

REFERENCE MANUAL

SYNCHRONOUS COMMUNICATIONS ACCESS METHOD:
GENERAL INFORMATION

SINGER
BUSINESS MACHINES

REFERENCE MANUAL

SYNCHRONOUS COMMUNICATIONS ACCESS METHOD:
GENERAL INFORMATION

PUBLICATION NO. 44-C499-401-00

FEBRUARY 1973

SINGER
BUSINESS MACHINES

2350 WASHINGTON AVE.
SAN LEANDRO, CALIF. 94577

PREFACE

This manual describes the general design of the Singer SCA Module. It is intended to support all synchronous data communications environments discussed in Section 3. Systems analysts and programmers should refer to this manual when designing and writing a User Application Program (UAP) that will interface with the Synchronous Communications Adapter (SCA) Module within a System Ten* computer, and when designing and writing a UAP that will interface with the SCA Module for a remote computer.

A familiarity is assumed with the hardware concepts of System Ten computer, of the System Ten Assembly Language, and of the System Ten Synchronous Communications Adapter.

The following manuals are pertinent:

Assembler I Reference Manual (524-700715-M30)
(optional for operators)

or

Assembler II Reference Manual (40-279)
(optional for operators)

Programmer's Reference Booklet (40-101R)
(optional for operators)

DMF Multipartition Loader Reference Manual (40-308)

or

Multipartition Loader (Card) Reference Manual (40-309)

SCAM Multipoint User Application Programming Guide (40-337)

*A trademark of the Singer Company

TABLE OF CONTENTS

Section 1	INTRODUCTION	
Section 2	GENERAL DESCRIPTION	
	Introduction.....	2-1
	SCAM Functions.....	2-1
	User Application Program Function.....	2-3
	Common.....	2-3
	Design and Assembly.....	2-4
Section 3	SCAM ROUTINES	
	Introduction.....	3-1
	Environment-Specific Routine.....	3-2
	Communications Control System.....	3-3
	Initialization Routine.....	3-3
	Get-Buffer Routine.....	3-5
	Error Routine.....	3-7
	Read Routine.....	3-10
	Write Routine.....	3-11
Section 4	INTERPARTITION INTERFACE	
	Introduction.....	4-1
	The Communications Control Block.....	4-2
	Partition Status Byte.....	4-15
	The System Mailbox.....	4-18
	Initialization Record.....	4-21
	Transmit and Receive Buffer.....	4-23
	Idle Message Buffer.....	4-32
	↑CWA and ↑PSWA.....	4-34
Section 5	ASSEMBLING THE SCA PROGRAM	
	Introduction.....	5-1
	Core Requirements.....	5-1
	Initial Values of Common.....	5-4
	Choosing the Housekeeping Routine.....	5-7
	Assembly Considerations.....	5-9

TABLE OF CONTENTS

Section 6 LOADING THE SCA PROGRAM

Using the DMF Multipartition Loader.....	6-1
Using Card Loaders.....	6-4

Appendix A LINE CONTROL CHARACTERS

Appendix B 4310 SCA MODULE

Section 1

INTRODUCTION

INTRODUCTION

This manual describes the general design of the Singer SCA Module for all supported synchronous data communications environments. Systems analysts and programmers should use this manual for reference while designing and writing:

- a User Application Program (UAP) that will interface with the SCA Module within a System Ten computer.
- a UAP that will interface with the communications access method in the remote computer.

To provide the reference information necessary to accomplish the above tasks, the sections of this manual do the following:

Section 2 provides a general description of the overall operation of the SCA Module.

Section 3 provides a more detailed description of each of the component routines in the SCA Module.

Section 4 discusses use of the various fields in the Common deck, and how they are used for interpartition interface.

Section 5 discusses how to assemble an SCA Module and UAP.

Section 6 discusses several alternate forms of loading the SCA Module and UAP.

Section 2

GENERAL DESCRIPTION

INTRODUCTION

SCAM FUNCTIONS

USER APPLICATION PROGRAM FUNCTION

COMMON

DESIGN AND ASSEMBLY

GENERAL DESCRIPTION

INTRODUCTION

The Synchronous Communication Access Method (SCAM) controls high-speed transmission or receipt of files of data over synchronous communication lines by a System Ten computer equipped with a Synchronous Communication Adapter (SCA). Using SCAM, a System Ten can communicate over leased or dial lines. It can communicate with:

- One or more other System Ten computers equipped with SCAs,
- Other manufacturers computers,
- Non-intelligent terminals such as visual display units, printers, tape drives, etc., or
- A combination of different types of computers and terminals.

SCAM FUNCTIONS

SCAM simplifies the application programmer's job by controlling the communication link. For example:

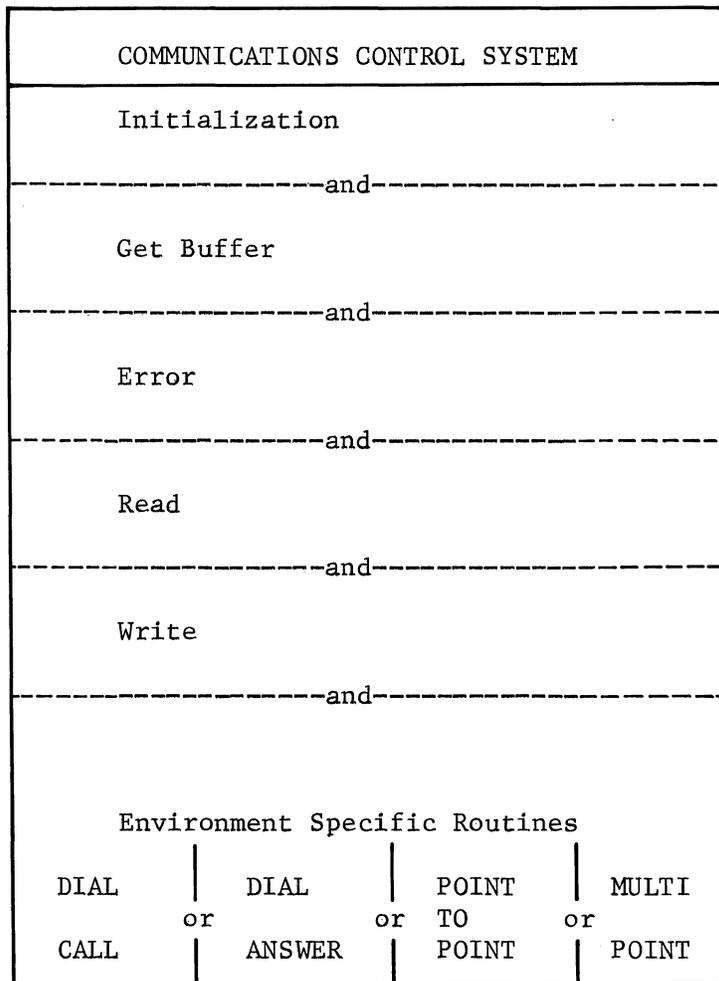
- SCAM prepares the communication line for transmission of data.
- It receives data from the other station and transmits data to the other station. When transmitting, it sends appropriate control characters before and after each data block.
- It sends appropriate control characters at the beginning and end of each data transmission to keep the other station functioning in synchronization with the System Ten.
- It checks for errors and retries when recoverable errors occur.
- It tabulates all errors to help the user identify hardware or line problems.

GENERAL DESCRIPTION

SCAM is distributed as a group of twenty routines. The user selects the routines appropriate for his application and hardware configuration and assembles them together to create a program that will perform the functions described above. This program is called a Synchronous Communication Adapter Module (SCA Module).

A complete data communications handling system consists of three basic categories of software:

USER
APPLICATION
PROGRAMS



The user application program is designed according to the specific needs of the user's system, and therefore must be supplied by the user. The communications control system and environment-specific routines make up the access method that is used by the UAP. These modules are supplied and supported by the Singer Business Machines Company.

