equipment suppliers, who wanted a faster protocol for their larger metrology data messages. Later, the HMTF recognized that there were common needs of this group and the Modular Equipment Standards Committee (MESC) CTMC group. Accordingly, HSMS was restructured into three related standards, as shown in Figure 5. SEMI Standard E37.1 defines HSMS-SS, which is essentially a replacement for SECS-I. SEMI Standard E37.2 defines HSMS-GS, which is used for MESC CTMC. Over both of these is SEMI Standard E37, HSMS Generic Services (HSMS), which defines aspects common to both HSMS-SS and HSMS-GS.

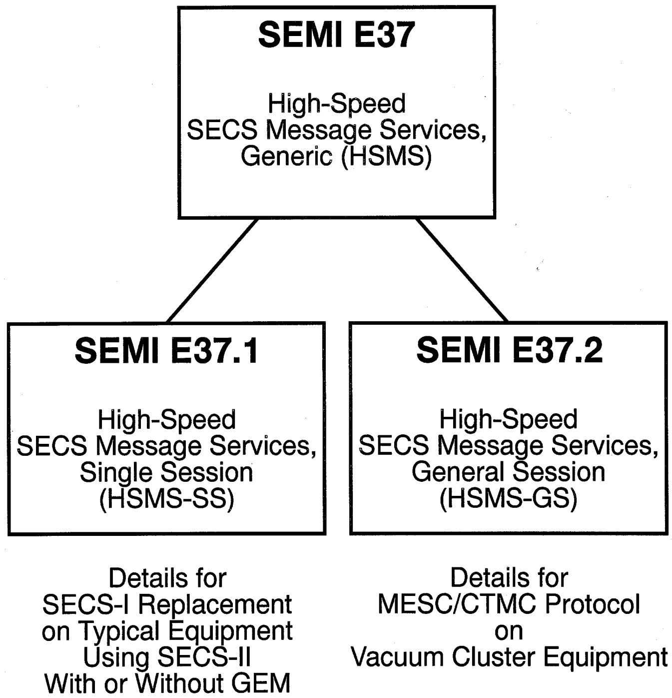


Figure 5 SEMI Standards for HSMS

HSMS-SS passed technical approval in July 1994. However, the HMTF determined that a re-writing would significantly improve the "readability" of HSMS-SS. The improved HSMS-SS document passed technical approval in February 1995, but unfortunately the SEMI North America Regional Standards Committee had not yet completed its procedural approval by the publication deadline. For this reason, SEMI Standard E37.1-95 (HSMS-SS for 1995) was published as the older, harder-to-read version. The newer easier-to-read version of HSMS-SS is SEMI Document 2392A, and probably be published as E37.1-96 in May, 1996. The newer document is available from SEMI by special request.

SHARED CABLES

Figure 6 compares SECS-I and HSMS-SS topology. In SECS-I, a Cell Controller might have two separate SECS-I RS-232 cables, one to each of two pieces of equipment. With HSMS-SS, a single physical Ethernet cable is used, on which HSMS-SS establishes two logical connections, one logical connection to each of the two equipments. The next example is more complicated, adding a third SECS-I link from the Cell Controller up to a Factory Host Computer. Using HSMS-SS, three logical connections share the single physical Ethernet cable. HSMS can greatly simplify factory cabling, since a single shared Ethernet cable replaces many dedicated RS-232 cables.