USER APPLICATION PROGRAM FUNCTIONS

The user designs and codes a user application program (UAP) to work with the SCA Module. The UAP controls all local input and output and all application-oriented functions, and, to a large extent, it controls the SCA Module. For example:

- The UAP initializes the SCA Module by passing it parameters (e.g., the addresses of the Transmit and Receive Buffers).
- It notifies the SCA Module when it is ready to transmit data, it prepares the actual data to be transmitted, and it indicates when there is no more data to transmit.
- When the other station is transmitting, it processes the received data.
- When a serious (non-recoverable) error has occurred and has been posted by the SCA Module, it indicates when it is ready to begin another transmission.
- It opens and closes all local files, and handles all local I/O, including communication with the operator.

COMMON

The SCA Module is loaded into the partition to which the SCA hardware is attached. The UAP is loaded into any other partition. They communicate with each other through fields in Common. For example:

- The UAP moves initialization information into the System Mailbox, where it is processed by the SCA Module.
- The transmission I/O buffers are in Common. Transmit Buffers are filled with data by the UAP and transmitted by the SCA Module. Receive Buffers are filled by the SCA Module and processed by the UAP.

- There are several status fields in Common in which the SCA Module can post errors and other special conditions, and the UAP can post requests for certain functions.

DESIGN AND ASSEMBLY

The user creates an SCA Module by selecting and assembling the appropriate SCAM routines. The criteria by which the user selects the routines are discussed in Section 3.

The user designs and codes his own UAP. The sample UAPs and "how-to" sections in the appropriate User Application Programming Guide (Point-to-Point, Dial, or Multipoint) will help the user.

A deck of memory definitions for Common (the Common deck) is distributed with the SCAM routines. The user may make minor changes to this deck. The possible changes are discussed in Section 5.

Three housekeeping routines are distributed with the SCAM routines included in the released Common Deck. The first is assembly code and the other two are comments. Choose the one you need (see Section 5-7), take out the others and comments. The user selects one, according to the loader to be used.

The SCA Module and UAP may be assembled separately (each with a copy of the same Common deck) or they may be assembled together. Assembly procedures are discussed in Section 5.

Section 3

SCAM ROUTINES

INTRODUCTION

ENVIRONMENT-SPECIFIC ROUTINE

COMMUNICATIONS CONTROL SYSTEM

SCAM ROUTINES

INTRODUCTION

SCAM consists of twenty source-level routines. The user selects five or six of these routines and assembles them into a single object program, the SCA Module.

SCA Module

The following table lists the SCAM routines and shows the number of versions available for each and the criteria by which the user selects a version for his SCA Module.

ROUTINE	VERSIONS	CRITERIA FOR SELECTION	
Environment-Specific Routines (Handshaking)	6	Type of communications lines and role in communication.	
Communications Control System Routines	Initialization	3	Application
	Get-Buffer	3	Application
	Error	2	Memory positions available
	Read	2	For some applications one of these routines may be omitted.
	Write	2	
	Constants	1	Always needed
Inter-partition interface Common (Includes housekeeping 3 versions)	1	May be modified or user may write his own.	

All SCA modules must consist of one ESR routine and one of each of the Communications control system routines. Some programs, however, will not require both a read and write function if communicating one way only.

ENVIRONMENT-SPECIFIC ROUTINE (ESR)

Note: Handshaking is the exchange of link control characters over the data communications facility for the purpose of synchronizing the data flow between the two terminals. An Environment-Specific Routine performs, among other things such as error recognition and passing data to buffers, certain handshaking functions. Therefore, the routine is an Environment-Specific Routine (ESR) and the function is handshaking.

The ESR, in cooperation with the other station's handshaking logic, establishes the data link between the two stations.

The distributed package contains six versions. The user must choose one for each SCA Module. The choice is based upon two factors:

- The type of data communications network in which the module will operate.
- The role the module will play in controlling the data link between the two stations.

Three types of data communications networks are supported:

- Point-to-Point -- The permanent connection of two stations on a leased or private line. One station is primary and the other is secondary.
- Centralized Multipoint -- The permanent connection of two or more stations on a single leased or private line. One station, designated the control station, can communicate with any of the other (tributary) stations.
- Dial -- The temporary connection of two stations over switched, DDD (dial) lines. One station places the call and the other answers. A calling station can dial more than one answering station, and an answering station can receive calls from more than one calling station.

There is one ESR that implements handshaking conventions for each supported type of communications network.

Version 1: Point-to-Point Primary

The Point-to-Point Primary ESR can initiate the data link by requesting to send data. When contention occurs (when the two stations make simultaneous requests to send data), the primary Handshaking Routine's request is honored first.

Version 2: Point-to-Point Secondary

The Point-to-Point Secondary ESR can initiate the data link by requesting to send data. When contention occurs, the secondary ESR allows the primary station to transmit first.

Version 3: Multipoint Control

The Multipoint Control Handshaking Routine can communicate with any of the tributary stations on the line.

Version 4: Multipoint Tributary

The Multipoint Tributary Handshaking Routine can communicate with the control station but not with the other tributary stations.

Version 5: Dial-Answering

The Dial-Answering Handshaking Routine can receive calls but cannot initiate them.

Version 6: Dial-Calling

The Dial-Calling Handshaking Routine can initiate calls but cannot receive them.

COMMUNICATIONS CONTROL SYSTEM

Initialization Routine

The Communications Control System (CCS) software consists of a variety of different types of programs designed to handle data control tasks within the host computer. They do not directly relate to the external data communications system and therefore are constant regardless of the particular attributes of the network. The CCS group of routines handle functions such as initialization, buffer management, error control, and I/O function control (read or write).

Before data communications can begin, the SCA and user partitions must be loaded and synchronized. To accomplish synchronization, the UAP passes certain parameters (e.g., addresses of the Transmit and Receive Buffers) to the SCA Module. The SCA Module expects the parameters to be in the System Mailbox in the form of an SCA Initialization Record, which is processed by the SCA Initialization Routine. The process of passing parameters to the SCA Module via the System Mailbox is referred to as "initialization." (See

Section 4 for a description of the System Mailbox and the Initialization Record).

The SCAM software package contains three initialization versions. The user must choose one. Version 1 (one-time initialization) will normally be used except when it is necessary to change the initialization parameters without reloading the communication programs (if it is necessary to "reinitialize") after communication has begun.

Reinitialization may be necessary in the following types of applications:

- If different UAPs with different initialization parameters are using the same SCA Module.
- If it is necessary to change the Transmit and Receive Buffer definitions. Possible reasons for changing buffer definitions are discussed under Transmit and Receive Buffers in Section 4.
- If the System Ten is the control station in a multipoint network and it is necessary to change the list of ID's by which the System Ten polls the tributary stations. This "polling list" and the reason it might be changed are discussed in the Multipoint User Application Programming Guide.

The Initialization Routine is always entered from the ESR. The first initialization must always take place before communication begins. If Version 2 or 3 of the Initialization Routine is selected, the UAP may reinitialize when the Handshaking Routine again calls the Initialization Routine. The precise point at which the Initialization Routine is re-entered depends on the version of the Handshaking Routine selected and is discussed in the "how-to" sections of the User Application Programming Guides (Point-to-Point, Multipoint, and Dial). Reinitialization is never possible during data transmission. It is normally (but not always) possible when a given transmission has been terminated by an end of transmission control character or by a serious error.

Version 1: One-Time Initialization

It is strongly recommended that this version be selected, because it is the easiest to use. Normally, reinitialization is unnecessary or can be avoided by careful system design. Initialization is performed once when the SCA Module is first loaded (when the ESR first calls the Initialization Routine). It may not be performed again until the SCA Module is reloaded. This version may

be used if the parameters never change during a given loading of the SCA Module.

Version 2: Optional Reinitialization

Reinitialization may occur whenever the ESR is re-entered. After initialization has been properly performed, when the ESR again calls the Initialization Routine, the Initialization Routine checks the System Mailbox. If it finds a valid Initialization Record (first three characters = 'BCO'), reinitialization takes place. If it does not find a valid Initialization Record in the System Mailbox, it assumes that the last Initialization Record processed is still in effect.

Version 3: Forced Initialization

Reinitialization must occur whenever the ESR is re-entered. When the Initialization Routine is called by the ESR the Initialization Routine will wait for a valid Initialization Record to be placed in the System Mailbox by the UAP.

This version, like Version 2, can be used if the parameters are expected to change while the SCA Module is in memory. If this version is selected, the UAP must move an Initialization Record to the System Mailbox each time the ESR calls the Initialization Routine.

Get-Buffer Routine

This routine checks the status of the user-defined buffers. The Read Routine calls the Get-Buffer Routine when it is ready to read a block of data from the other station and the Write Routine calls it when it is ready to transmit a block of data to the other station. When it is called by the Read Routine, the Get-Buffer Routine checks the status of the Receive Buffers to see if one is ready to receive data. When it is called by the Write Routine, it checks the status of the Transmit Buffers to see if one is ready to transmit to the other station. It then returns to the calling routine either with the address of the buffer that is ready or with a parameter indicating that no buffers are currently ready.

Transmit Buffers are posted "ready to transmit" and Receive Buffers are posted "ready to receive" by the UAP. See Transmit and Receive Buffers in Section 4 for further discussion of buffer status and for the criteria by which the user determines how many buffers he should define.

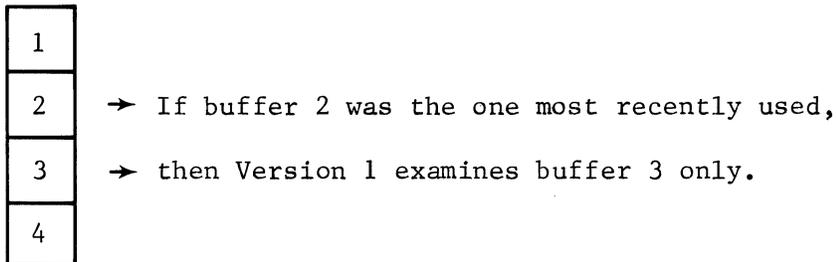
Buffers are defined by the UAP in the Initialization Record. The UAP determines the size, location, and number of buffers. There

may be up to four Transmit and up to four Receive Buffers. If the UAP defines only one Transmit and one Receive Buffer, then any version of the Get-Buffer Routine may be selected -- they will function identically. If the UAP defines more than one Transmit or more than one Receive Buffer, the user's selection of a Get-Buffer Routine version will affect the sequence with which his Transmit Buffers are transmitted and his Receive Buffers are filled. Transmit and Receive Buffers may be defined in the same area of Common.

SCAM contains three versions of the Get-Buffer Routine. The user must choose one for each SCA Module.

Version 1: Sequential

This version should be selected for most standard file transmissions (for transmissions in which the records are to be kept in sequence). Each time Get-Buffer is called, it will examine only one buffer, the buffer immediately following the one most recently used. This version guarantees the user that the buffers are used in their defined order. An example is shown below.



If buffer 3 has been posted "ready" by the UAP, Version 1 passes the buffer's address to the calling routine. If buffer 3 is not ready, then Get-Buffer indicates to the calling routine that no buffers are currently ready.

Version 2: Priority

This version should be selected if the records are to be used according to a priority scheme instead of sequentially. When Get-Buffer is called, it will examine all the buffers in their defined order and return the address of the first one which it finds available. This version effectively attaches a priority to each buffer. The first buffer of a defined set (Transmit or Receive) has the highest priority since the examination always starts with it. An example is shown below.

1	→ This is the buffer with highest priority. Version 2 always examines this buffer first.
2	→ This buffer is examined if the first buffer is not ready.
3	→ This buffer is examined if the second buffer is also not ready.
4	→ This buffer is examined only when all other buffers are found to be not ready.

If buffer 4 is not ready, then Get-Buffer indicates to the calling routine that no buffers are ready.

Version 3: Priority-Sequential

This version provides sequential use of the buffers but assigns priority to the first Transmit Buffer. (The Receive Buffers are filled sequentially, as described under Version 1.) The first Transmit Buffer has the highest priority; the remainder are sequential. There will be one priority and up to three sequential buffers. An example is shown below:

1	→ This first Transmit Buffer has priority status. It is always examined first. The remainder have sequential status.
2	
3	→ If buffer 3 was the buffer most recently used and buffer 1 has been found to be not ready,
4	→ then buffer 4 will be examined next.

If buffer 4 is also not ready, then Get-Buffer indicates to the calling routine that no buffers are ready.

Error Routine

This routine is called by the ESR, the Read Routine, and the Write Routine whenever an abnormal response is received from the other station, or whenever a read or write operation results in an "error" or "fault" status (Condition Code 1 or 4). The SCA Module always retries when there has been an I/O error. The Error Routine counts the number of retries so that an error code can be passed to the UAP after seven retries (after a total of eight unsuccessful attempts to perform the operation). It posts the error code in the Partition Status Byte.

The Error Routine keeps statistics in an Error Table of the number of times an attempt to read or write resulted in an error condition. It keeps separate counts for each possible type of error. Error codes, indicating the error type, are included in the Error Table. The error codes are listed and explained under PSB Codes in Section 4. The error counts may be useful in recognizing and identifying certain hardware problems. They are initialized to zeros each time the SCA Module is loaded; they are not reset until it is reloaded. The SCA Module does not pass them to the UAP. The Error Table can be found in a core dump of the SCA partition at the location of label ↑ETAB in the Error Routine.

The SCAM software package contains two versions. The user must choose one. Version 2 should be selected unless there is not enough memory available for its use.

Version 1 (abbreviated error)

This version tallies the total number of errors, by error code. It does not keep separate totals for each of the eight possible attempts to read or write a data block. See Table 3-1 (page 3-4) for the format of the Error Table kept by Version 1.

Version 2 (full error statistics)

This version totals, by error code, the number of errors for each of the eight possible attempts to read or write. See Table 3-1 (page 3-4) for the format of the Error Table kept by Version 2.

Write and Read Counts

Two additional counts will help the user evaluate the significance of the error counts:

- **Write Count:** The total number of data blocks the Write Routine attempted to transmit. (Note: Although a data block may be retransmitted because of errors, only the first attempt is included in this count.) The Write Count can be found in a core dump of the SCA partition at the location of label ↑WRCNT immediately following ↑ETAB.
- **Read Count:** The total number of times the Read Routine attempted to read a data block. (Note: Although a data block may be reread because of errors, only the first attempt is included in this count.) The Read Count can be found in a core dump of the SCA partition at the location of label ↑RDCNT immediately following ↑WRCNT.

TABLE 3-1. ERROR TABLE (ERROR ROUTINE VERSIONS 1 and 2)

VERSION 1

STARTING LOCATION	Error Code	Error Count
↑ETAB	A	NNNN
↑ETAB+6	C	NNNN
↑ETAB+12	D	NNNN
.	.	.
.	.	.
.	.	.

VERSION 2

The attempt (Read or Write) that resulted in an error.

STARTING LOCATION	Error Code	Error Count							
		1	2	3	4	5	6	7	8
↑ETAB	A	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN
↑ETAB+33	C	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN
↑ETAB+66	D	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN
.
.
.
.

ERROR TABLE KEY

FIELD	POSITIONS	MEANING
Error Type	1	The error code being tallied. Note: See "Partition Status Byte" in Section 4 of this chapter for a description of the error codes tallied.
Error Count (Version 1)	4	The total number of errors.
Error Count 1 (Version 2)	4	The number of times a first attempt to read or write a data block resulted in an error.
Error Count 2 (Version 2)	4	The number of times a second attempt to read or write a data block resulted in an error.
.	.	.
.	.	.
.	.	.
Error Count 8 (Version 2)	4	The number of times the last attempt to read or write a data block resulted in an error. (The number of times an error was posted to the UAP).