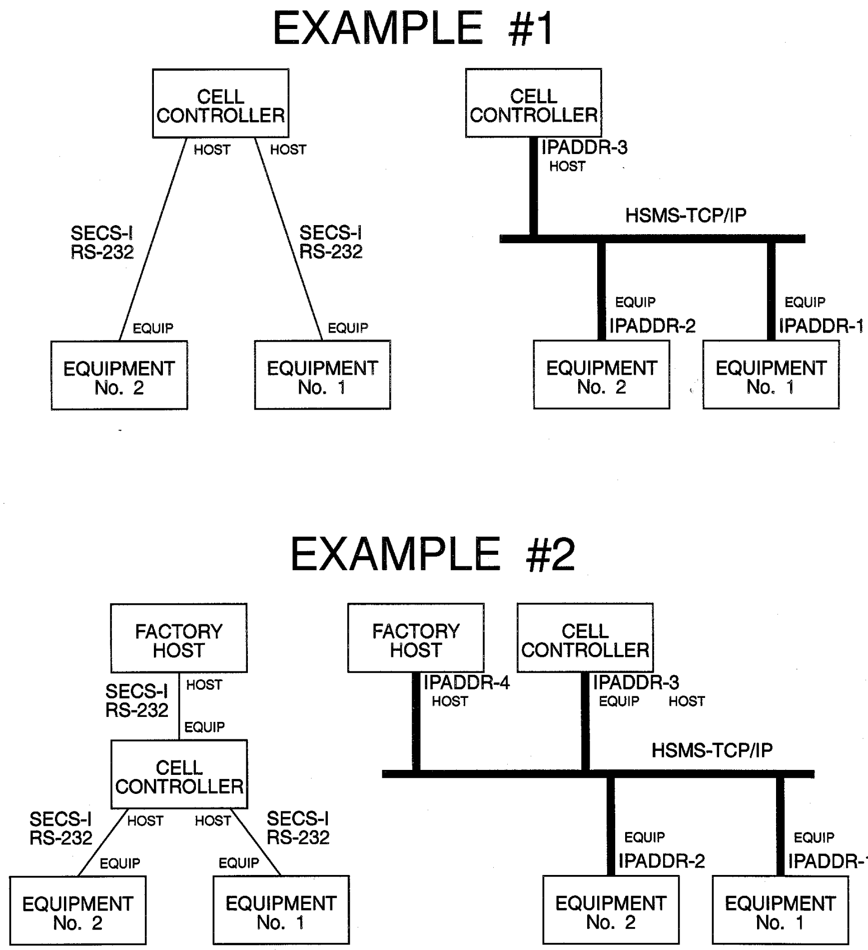


Figure 6 SECS-I RS-232 Connections Versus HSMS TCP/IP Ethernet Connections

HSMS MESSAGE FORMAT

HSMS-SS accesses TCP/IP utilizing a method called “TCP Streams.” Each TCP/IP Stream transmits data as an endless stream of bytes. TCP/IP is logically a full-duplex protocol, so there is one stream of bytes in one direction, and another stream of bytes in the reverse direction. HSMS-SS subdivides the TCP/IP Stream into discrete messages (see Figure 7). Each HSMS-SS message begins with a four-byte Message Length field. This Message Length is always transmitted most significant byte first and least significant byte last. Next, there is a ten-byte Message Header, and finally the useful text of the HSMS-SS message, which can range from no data (Header-Only Message) to several megabytes in size. The Message Text is formatted as specified in the SECS-II standard.

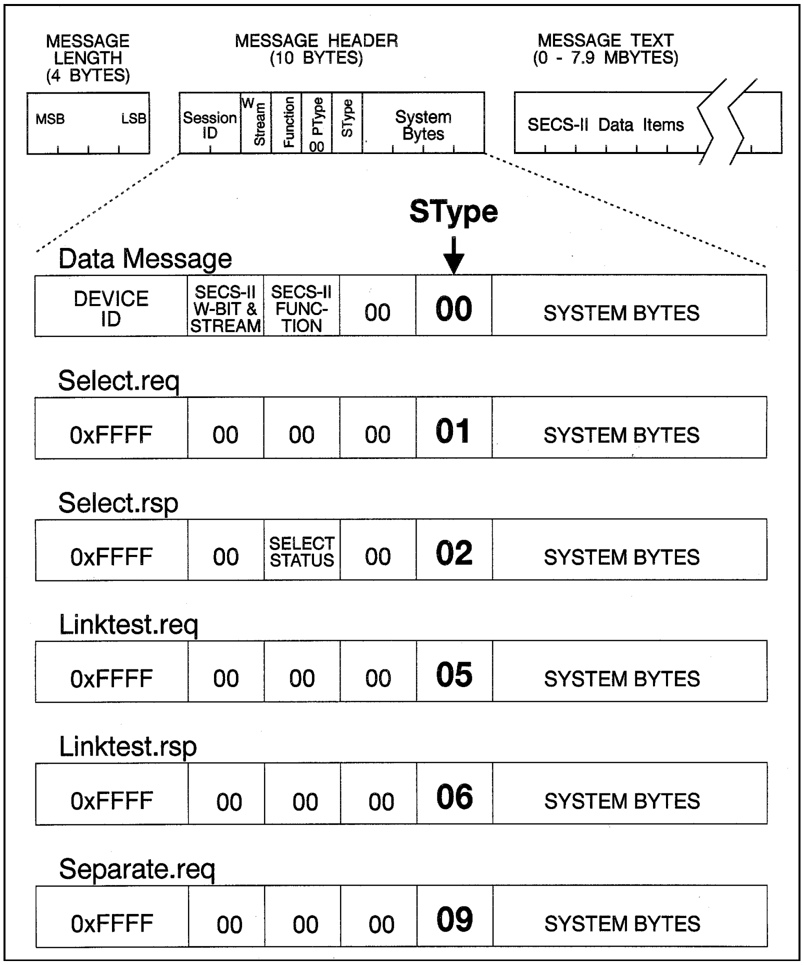


Figure 7 HSMS-SS Message Format

The Message Length specifies the number of bytes in the Message Header and Message Text (if any), but excludes the four bytes of the Message Length itself.

Within the ten-byte Message Header, the first two bytes contain a Device ID, useful in complex equipment to identify a major subsystem. The third and fourth bytes of the Message Header have different uses depending on Stype, as described below. The fifth byte of the Message Header (Ptype) is always zero. The sixth byte of the Message Header (Stype) contains a code indicating whether this message is a Data Message (containing useful application data) or one of the five or six HSMS-SS Control Messages used for link management. The Control Messages Select.req and Select.rsp are used to establish a connection between the Host and the Equipment. Linktest.req and Linktest.rsp are used to verify that the connection is still active. Separate.req is used to terminate the connection. The seventh through tenth bytes of the Message Header contain the System Bytes, which are used logically to associate a Primary Message with the corresponding Reply Message.

For a Data Message (Stype 00), bytes three and four of the Message Header contain SECS-II Stream and Function codes, which identify the topic of the message and which are further described later in the SECS-II standard. An odd numbered Function (least significant bit of

Function is "1") signifies a Primary Data Message, and an even numbered Function (with value one greater than the corresponding Primary Data Message) signifies a Reply Data Message.

The ten-byte HSMS-SS Message Header looks a lot like the older SECS-I Block Header. In SECS-I, we had a one-byte Block Length, and a ten-byte header for each block. For HSMS-SS, the TCP/IP layer provides “hidden” logic for blocking the TCP/IP Stream transmission, in HSMS-SS we don't need to worry about blocks. Instead, we deal with complete SECS-II messages. Only one ten-byte header is needed for the entire HSMS-SS message. In SECS-I, the fifth and sixth bytes of the Block Header contained a Block Count and E-Bit. In HSMS-SS, we don't worry about blocks, so these bytes of the header are used for Ptype and Stype, as described above.

HSMS-SS PROCEDURES

Figure 8 shows the HSMS Procedures. The most important procedure is the Data procedure, which consists of sending a Primary SECS-II message in one direction, and possibly sending the appropriate SECS-II reply message back. Like SECS-I, either end of the HSMS-SS connection can initiate a transaction, several transactions can be in progress simultaneously, and HSMS-SS associates each Reply Message to the appropriate Primary Message. Unlike SECS-I, HSMS-SS also defines several additional procedures which are used to manage the TCP/IP connection.