Error codes and their meaning may be found on page 4-3 (Inhibit Switch Codes) and on pages 4-15 and 4-16 (Partition Status Byte Codes). When an "E" is placed in the error code area, it takes the meaning on Page 4-15, rather than the Inhibit Switch meaning.

Read Routine

The Read Routine reads blocks of data from the other station. It calls the Get-Buffer Routine to get the address of a Receive Buffer that has been posted ready to receive data by the UAP. If there is a buffer ready, it attempts to read data from the other station. (Note: maximum block size is passed to the SCA module in the initialization record.)

If there is an error on the read, the Read Routine calls the Error Routine so that error counts can be accumulated. Control returns to the Read Routine, which will retry up to seven times (for a total of eight reads). If the read is still unsuccessful after eight tries, the Error Routine posts an error code in the SCA's Partition Status Byte and control again returns to the Read Routine. (See Partition Status Byte in Section 4 for the error codes and their meanings.) Certain serious errors and the receipt of certain control characters will cause the Read Routine to post a code in the Inhibit Switch and pass control to the Handshaking Routine. (See Inhibit Switch in Section 4.)

If the Read Routine successfully reads data into the Receive Buffer, it updates the Buffer Prefix and again calls the Get-Buffer Routine.

Two versions of the Read routine are provided in the SCAM software.

Version 1 -- Read

This version is used if the remote terminal is another System Ten computer. The remote terminal may also be any other computer (such as a 360 or 370) that uses binary synchronous communications line control conventions. The read module does, however, contain extensive message error detection logic that requires the application program in the remote terminal to provide length and sequence indications. These requirements are detailed elsewhere in this document.

Version 2 -- 4310 Read

This version is used when the remote terminal is a Singer 4310 series Magnetic Data Recording System. It does not contain the length and sequence check features of Version 1. The 4310 Read and Write modules are contained in Appendix B.

The read routine may be omitted from the assembly of the SCA Module if the System Ten computer never receives data from the other station (unless the Dial-Answering version of the ESR has been selected: a Dial-Answering SCA Module must contain a Read Routine, even if it doesn't receive data).

Write Routine

The Write Routine transmits data blocks to the other station. It calls Get-Buffer to get the address of a Transmit Buffer that is ready to be transmitted. If a buffer is ready, it creates and formats control information in the Buffer Prefix and then attempts to transmit the data block in the buffer. (Note: in the control information, it receives the size of each block.)

If there is an error on the write, the Write Routine calls the Error Routine so that error counts can be accumulated. Control returns to the Write Routine, which will retry up to seven times (for a total of eight writes). If the write is still unsuccessful after eight tries, the Error Routine posts an error code in the SCA's Partition Status Byte, and control again returns to the Write Routine. (See Partition Status Byte in Section 4 for the error codes and their meanings.) Certain serious errors and the receipt of certain control characters will cause the Write Routine to post a code in the Inhibit Switch and pass control to the ESR. (See Inhibit Switch in Section 4.)

If the Write Routine successfully transmits the data block, it again calls the Get-Buffer Routine.

Two versions of the Write routine are provided in the SCAM software.

Version 1 -- Write

This version is used if the remote terminal is another System Ten computer. The remote terminal may also be any other computer (such as a 360 or 370) that implements binary synchronous communications line control conventions. The write module sends length and sequence indicators, to enable error checking by the read module in the remote terminal.

Version 2 -- 4310 Write

This version is used when the remote terminal is a Singer 4310 series Magnetic Data Recording System. It does not send length and sequence indicators. It does send an STX-ENO pair instead of the IDLE message that the System Ten Write module sends. Further, it provides for device selection. See Appendix B.

The Write Routine may be omitted from the assembly of the SCA Module if the System Ten never transmits data to the other station (unless the Dial-Calling Version of the Handshaking Routine has been selected: a Dial-Calling SCA Module must contain a Write Routine, even if it doesn't transmit data).

Section 4

INTERPARTITION INTERFACE

INTRODUCTION

THE COMMUNICATIONS CONTROL BLOCK

PARTITION STATUS BYTE

THE SYSTEM MAILBOX

INITIALIZATION RECORD

TRANSMIT AND RECEIVE BUFFER

IDLE MESSAGE BUFFER

↑CWA AND ↑PSWA

INTERPARTITION INTERFACE

INTRODUCTION

The SCA Module resides in the SCA partition of the System Ten computer while the user application program is in another partition. The UAP and the SCA Module communicate with each other via seven areas in Common, listed below. The first four areas are always used. The last three areas are used only when the indicated version of the ESR is selected.

- Communications Control Block
- Partition Status Byte
- System Mailbox
- A Set of Transmit and/or Receive Buffers
- IDLE Message Buffer (Point-to-Point and Dial & Answer only) except 4310 Read and Write routines.
- Call Work Area (Dial-Call only)
- Poll/Select Work Area (Multipoint Tributary only)

Inhibit Switch Codes Set by the UAP

The UAP may wish to inhibit communications for application-determined reasons. Since the SCA Module uses alphabetic codes to inhibit communications, the UAP should use numeric codes for the same purpose. A user-set code in the Inhibit Switch will be recognized only by the ESR. The ESR is re-entered when control leaves the Read or Write Routine. The only time the SCA Module exits its Read or Write Routine without first posting a code in the Inhibit Switch is when the UAP requests in the Read Flag or the Write Flag that an end of transmission control character (EOT or DLE EOT) be sent when there is no more data to send or when an error condition makes it imperative to stop. In this case, the UAP can inhibit communications by posting a numeric code in the Inhibit Switch before it posts its EOT or DLE EOT request. (In a dial system, posting a non-zero Inhibit Switch code after posting the Read Flag will cause the SCA Module to hang up.)

Inhibit Switch Codes

Table 4-1 shows the possible Inhibit Switch codes, the routines that post them, and their meanings. The meaning of each code is explained more fully in a listing of codes, by routine, after the table.

TABLE 4-1: INHIBIT SWITCH CODES

<u>Code</u>	<u>Set By</u>	<u>Meaning</u>
0	UAP	OFF -- Communication may proceed.
1-9	UAP	ON -- Inhibit communication.
B	Dial-Call Dial-Answer	DLE EOT received. Line disconnect.
C	Read	Timeout when trying to read data.
E	Initialization	Initialization error.
G	Read	Invalid response to WACKs
I	Mpt Ctl	Invalid response to select sequence.
I	Mpt Trib	Error on first text read.
K	Mpt Trib	UAP failed to respond after being addressed.
M	Mpt Ctl	Invalid response to poll sequence.

INTERPARTITION INTERFACE

Ø	Dial-Call Dial-Answer	EOT or DLE EOT received during handshaking immediately after call is placed.
P	Dial-Call Dial-Answer	Error in handshaking immediately after call is placed
Q	Write	EOT or DLE EOT received while transmitting.
Q	Multipoint Control	EOT received in response to select sequence.
R	Write	No Transmit Buffers ready.
W	Read	EOT received at end of data.
X	Dial-Call Dial-Answer	EOT received during handshaking.
Y	Read	DLE EOT received at end of data.
Z	Dial-Call Dial-Answer	Error in handshaking.

Note: Point-to-point routines never set the Inhibit Switch.

User Application Program Inhibit Switch Codes

<u>Code</u>	<u>Meaning</u>
0	OFF. Handshaking, under the control of the ESR, may proceed normally. The UAP must always reset the Inhibit Switch to zero after it has been set on by either the UAP or the SCA Module before communication can continue.
1-9	ON. The ESR is not to attempt to establish a connection with the other station. If the SCA Module is in the process of communicating with the other station (reading data, writing data, or handshaking with the other station), communication will continue until the ESR is re-entered. Only then will the SCA Module detect the UAP's code.