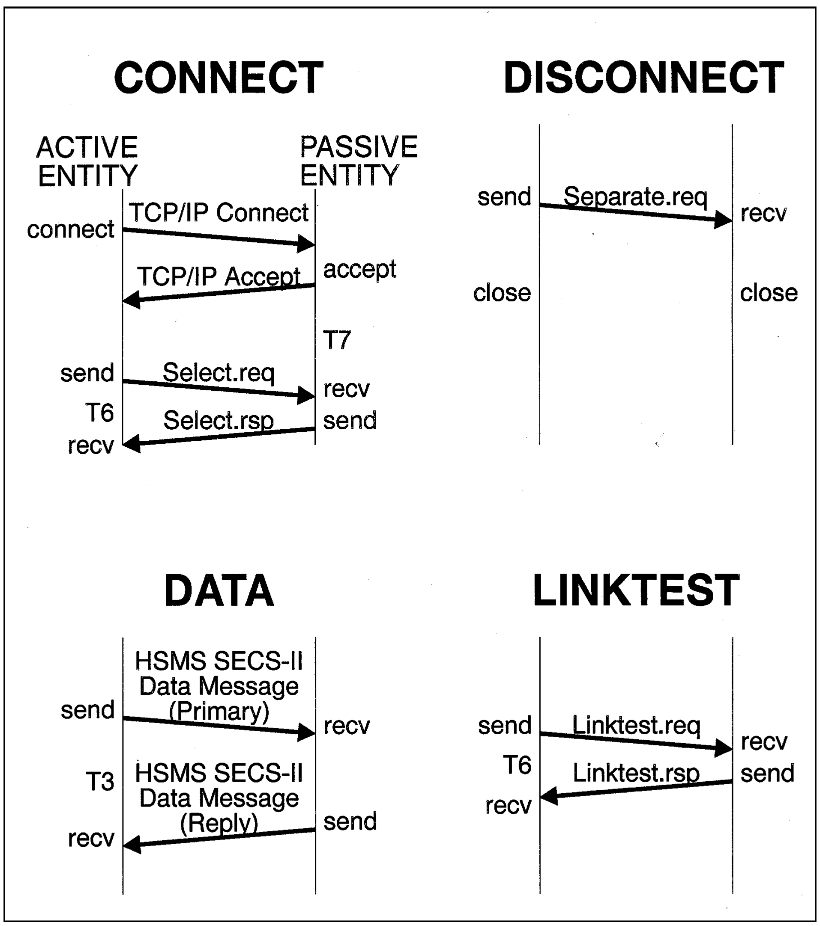


Figure 8 HSMS-SS Procedures

"CONNECT" PROCEDURE

The HSMS Connect Procedure establishes a logical connection between the Host and the Equipment. One end of the link, called the ACTIVE entity, initiates establishment of the connection by means of the TCP “connect” function. The opposite entity, called the PASSIVE entity, accepts the connection by means of the TCP "accept" function. In early draft specifications of HSMS-SS, this was the only logic required. However, testing of actual reference implementations uncovered undetected error conditions. For example, an HSMS ACTIVE Entity could (by mistake) establish a connection to an inappropriate partner, such as a print server, with very strange results when it proceeded to send SECS data messages. In the published HSMS-SS standard, the Connect Procedure has been enhanced to detect such errors. The ACTIVE Entity must send the HSMS-SS Control Message Select.req, and the PASSIVE Entity replies with Select.rsp. This step assures that both entities are in fact HSMS-SS entities.

Various errors can occur during the Connect Procedure. The connect may fail, because of no matching accept. In this case, the ACTIVE Entity waits a specified interval (the T5 Timeout), and then again attempts the connect. After the TCP/IP connect/accept succeeds, the PASSIVE Entity initiates the T7 Timeout. If the PASSIVE entity fails to receive Select.req within the T7 timeout, the Connect Procedure fails. After the ACTIVE Entity sends Select.req, it starts the T6 Timeout. If the ACTIVE Entity fails to receive Select.rsp within the T6 timeout, the Connect Procedure fails.

In most conventional TCP/IP implementations, even after the PASSIVE Entity has accepted the TCP/IP connect, a second ACTIVE Entity can attempt to connect to the same PASSIVE Entity. Many non-HSMS TCP/IP protocols (e.g., a print server) allow several ACTIVE Entities to connect simultaneously to a single PASSIVE Entity. HSMS-SS typically does not allow this. So, the Select.rsp message contains a Select Status code (in Message Header Byte 4). In a successful connect procedure, the PASSIVE Entity sends Select.rsp with Select Status zero. When rejecting an attempted simultaneous connect by a second ACTIVE Entity, the PASSIVE Entity sends Select.rsp with Select Status non-zero. Testing on a variety of TCP/IP implementations has proven that this logic provides rapid detection of a connect failure by both sides.

"SEPARATE" PROCEDURE

To break an HSMS-SS connection, either side can send the HSMS-SS Control Message Separate.req. After sending or receiving Separate.req, and HSMS-SS entity should use the TCP/IP "close" function to terminate the connection.

"LINKTEST" PROCEDURE

It is always possible for one end of a TCP/IP connection to "die" or to close the link ungracefully. Some TCP/IP implementations report this condition promptly to the other end of the link. Unfortunately, certain TCP/IP implementations may take as long as 15 minutes to report such a condition. For this reason, the Linktest procedure is sometimes useful to determine whether the HSMS-SS connection is still active. The entity initiating the test sends the HSMS-SS Control Message Linktest.req, and starts the T6 timeout. The opposite end replies with Linktest.rsp. If the initiating entity fails to receive Linktest.rsp within the T6 timeout, it assumes that the connection has failed and terminates it using the TCP/IP close function.

HSMS-SS STATE MACHINE

Figure 9 shows the HSMS-SS Connect logic expressed as a finite state machine. In the TCP/IP Not Connected state, the PASSIVE entity listens on the line, and the ACTIVE entity repeatedly attempts the TCP/IP connect. When the TCP/IP connect succeeds, both entities transit to the HSMS Not Selected state, and the ACTIVE entity initiates the HSMS Select control transaction. If this fails, the connection is broken and state transits back to TCP/IP Not Connected. If the Select succeeds, the state transits to HSMS Selected, which is the normal "active" state of the HSMS-SS link, where SECS-II messages are exchanged. An HSMS-SS "separate" procedure or various error conditions can break the connection, and state transits back to TCP/IP Not Connected. As with SECS-I, T3 Reply Timeout errors do not break the connection.

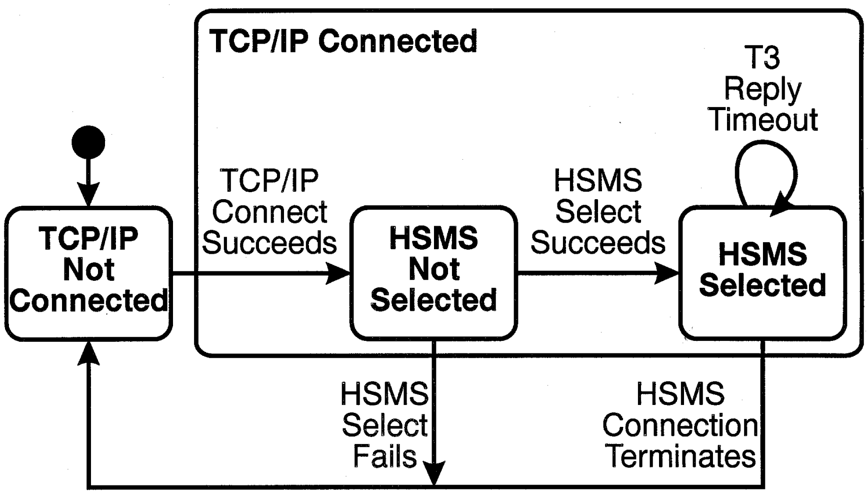


Figure 9 HSMS-SS State Machine

CONFIGURING HSMS-SS PARAMETERS

Figure 10 shows the configuration parameters for HSMS-SS. Like SECS-I, you must configure one end of the link as the HOST and the other end as EQUIPMENT, and you must specify the Equipment’s SECS Device ID. For connection purposes, you must configure one end as ACTIVE and the other end as PASSIVE. There is some advantage to configuring HOST as ACTIVE, since this makes it easier to switch Host Computers; however, this is a factory choice. You must configure the IP Address and TCP Port number of the PASSIVE Entity. The T3 Reply Timeout is the same as for SECS-I. The T5 Connect Separation Timeout controls how often the ACTIVE Entity will re-try its attempts to establish the TCP/IP connection. The T6 Control Transaction Timeout is a reply timeout for HSMS Control Transactions (Select and Linktest). The T7 Not Selected Timeout controls how long the PASSIVE Entity will wait to receive Select.req during a Connect procedure. The T8 Network Timeout controls how long an application waits for an unresponsive TCP/IP layer.

Parameter	Typical Value	
HOST or EQUIPMENT		*
SECS DEVICE ID		*
ACTIVE or PASSIVE	Host is ACTIVE	
PASSIVE Entity IP Address	192.9.200.1	
PASSIVE Entity TCP Port	5000	
T3 Reply Timeout	30 Seconds	*
T5 Connect Separation Timeout	10 Seconds	
T6 Control Transaction Timeout	10 Seconds	
T7 Not Selected Timeout	10 Seconds	
T8 Network Timeout	10 Seconds	

* = Same as SECS-I

SECS-I Parameters Not Needed:
Baud Rate, T1, T2, T4, RTY

Figure 10 HSMS Configuration

Configuring timeouts in a shared Ethernet network takes more wisdom than was required for the older dedicated SECS-I link. In some complex networks using bridges and routers, the timeout values may need to be quite large.

COMPARING SECS-I AND HSMS-SS PROTOCOL STACKS

As shown in Figure 11, the GEM and SECS-II standards can be used with either SECS-I or HSMS-SS. SECS-I uses RS-232 and a four-wire serial cable as its physical layer. HSMS-SS requires a foundation layer of TCP/IP software. Most HSMS-SS users prefer Ethernet (IEEE 802.3), but TCP/IP also supports other protocols, such as Token Ring (IEEE 802.5). Even with Ethernet, there are several cable options, including thick Coaxial cable, Thin Coaxial Cable (10-Base-2), and Twisted Pair (10-Base-T). The HSMS standards do not specify the physical layer, so it is important for users and suppliers of HSMS to negotiate agreements for local standards at the physical layer. An advantage of not specifying the physical layer is that as new TCP/IP supported protocols such as Fast Ethernet, Fiber Distributed Data Interface (FDDI), etc., become commercially practical, they can be used with HSMS-SS.