Initialization Routine Inhibit Switch Code

<u>Code</u>	<u>Meaning</u>
E	The Initialization Routine has found a format error in the parameters of the Initialization Record.

In addition to setting the Inhibit Switch to E, the SCA Module moves zeros to ↑ECB+4,+5 (Partition that Last Initialized).

Processing will continue if the UAP moves a correct Initialization Record to the Mailbox and resets the Inhibit Switch to zero. However, normally it is necessary to analyze the program and perhaps reassemble to correct the problem. In most cases the UAP should reset the Mailbox to available (move an A to the Mailbox Status Byte), print an error message, request a memory dump, and terminate.

Read Routine Inhibit Switch Codes

<u>Code</u>	<u>Meaning</u>
C	Eight consecutive attempts to read data from the other station have terminated with "fault status" (condition code 4 - ten-second timeout). A C is also posted in the PSB. The cause of the error is described more fully under PSB code C.

- G Invalid response to WACKs. The Read Routine is unable to find a Receive Buffer ready to be filled. It is sending WACKs to the other station. The Read Routine has been unable to read a correct response (an ENQ) to eight consecutive WACKs.
- W EOT received. The other station has sent an EOT (end of transmission) control character, indicating that it has completed its transmission of data.
- Y DLE EOT received. The other station has sent a DLE EOT control character, indicating that it has completed its transmission of data. A DLE EOT has the same meaning as an EOT except that in a dial system it means that the line is being disconnected. In a point-to-point or multipoint system, a DLE EOT may be sent instead of an EOT to indicate abnormal end of transmission.

Write Routine Inhibit Switch Codes

<u>Code</u>	<u>Meaning</u>
Q	EOT or DLE EOT received. The other station has sent an EOT or DLE EOT response to the last block transmitted. For some reason (probably a hardware failure), it does not want to receive any more data at this time.

CAUTION: The last block of data sent may or may not have been properly received by the other station. It is possible that it was lost.

R	The Get-Buffer Routine has been unable to find a Transmit Buffer ready to be sent to the other station.
---	---

If a Dial or Point-to-Point ESR has been selected, the Write Routine has tried 50 times to find a ready Transmit Buffer. When it receives a "no buffers ready" indication from the Get-Buffer Routine, it sends an ENQ to the other station, and then it calls Get-Buffer again. When it has sent five consecutive ENQs, it sends an IDLE Message, and then it calls Get-Buffer again. When it has sent ten IDLE Messages (50 ENQs), it sends an EOT and then posts an R in the Inhibit Switch. 4310 Write Routine does not send ENQ's and IDLE, instead it sends STX ENQ ten times before sending EOT and posting R in the Inhibit Switch.

If a Multipoint ESR has been selected, the Write Routine has tried five times to find a ready Transmit Buffer. Multipoint SCA Modules do not send IDLE

Messages. The Write Routine calls Get-Buffer five times (and send five ENQs) before it sends an EOT and posts an R in the Inhibit Switch.

The other station's responses to these ENQs and IDLE Messages are not checked for errors.

Multipoint Control ESR Inhibit Switch Codes

<u>Code</u>	<u>Meaning</u>
I	A tributary station has sent an invalid response to a select sequence. The SCA Module tries twice to select the tributary station before it posts this code. After posting the I, the SCA Module sends an EOT to place the line in control mode.
M	A tributary station has sent an invalid response to a polling sequence. The SCA Module tries twice to poll the tributary station before it posts this code. After posting the M, the SCA Module sends an EOT to place the line in control mode.
Q	EOT received in response to a select sequence. The selected station is unable to receive data at this time.

Multipoint Tributary ESR Inhibit Switch Codes

<u>Code</u>	<u>Meaning</u>
I	Error while reading first data block. The tributary station has been selected and has responded with an ACKO. It has tried twice to read the first data block, and both attempts have resulted in an error condition. The PSB is set to A, C, or D. See PSB codes A, C and D for further description of the specific errors that can cause an I to be posted.
K	The tributary UAP failed to reset the Poll/Select Status Byte to A or Q within ten seconds after the station was polled or selected. The tributary SCA Module sends an EOT to the control station after it posts this code.

Dial-Answer ESR Inhibit Switch Codes

Codes \emptyset and P are posted immediately after the answering station is called by the calling station. They represent errors occurring when the answering station is trying to read an ENQ (or an ID and an ENQ if there is ID checking) from the calling station.

<u>Code</u>	<u>Meaning</u>
B	DLE EOT received. The SCA Module has just exited its Write Routine and it is the other station's turn to transmit data. The other station has sent a DLE EOT (and disconnected the line) instead of sending data.
\emptyset	The calling station sent a DLE EOT instead of an ENQ or ID ENQ. This error should not occur. It's as if the caller said goodbye instead of hello when his call was answered. The answering SCA Module hangs up in dismay after it posts the code \emptyset .
P	The calling station sent an invalid response or no response. The errors discussed under PSB codes A, C, and M could cause this code to be posted in the Inhibit Switch. If it is one of these errors, the SCA Module tries eight times to read an ENQ or an ID and an ENQ before it posts the P, and it also posts a code (an A, C, or M) in the PSB.

A P is also posted if the calling station sends an invalid ID (one not included in the UAP's WRU list in the Initialization Record) or no ID when ID checking is to be performed. If an invalid ID was received, the SCA Module places it in relative locations 8 - 13 of the CCB. (See Invalid Dial ID in this section.)

In either case, the SCA Module hangs up after it posts the P.

Codes X and Z are posted during handshaking after the answering SCA Module has entered its Read Routine at least once.

<u>Code</u>	<u>Meaning</u>
X	EOT received by the ESR. The calling station has sent an EOT (end of transmission) instead of data. It has no data to transmit. The answering station may still transmit data.
Z	Invalid response. The SCA Module has just sent an ENQ and is expecting an ACKO response (the UAP has data to

send); or it is the calling station's turn to transmit and the SCA Module is expecting an ENQ or EOT. The SCA Module has tried eight times to read the correct response from the calling station.

If the error occurred while trying to read an ACKO, the code A, C, or I will be posted in the PSB. If the error occurred while trying to read an ENQ or EOT, the code A, C, or M will be posted in the PSB. See PSB codes A, C, I, and M for descriptions of the specific errors that may cause a Z to be posted in the Inhibit Switch.

The SCA Module sends a DLE EOT to the calling station and hangs up after posting the Z.

Dial-Call SCA Module Inhibit Switch Codes

Codes \emptyset and P are posted immediately after the calling station has placed a call. They represent errors occurring when the calling station is trying to read an ACKO or ID and ACKO in response to its first ENQ.

<u>Code</u>	<u>Meaning</u>
B	DLE EOT received. The SCA Module has just exited its Write Routine and it is the other station's turn to transmit data. The other station has sent a DLE EOT (and disconnected the line) instead of sending data.
\emptyset	The answering station sent an EOT instead of an ACKO or ID and ACKO. This error should not occur. It is as if the answering station said goodbye instead of hello when it answered the call. The calling SCA Module hangs up after it posts the code \emptyset .
P	The answering station sent an invalid response or no response. The errors discussed under PSB codes A, C, I, and S could cause this code to be posted. If it is one of these errors, the SCA Module tries eight times to read an ACKO or an ID and an ACKO before it posts the P, and it also posts a code (an A, C, I, or S) in the PSB.

A P is also posted if the answering station sends an invalid ID (one not included in the UAP's WRU list in the Initialization Record) or no ID when ID checking is to be performed. If an invalid ID is received, the SCA

Module places it in relative locations 8 - 13 of the CCB. (See Invalid Dial ID in this section.)

In either case, the SCA Module hangs up after it posts the P.

Codes X and Z are posted during handshaking after the calling SCA Module has entered its Write Routine at least once.

<u>Code</u>	<u>Meaning</u>
X	EOT received by the ESR. The answering station has sent an EOT (end of transmission) instead of data. It has no data to send. The calling station may still transmit data.
Z	Invalid Response. The SCA Module has just sent an ENQ and is expecting an ACKO response (the UAP has data to send); or it is the answering station's turn to transmit and the SCA Module is expecting an ENQ or EOT.