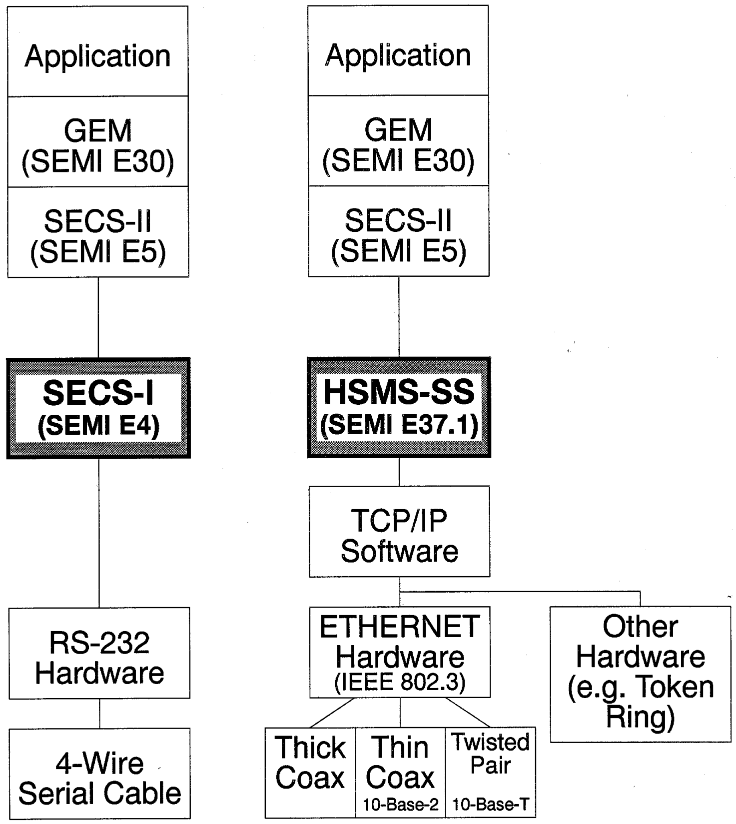


Figure 11 SECS-I versus HSMS-SS Protocol Stacks

SUPPORTING BOTH HSMS-SS AND SECS-I

Most existing factories are set up to use SECS-I, while only a few newer factories are ready to use HSMS-SS. A gradual shift to HSMS-SS is anticipated, but for several years, equipment suppliers will need to provide both protocols for different customers. Equipment can be designed with "plug and play" software components to make this straightforward. Figure 12 shows how substituting SECS-I for HSMS changes only the lower levels of the protocols. SECS-II, GEM, and (most importantly) the equipment or Line Control Computer application software does not need to change. Converting an equipment or Line Control Computer is easy—one simply swaps models of the low-level commercial SDR SECS Driver software. Except for speed, most other aspects of the equipment remain unchanged.

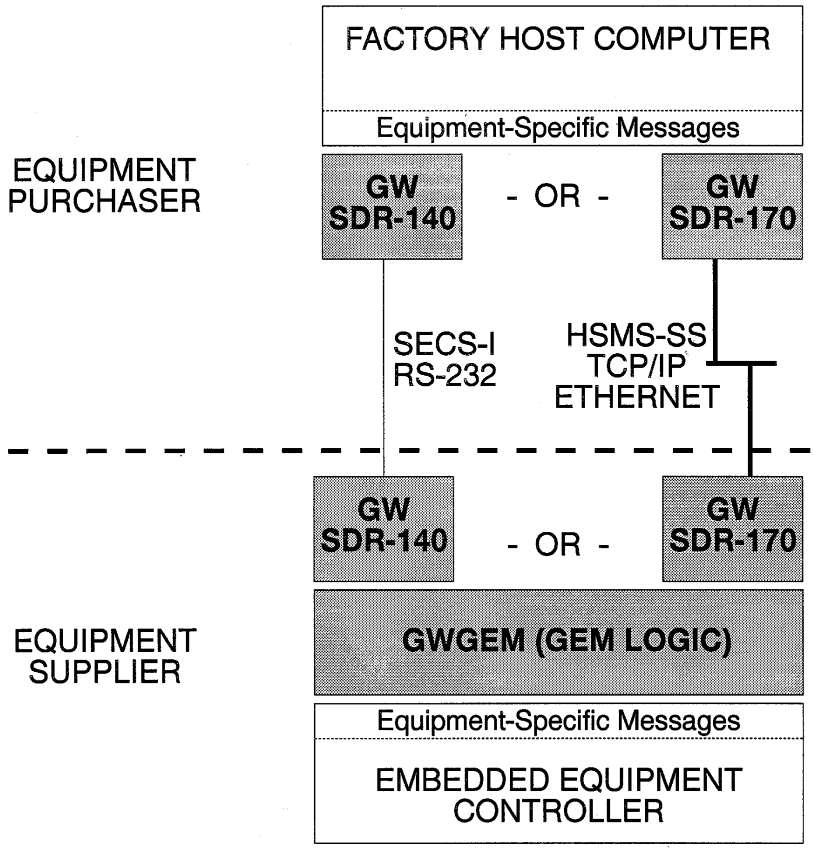


Figure 12 Easy Conversion Between SECS-I and HSMS

COMPUTER AND OPERATING SYSTEM PLATFORMS

To ensure that HSMS-SS was practical, the HMTF spent significant effort building “reference implementations” of HSMS-SS, which were “quick and dirty” software modules, and testing them on a variety of computer and operating system “platforms.” Several important lessons were learned, especially in the area of error recovery, and the standard was changed accordingly. Subsequently, several suppliers have implemented more robust “productized” HSMS-SS software. Figure 13 shows the platforms our company has tested for HSMS-SS. It was found that HSMS-SS can work on all of them. However, it was discovered that different implementations of TCP/IP differ significantly in their response to certain error conditions and in the sustained HSMS-SS transmission speed which can practically be attained. SEMI/Japan recently completed an HSMS-SS Inter-operability Test between two different implementations. Some minor problems were identified and corrected, and the test went well. In conjunction with this STEP/HSMS, SEMI is presenting an HSMS Inter-operability Test involving several suppliers.

Computer	Operating System
IBM PC	MS-DOS Extended MS-DOS Windows NT OS/2 UNIX QNX
Sun SPARC	SunOS Solaris
DECstation MIPS	ULTRIX OSF/1
IBM RS/6000 PowerPC	AIX
HP9000	HP/UX
Silicon Graphics	IRIX
Motorola 88000	UNIX
Motorola 68000	OS-9 VxWORKS
DEC VAX	OpenVMS
DEC Alpha	OpenVMS

Figure 13 HSMS-SS Reference Platforms

FIELD EXPERIENCE WITH HSMS-SS

HSMS-SS is in the early stages of acceptance by the industry. GW Associates offered its first commercial release of HSMS-SS (based on preliminary standards drafts) in 1993, and has worked with several companies in the United States, Europe, and Asia to implement HSMS-SS on a variety of different equipment controllers and host computers.