If the error occurred when the SCA Module was trying to read an ACKO, the errors discussed under PSB codes A, C, and I can cause the Z to be posted. The SCA Module tries twice to read the ACKO. It does not post a code in the PSB.

If the error occurred when the SCA Module was trying to read an ENQ or EOT, the errors discussed under PSB codes A, C, and M can cause the Z to be posted. The SCA Module has tried eight times to read a correct response before it posts the code Z. It posts an A, C, or M in the PSB.

The SCA Module sends a DLE EOT and hangs up after posting the Z.

Read Flag (\uparrow ECB+1)

This flag is set ON by the UAP when it wants the other station to stop transmitting data. By setting the flag ON, the UAP requests that the SCA Module send an EOT, a DLE EOT, or a TI to the other station. The meanings of these transmission control characters are discussed further in Appendix A. The Read Routine resets the flag to zero (OFF) after it sends the appropriate control character, or when, for any reason, it returns control to the ESR.

The Read Flag is tested only by the SCA Module's Read Routine and only when there is a Receive Buffer ready to receive data. (When posting a code in the Read Flag, be sure that the appropriate Receive Buffer is posted ready, even though it is not to be filled with data.)

Read Flag Codes

<u>Code</u>	<u>Meaning</u>
0	OFF. Normal. Receipt of data is to continue normally. The Read Routine resets the flag to zero after it sends the appropriate control character, or when, for any reason, it returns control to the ESR. The flag should be initialized to zero. (See <u>Initial Values of Common</u> in Section 5.)
1	EOT request. The SCA Module is to send an EOT to the other station. An EOT might be sent when the receiving station is unable to receive more data because of a hardware problem. The SCA Module reenters its ESR after it sends the EOT.
2	DLE EOT request. The SCA Module is to send a DLE EOT to the other station. In multipoint and point-to-point networks, the transmission of a DLE EOT has the same results as the transmission of an EOT. However, for some applications, a DLE EOT may be sent instead of an EOT to indicate abnormal end of transmission. In dial systems, the transmission of a DLE EOT causes the line to be disconnected. The SCA Module reenters its ESR after it sends the DLE EOT.
3	TI request. The SCA Module is to send a TI (transmission interrupt) to the other station. The other station should send an EOT to this station and then allow this station to transmit data, but the other station <u>may</u> ignore the TI. Control remains in the Read Routine until an EOT is received.

A TI should be sent when the UAP wants to temporarily interrupt the other station's transmission so that this station can send data. Before designing a system that sends TIs, be sure that the other station recognizes them and responds correctly.

Operation Code (\uparrow ECB+2)

This code is set ON by the UAP when it has data to transmit or (dial networks only) when it wants to disconnect the line.

<u>Code</u>	<u>Meaning</u>
D	ON. Write request posted by the UAP. The UAP has data to transmit to the other station. The SCA Module is to enter its Write Routine.
E	(Dial only.) ON. Disconnect request posted by a dial UAP. The UAP has no data to send and it wants to disconnect the line.
O	OFF. The SCA Module resets the Operation Code to zero when it exits its Write Routine (after it sends an EOT or DLE EOT or after a serious error has occurred). The code should normally be initialized to zero. (See <u>Initial Values of Common</u> in Section 5.)

Point-to-Point Use of the Operation Code

A Point-to-Point SCA Module will never enter its Write Routine unless the UAP has posted a D in the Operation Code (and the Inhibit Switch is OFF). It tests for a D in the Operation Code:

- When it is first loaded (after Initialization),
- Whenever it exits its Read or Write Routine (after the UAP has reset the Inhibit Switch, if an Inhibit Switch Code has been posted),
- And after the other station has sent an EOT indicating it has no data to send.

If the UAP has not posted a D in the Operation Code, the SCA Module sends an EOT (no data to send) to the other station and checks to see if the other station has data to send.

If the UAP has posted a D in the Operation Code, the SCA Module sends an ENQ (write request) to the other station. If the other station responds correctly, it enters its Write Routine. If the other station responds with an ENQ (it also wants to send data), the primary station transmits first. If the secondary UAP does not reset the D in the Operation Code, the SCA Module will detect it, and try again to transmit when the primary station's transmission of data is terminated.

Dial Use of the Operation Code

A dial SCA Module will not enter its Write Routine unless the UAP has posted a D in the Operation Code except in the following instance: The Dial-Calling SCA Module automatically enters its Write Routine immediately after placing a call, regardless of the Operation Code setting.

A dial SCA Module always checks the Operation Code after receiving an EOT terminating a transmission from the other computer (after the UAP has reset the Inhibit Switch to zero). If it finds a D, in the operation code, it enters its Write Routine. If it finds an E, in the operation code, it sends a DLE EOT and disconnects the line. If it finds a zero, it sends an EOT and checks to see if the other station has data to send. When neither station has data to send, the calling SCA Module sends a DLE EOT and disconnects the line.

Multipoint Use of the Operation Code

The Operation Code is not used in a multipoint tributary station. The multipoint tributary UAP uses the Poll/Select Status Byte (described under Poll/Select Work Area in this section) instead of the Operation Code to indicate when it is ready to read or write data.

The Multipoint Control SCA Module checks the Operation Code for a D before it enters its Write Routine. (This is done whenever it has finished polling a station, in order to see if there is a station to be selected.) If it does not find a D, it polls the next station in the polling list.

Activity Switch (\uparrow ECB+3)

The SCA Module posts a code in the Activity Switch to indicate whether it is in its Read Routine, its Write Routine, or its ESR. The UAP can test the switch but it should not alter it.

<u>Code</u>	<u>Meaning</u>
A	The SCA Module is in its Read Routine.
D	The SCA Module is in its Write Routine.
Zero	The SCA Module is in its ESR.

Point-to-Point Use of the Activity Switch

The Activity Switch is zero except when the station is involved in actual data transfer.

Dial Use of the Activity Switch

The Activity Switch is zero from the time the SCA Module is loaded until it establishes the first connection and completes initial handshaking and ID exchange. It is reset to zero when the line is disconnected and remains zero until a new connection is established and initial handshaking and ID exchange is again completed.

Multipoint Control Use of the Activity Switch

The Activity Switch is zero from the time the SCA Module is first loaded until it polls or selects the first station. It contains an A when the SCA Module is polling or reading data. It contains a D when the SCA Module is selecting or transmitting data. It is reset to zero when the line is placed in control mode and remains zero until another station is addressed.

Multipoint Tributary Use of the Activity Switch

The Activity Switch is zero from the time the SCA Module is first loaded until the tributary station begins transmitting or receiving data. It contains a D when the station is transmitting data and an A when the station is receiving data. It is reset to zero whenever the line is placed in control mode. The Activity Switch in a given tributary station is zero when the control station is communicating with other stations on the line and during polling and selection of the tributary. It contains an A or D only when the tributary is involved in actual data transfer.

Partition That Last Initialized or Poll Address (Multipoint Control) (\uparrow ECB+4,5)

When the Initialization Routine finds parameters in the Mailbox, it moves the From Address field from the Mailbox into this field. If more than one UAP is using the SCA Module, a UAP can test this field to see if the initialization parameters currently in effect are appropriate for its use.

If the Initialization Routine finds a format error in the Initialization Record, it sets this field to zeros and the Inhibit Switch to E.

In a multipoint control station the ID of the partition that last initialized will be overlaid in \uparrow ECB+4,5 as soon as the first station is polled. When the Multipoint Control SCA Module polls a station, it places the first two characters of the poll sequence in \uparrow ECB+4,5. When an EOT is received, the SCA Module resets the field to zeros. If an attempt to poll or read data is terminated by a serious error, the field is not reset until the next time a station is polled. The UAP can use \uparrow ECB+4,5 to determine the address of a

tributary that is sending data or of the station with which it was communicating when a serious error occurred. If one-character poll sequences are being used, the UAP should ignore \uparrow ECB+5.