One area of interest in HSMS-SS is for functional chip test equipment. Almost all types of semiconductor manufacturing equipment now offer SECS communications, with the single (and important) exception of functional test equipment, which requires large data transfers, for which SECS-I was too slow. Today, most functional testers use Ethernet, but each supplier uses his own proprietary protocols, causing difficulty for factory integrators. Strong interest is being seen in converting functional testers to a common HSMS-SS standard. Also, great interest is apparent from metrology equipment suppliers, where larger wafers and smaller feature size has increased the volume of data. Another interesting use of HSMS-SS is in the manufacturing of printed circuit boards using surface mount technology. In that branch of the microelectronics industry, GW Associates has helped deploy several types of embedded equipment controllers and host computers, incorporating HSMS-SS, SECS-II, and GEM protocols, and the factories have recently gone into full production.

GW Associates has run a variety of performance measurements for HSMS-SS. SECS-I is based on RS-232 and is typically run at 9600 bits per second. In contrast, HSMS-SS is based on TCP/IP, and is typically run on Ethernet at 10 megabits per second. In theoretical bandwidth, HSMS-SS is over 1000 times the speed of SECS-I. GW Associates’ experience is that typical implementations of both protocols achieve significantly lower sustained transmittal speeds, perhaps 250 bytes/second for SECS-I and 50,000 bytes/second for HSMS-SS. Still, HSMS-SS achieves about 200 times the speed of SECS-I, with significant differences on different platforms.

The non-deterministic nature of Ethernet's Carrier Sense Multiple Access with Collision Detection (CSMA/CD) can cause performance degradation in heavily loaded networks. In all applications seen so far, the factory Ethernet network has been so lightly loaded that plenty of growth exists for the future. However, network loading should be carefully monitored in factories where message traffic is heavy.

TESTING HSMS-SS

For testing older SECS-I interfaces, commercially available testers such as SECSIM are widely used to display SECS message traffic and diagnose various errors. For HSMS-SS, similar tools such as SECSIM/Pro and SDRSIM testers are used (Figure14). In this way, an equipment supplier can use the same test procedures and the same files of SECS Message Language (SML) language SECS-II test messages to test either SECS-I or HSMS-SS.

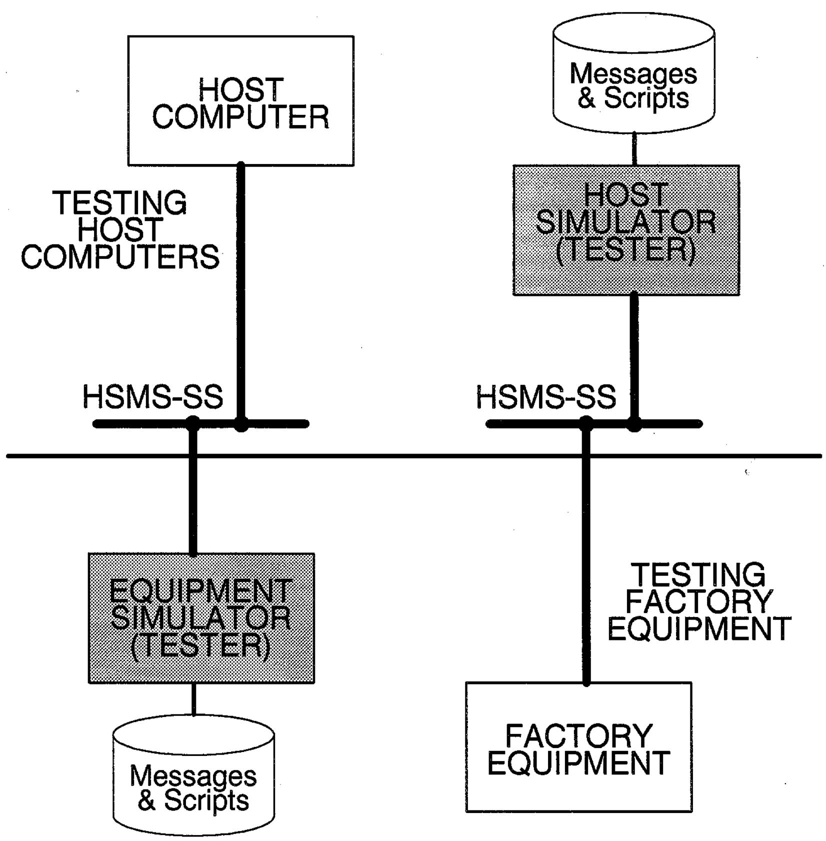


Figure 14 Testing HSMS-SS Links

SHARING THE NETWORK WITH NON-HSMS PROTOCOLS

On most platforms, TCP/IP supports a variety of higher level protocols, all of which can share the TCP/IP communication link (see Figure 15). This works seamlessly and can be very convenient. For example, Equipment and Host might find it convenient to use HSMS-SS SECS messages to trigger the transfer of a large data file using a TCP/IP protocol such as File Transfer Protocol (FTP).

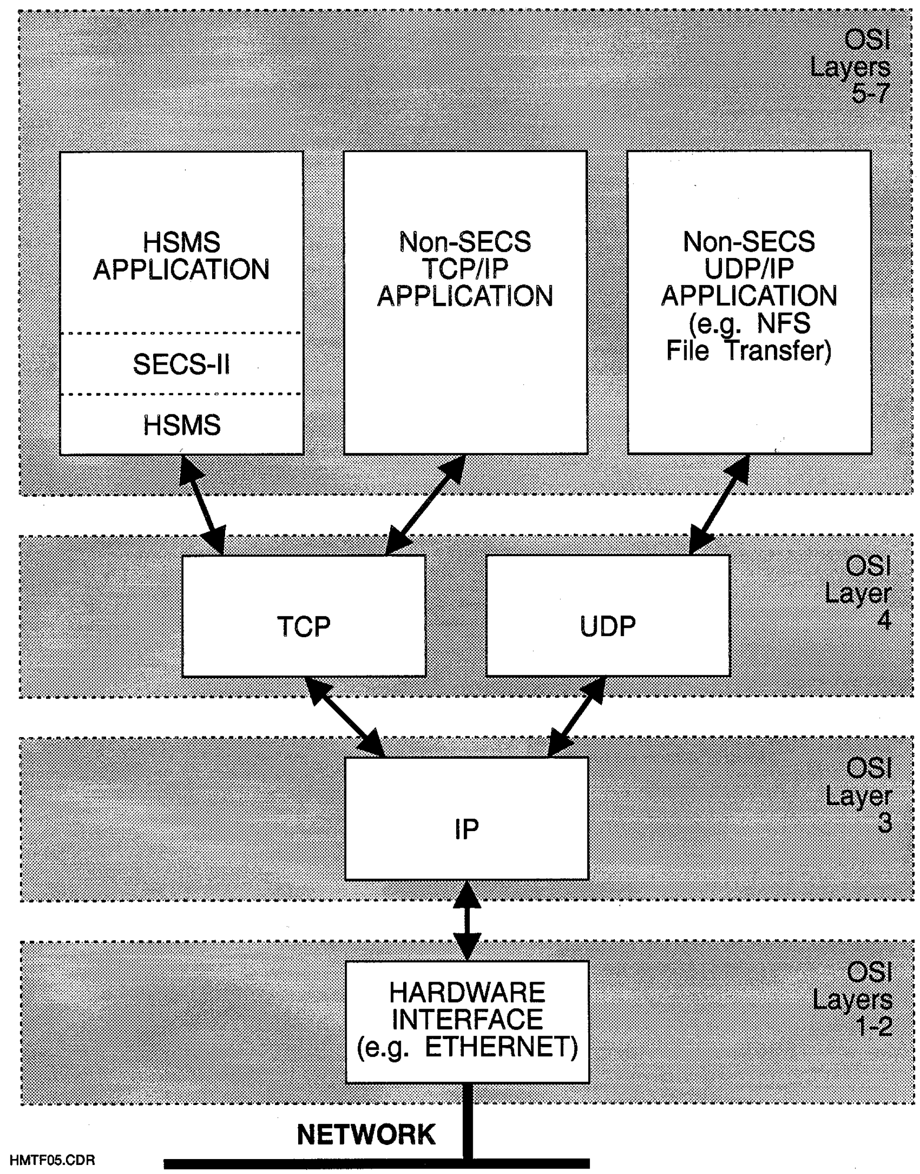


Figure 15 HSMS Can Share Network with Other TCP/IP Protocols

REFERENCES

1. SEMI Standard E4—*SEMI Equipment Communication Standard 1, Message Transport (SECS-I).* Semiconductor Equipment and Materials International Mountain View, CA 94043.
2. SEMI Standard E5—*SEMI Equipment Communication Standard 1, Message Content (SECS-II).*
3. SEMI Standard E13—*SECS Message Services (SMS).*
4. SEMI Standard E30—*Generic Model for Communications and Control of SEMI Equipment (GEM).*
5. SEMI Standard E37—*High-Speed SECS Message Services, Generic Services (HSMS).*
6. SEMI Standard E37.1—*High-Speed SECS Message Services, Single Session (HSMS-SS).*
7. SEMI Standard E37.2—*High-Speed SECS Message Services, General Session (HSMS-GS).*

APPENDIX D
IMPLEMENTING GEM/SECS-II WITH HSMS ON FACTORY EQUIPMENT

by
Thomas J. Dinnel
Software Engineer
Universal Instruments Corporation

Topics Discussed in this Presentation

1. What are SECS-I and HSMS?
2. Why HSMS?
3. Additional Factory Equipment Requirements for HSMS
4. Summary

What are SECS-I and HSMS?

- SECS-I
  - Serial (usually RS-232) point-to-point data path.
  - Configuration includes:
    a. Description of the physical connector.
    b. Signal levels
    c. Data rate.

HSMS

- Network TCP/IP protocol using a logical connection (i.e., sockets, TLI)
- HSMS is still a point-to-point connection, but the connection is a logical connection over the network.

Why HSMS?

- Communications speed advantage
  1. During a Process Program transfer.
  2. When many events (machine status messages) occur at once, such as when a product is finished and lots of data related to the product is sent.

Flexibility of the physical placement in the factory.

No physical limitation (maximum length of the cable)

- All the benefits of a Network
- Network File System (NFS), File Transfer Protocol (FTP), Telnet