SCA Partition Number (\uparrow ECB+6,7)

When the SCA Module is loaded, it moves its partition number into this field.

The SCA module must be loaded first with Common and with this field initialized to zeros (in the Common Deck definitions). (See Assembling the SCA Module in Section 2 of this chapter.) The UAP is then loaded and must test to ascertain whether or not the SCA Partition has been properly loaded. A valid partition number in the field will indicate to the UAP that the SCA Module has been loaded.

The UAP will find the SCA Partition Status Byte at the address: 560C plus the SCA's Partition number.

Invalid Dial ID (\uparrow ECB+8 - 13)

This field is used by dial SCA Modules only. If an invalid Who Are You (WRU) ID is received after a call has been placed, the dial ESR moves the invalid ID to this field so that the UAP can access it. A P in the Inhibit Switch indicates to the UAP that an invalid ID may have been received. This field is reset to zeros when the next call is placed.

Remainder of CCB (\uparrow ECB+14-17)

The remainder of the CCB is reserved. It must not be used by the UAP.

PARTITION STATUS BYTE

One of the locations 0560-0579 of Common serves as the SCA Module's Partition Status Byte (PSB). The location of the SCA Module's PSB may be found by adding the SCA Module's partition number to 560C. The SCA Module's partition number can be found by the UAP at relative locations 6-7 of the Communications Control Block.

The Read, Write, and ESRs may use the PSB to post status information to the UAP after attempting to perform a read or write operation. Most of the codes posted in the PSB represent errors. Normally before an error is posted in the PSB, the SCA Module has tried eight times to perform the operation.

Read Errors

The SCA Module does not change the Receive Buffer status when there is an error on an attempt to read data. The buffer will still be ready (code A in the Buffer Status Byte), and the Buffer Prefix will not have been updated. The SCA Module will continue to try to read until the UAP resets the Buffer Status Byte.

Write Errors

The SCA Module posts an E (error) in the Transmit Buffer Status Byte after posting a code in the PSB. It makes no change to the remainder of the buffer. The Buffer Prefix will not have been updated, and the data in the buffer will not have been altered. The SCA Module will continue to try to write if the UAP merely resets the Buffer Status Byte to a D except under the following conditions:

- If Version 2 or 3 of the Get-Buffer Routine (Priority or Priority/Sequential) has been selected,
- or if the code posted in the PSB was a T (indicating that a TI was received from the other station).

Under either of these conditions the SCA Module requires that the UAP reset the PSB to N (in addition to setting the Buffer Status Byte to D) before it will continue to try to write.

Resetting the PSB

Although it is not normally required that the UAP reset the PSB before the SCA Module can continue to try to read or write, it is advisable to reset it because the SCA Module will not. The UAP should reset the PSB (to N) so that it can count the number of times a given error has been posted by the SCA Module. The UAP will probably want to branch to its own error routine after receiving an error code some (user-determined) number of times.

Partition Status Byte Codes

The following table shows the PSB codes and the routines which may post them.

TABLE 4-2: PARTITION STATUS BYTE CODES

	A	C	D	E	F	H	I	J	M	N	S	T
Read	X	X	X	X	X					X		
Write	X	X				X		X		X	X	X
Point-to-Point Primary	X	X										
Point-to-Point Secondary	X	X					X	X		X		
Multipoint Control												
Multipoint Tributary	X	X	X							X		
Dial-Answering	X	X					X		X	X		
Dial-Calling	X	X					X		X	X	X	

Following is a list of the codes that may be posted in the PSB. All of the codes posted by the SCA Module, except T and N, represent errors and are therefore actually posted by the Error Routine; the routines listed with these codes are the routines that may branch to the Error Routine for the error type indicated.

Statistics on the number of occurrences of each of the error types (all codes except T and N) can be found in the Error Table (See Error Routine in Section 3.)

Code Meaning

A The most recently performed operation persistently terminated with "error status" (condition code 1) set.

C The most recently performed I/O operation persistently terminated with "fault status" (condition code 4) set. This can be caused by failure of any configuration component.

It usually means:

- The modem's power is off.
- Ten seconds have elapsed without receiving a non-SYN character (time-out).

D A data block sent by the other station was terminated by a DLE pair (other than DLE EOT). The SCA Module expects data blocks sent by the other station to be terminated by either ETX or ETB. This is probably caused by the other station sending an illegal DLE pair.

E Length error. The length of data received does not correspond to the length code sent by the other station. This is caused either by the other station sending the wrong length or by characters being lost and not detected by the hardware-generated VRC/BCC. (Not used by the 4310 Read routine.)

- F Sequence error. The data block received is not in sequence relative to the last block received. Either the sequence number is not being generated properly (by software) or some number of data records have been "lost" during the transmission. (Not used by the 4310 Read routine.)
- H The other station is sending unexpected responses to data. It is likely that the two stations have become unsynchronized due to some unusual error. However it is possible that the other station is sending uninterpretable responses.
- I The other station is sending a response other than ACKO, ENQ, or NAK to this station's ENQ.
- J The other station is sending a NAK response to data or (point-to-point) to an ENQ transmitted by this station. For some reason the other station is unable to read the data or ENQ.
- M The other station is sending a DLE pair when the System Ten is expecting to read an ENQ.
- N Normal. The last I/O instruction completed without error.
- S WACK received. The other station has sent persistent WACK's (waiting characters) in response to the last data block sent or (dial-calling) in response to the ENQ sent by this station after it placed the call. For some reason, possibly that no Receive Buffers are available, the other station cannot accept any more data at this time.
- T The other station has sent a TI. This condition does not cause control to be passed to the error routine. Control stays in the Write Routine, but the Write Routine does not attempt to write until the UAP has reset the PSB to N and the Buffer Status Byte to A. See TI in Appendix A.

THE SYSTEM MAILBOX

The System Mailbox is a 420 byte area from 0580 to 0999 in Common. The Mailbox is used by any System Ten partition when it desires to send a message to another partition. The SCA Module uses the Mailbox to get Initialization parameters from the UAP.

The first five positions of the System Mailbox are reserved for the status of the System Mailbox and the To and From Addresses of the communicating partitions:

<u>Field</u>	<u>Location</u>	<u>Use</u>
Status Byte	580	Status of the System Mailbox
		A Available for use by any partition.
		B Busy; not available until sending or receiving partition "frees" Mailbox by setting an A here.
To-Address	581 & 582	Two-character address of receiving partition.
		CO To SCA Module partition.
		XX To partition XX, where XX is any numeric or mnemonic value.

<u>Field</u>	<u>Location</u>	<u>Use</u>
From Address	583 & 584	Two character address of sending partition.
		CO From SCA Module partition.
		XX From partition XX where XX is any numeric or mnemonic value.

Initialization

If other applications use the System Mailbox, the UAP must check for an A in the Mailbox Status Byte before it can move an Initialization Record to the Mailbox. After moving the Initialization Record to the Mailbox, the UAP must post the Mailbox busy (B in the Status Byte) and load the To- and From-Address fields.

The UAP can detect when initialization has been successfully completed (when the SCA Module has processed the Initialization Record) by testing for an A in the Status Byte. If there is an error in the Initialization Record, the SCA Module posts an E in the Inhibit Switch. When this happens, the UAP must reset the Mailbox available; the SCA Module does not. Because until the UAP's Mailbox-handling logic has been thoroughly debugged, the UAP should not be tested while other applications that use the Mailbox are in memory.