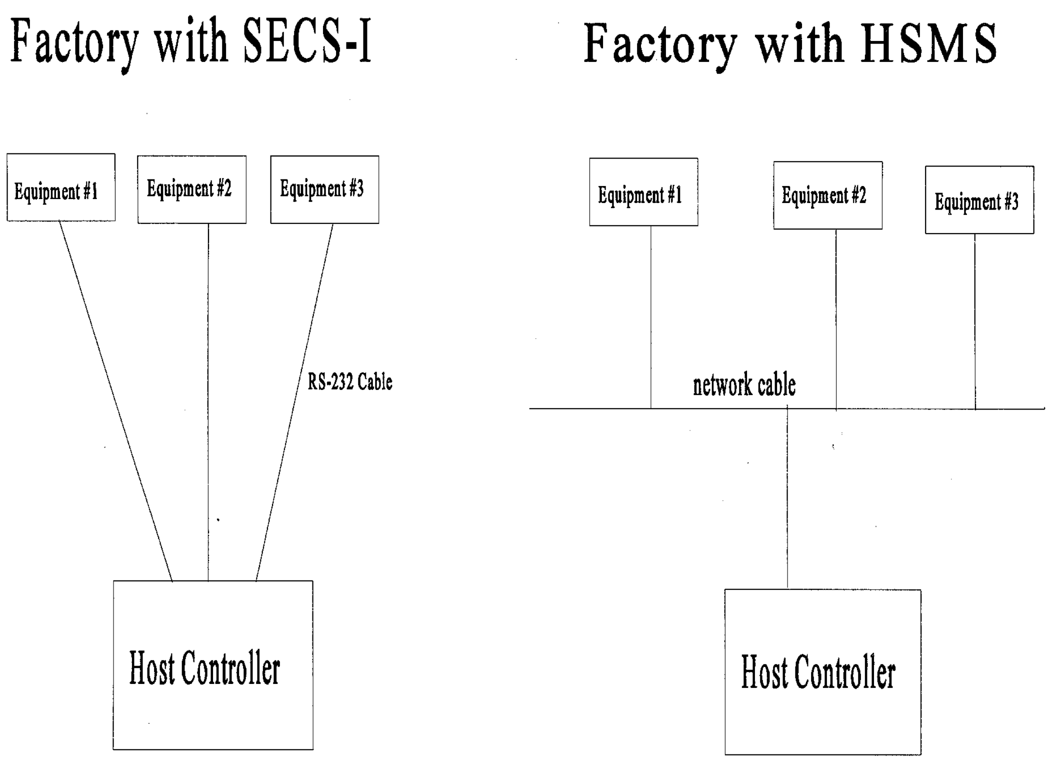


Figure 16 Factory Configurations: SECS-I versus HSMS

Additional Factory Equipment Requirements for HSMS.

To implement HSMS on factory equipment the following five areas need to be addressed:

1. Support of either SECS-I or HSMS on the factory equipment.
2. TCP/IP software and network card.
3. HSMS driver software.
4. Factory equipment “User Interface” changes
5. Field service and installation training issues

1) Support Either SECS-I or HSMS on the Factory Equipment.

Factory requirement is that both the host controller and factory equipment support the same protocol, either SECS-I or HSMS.

The GEM/SECS-II commands remain the same regardless of the SECS-I or HSMS protocol in use.

a) customer orders GEM/SECS-II with SECS-I

- Install a Serial RS-232 card.
- Install the SECS-I level drivers software.
- Install the GEM/SECS-II software.

b) customer orders GEM/SECS-II with HSMS

- Install a network card.
- Install TCP/IP software.
- Install the HSMS level drivers software.
- Install the GEM/SECS-II software.

Note: GEM/SECS-II software remains the same while all the lower levels of hardware and software are changed based on the customer order.

2) TCP/IP Software and Network Card.

Factory equipment must support

TCP/IP software

Network card (usually an Ethernet card)

3) HSMS Driver Software

HSMS is a layer of software below the SECS-II layer that communicates to the TCP/IP software.

HSMS replaces the SECS-I level software. HSMS communication is the exact same as SECS-I to the SECS-II level software.

Both SECS-I and HSMS exchange SECS-II encoded messages between the host and factory equipment.

A given HSMS implementation must match the intended hardware and software platform.

HSMS uses TCP/IP stream support, which provides reliable two way simultaneous transmission of streams of contiguous bytes.

The HSMS driver software can be purchased off the shelf from some companies that specialize in supporting SEMI standard software.

Table 3 Communications Levels

Level 4	Process equipment-specific software
Level 3	GEM Capabilities
Level 2	SECS-II commands
Level 1	SECS-I or HSMS transfer protocols

4) Factory Equipment "User Interface" Changes

Allow the operator the ability to display and change the Network IP address of the equipment.

Note: From the operator's point of view, everything else on the factory equipment will look and work the same with HSMS as with SECS-I.

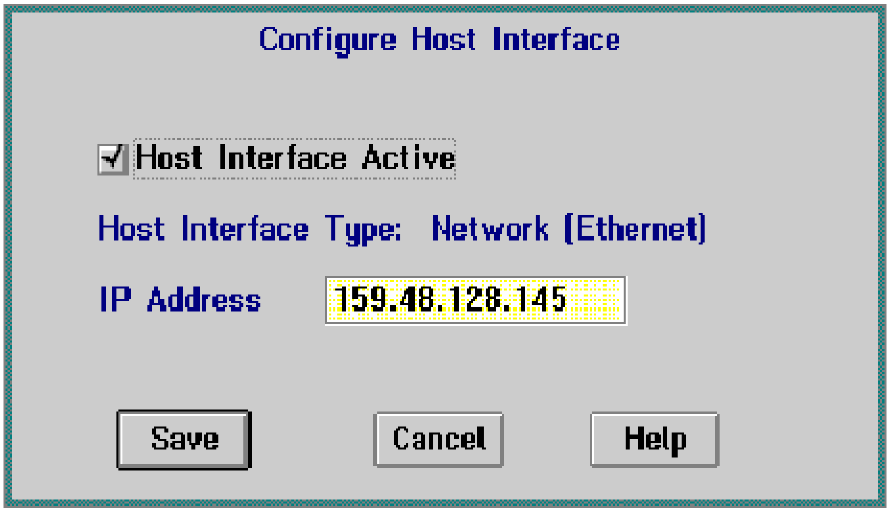


Figure 17 Equipment User Interface Example

5) Field Service and Installation Training Issues

- Whose responsibility is the network?
- Who will connect (and test) the network cable?
- Who will assign and maintain the network address?
- Who will troubleshoot the network connection if it does not work?
- Training of Field Service personnel

Summary

- HSMS is more difficult to develop/install/debug than SECS-I.
- HSMS has more benefits for the customer (host).
- HSMS is required on the equipment if the host is using HSMS.

SEMATECH Technology Transfer
2706 Montopolis Drive
Austin, TX 78741

http://www.sematech.org