INTERPARTITION INTERFACE

STANDARD MAILBOX FIELDS				INITIALIZATION RECORD			
STATUS BYTE	TO-ADDRESS		FROM-ADDRESS		TRANSMIT BUFFER DEFINITIONS		RECEIVE BUFFER DEFINITIONS
580	581	582	583	584	585	601	602
							622

CONFIGURATION-DEPENDENT FIELDS

MULTIPOINT-PRIMARY (Supplies remote station ID's to be transmitted by the SCA module)

MAILBOX			
IGNORED	REMOTE STATION ID LENGTH	REMOTE STATIONS ID's: First character is one of 32 used in SCA hardware-performed comparison. Second character is one of 64 used in software-performed comparison. ID list ends with a Z.	
623	643	644	645
			999

MULTIPOINT-TRIBUTARY (Assigns the SCA module an ID)

MAILBOX			
IGNORED	ID LENGTH	SYSTEM TEN ID CHARACTERS (left justified)	TO BE DEFINED BY USER
623	643	644	645 646 647
			999

DIAL (Calling or Answering)

MAILBOX					
UNUSED	LENGTH OF SCA's ID CODE (0-6)*	LENGTH OF ALL OTHER STATIONS' ID CODES (0-6)*	SCA's ID CODE (I AM)	ID CODES (WHO ARE YOU?) FOR OTHER STATIONS. ID LIST ENDS WITH Z	UNUSED
623	642	643	644	645 650 651	741 742
					999

* 0=Station(s) will not be sending ID code

FIGURE 4-1. FORMAT OF THE SYSTEM MAILBOX DURING SCA MODULE INITIALIZATION.

INITIALIZATION RECORD

<u>LOCATION</u>	<u>BYTES</u>	<u>VALUE</u>
580	1	B = Busy
581-582	2	C \emptyset = ID of SCA partition
583-584	2	n n = User partition number/mnemonic
585-601	17	Transmit buffer addresses. The interpretation of this field varies depending on the version of Get-Buffer selected.

Versions 1 (Sequential) and 2 (Priority) of Get-Buffer

585-601	17	Up to four 4-digit addresses, the last of which is immediately followed by the character Z. These are the addresses of the Transmit Buffers.
---------	----	--

Z in 585 signifies that no Transmit Buffers are being defined.

Version 3 of Get-Buffer (Priority/Sequential)

585-597	13	Up to three four-digit addresses of the sequential Transmit buffers, the last of which is immediately followed by the character Z.
---------	----	--

598-601	4	The address of the priority buffer.
---------	---	-------------------------------------

602-618	17	Up to four 4-digit addresses, the last of which is immediately followed by the character Z. These are the addresses of the Receive Buffers.
---------	----	---

Z in 602 signifies that no Receive Buffers are being defined.

619-622	4	The length (4 digits) of each Receive Buffer. Receive Buffers are all the same size. (See <u>Transmit and Receive Buffers: Receive Buffer Length</u> in this section.)
---------	---	--

623-999	119	The interpretation of this field varies from one type of SCA Module to another as shown below:
---------	-----	--

Point-To-Point Configuration

623-999	119	Ignored
---------	-----	---------

Centralized, Multipoint (Control Station) Configuration
 [Assigns the SCA module a station ID]

623-643	21	Ignored
644	1	Length of each poll list entry
645-999		Poll list

Centralized, Multipoint (Tributary Station) Configuration
 [Assigns the SCA Module a Tributary station ID]

623-643	2	Ignored
644	1	Length of ID (1 or 2)
645 or	1 or 2	The ID of this System Ten (1 or 2 digits).
645-646	2	The first character is one of 32 which the SCA uses in a hardware-performed comparison of the ID. The second character is one of 64 which the SCA uses in a software-performed comparison of the ID.

Dial Configuration

623-642	20	Unused.
643	1	Length of SCA Module's identification code (0-6). Zero specifies that the SCA Module is not to send an identification code.
644	1	Length of the other computers' identification codes (0-6). Zero specifies that the other computer(s) will not send identification codes. All remote computers must have the same length ID.
645-650	6	The SCA Module's identification code. This code is referred to as the I AM code.
651-741	90	The valid identification codes from the other computer(s) are stored here. This field may contain up to 15 6-digit codes, 18 5-digit codes, etc. When the field contains multiple codes, this System Ten may call or may be called by more than one other computer without reinitializing. These are referred to as WHO ARE YOU codes. They are ended by a Z.
742-999		Unused.

TRANSMIT AND RECEIVE BUFFERS

Transmitting or receiving data via the Singer SCA Module requires a set of one to eight buffers. This set, which consists of zero to four Transmit Buffers and zero to four Receive Buffers, resides in Common and is defined in the Initialization Record. At any time the SCA Module uses only the set of buffers defined in the last Initialization Record processed by the Initialization Routine.

Transmit and Receive Buffers may have the same addresses. A Transmit Buffer or a Receive Buffer or a combined Transmit/Receive Buffer may be defined at 0580C, thus overlapping the System Mailbox. However, overlapping buffers with the Mailbox may lead to serious timing problems if many partitions are contending for use of the Mailbox.

The design of the UAP is simplified if only one Transmit Buffer and one Receive Buffer are defined. However, transmission will be faster if more buffers are defined. Normally, the use of two Transmit and two Receive Buffers will provide optimum transmission speed.

Buffer Access

If Version 1 (Sequential) Get-Buffer is selected, the SCA Module uses the buffers one at a time, in sequence, beginning with the first. The UAP must use the buffers in the same sequence. The SCA Module uses index registers to point to the "current" Transmit and Receive Buffers. The following rules, showing the conditions under which the sequential Get-Buffer Routine updates its Transmit Buffer pointer, should help the user to use the buffers correctly.

1. If the current Transmit Buffer is not ready (status byte not = D), the SCA Module waits for that buffer to be posted ready.

In a Multipoint environment, each time the SCA module finds the buffer not ready, it sends an ENQ. If the buffer is still not ready after five ENQs have been sent, the SCA Module sends an EOT, posts an R in the Inhibit Switch, and waits for the UAP to clear the R. The next time the Write Routine is entered it waits for the same buffer to be posted ready.

In Dial and Point-to-Point environments, the SCA module sends the IDLE buffer after each five ENQ's. Ten IDLE buffer transmissions are sent before the EOT is sent. See IDLE message, page 4-29.

2. If the buffer is ready (status byte = D) but the UAP has requested in the Write Flag of the buffer that an EOT or DLE EOT be sent instead of data, it sends the EOT or DLE EOT (and it resets the Write Flag to V and the buffer status byte to A), but it does not update the Transmit Buffer pointer. The next time the Write Routine is entered, it waits for the same Transmit Buffer to be posted ready.
3. If the buffer is ready and contains data to be transmitted (no EOT or DLE EOT request), the SCA Module tries to transmit the data block. If there is an error in transmission, it posts an E (error) in the buffer status byte and an error code in the Inhibit Switch or the PSB or both, but it does not update its Transmit Buffer pointer. The next time the Write Routine is entered, it waits for the same buffer to be posted ready.
4. When the Write Routine has successfully transmitted data from a given Transmit Buffer, it resets the status byte to A (available) and the Write Flag to V and then it continues to the next Transmit Buffer.
5. If the SCA module detects a T (Transmission Interrupt) in the Partition Status Byte, then the buffer has already been transmitted; the buffer pointer is then updated to point to the next buffer in sequence.

Sequential Receive Buffers are handled in approximately the same way.

1. If the current Receive Buffer is not available (status byte not = A), the SCA Module waits for the buffer to be posted available. (The Read Routine sends WACKs while it waits.)

Note: This applies only after a station has been polled. The SCA Module will not poll a station if the current Receive Buffer is not available.

2. If the current Receive Buffer is available, the SCA Module tries to receive data from the other station. If it receives an EOT or DLE EOT instead of data, or if an error occurs, it posts the appropriate code in the PSB or the Inhibit Switch or both, but it does not update its Receive Buffer pointer. The next time the Read Routine is entered it will try to fill the same Receive Buffer.
3. When the Read Routine has successfully received data into a given Receive Buffer, it posts an E (filled) in the buffer status byte and then it tries to fill the next buffer.

Version 3 of the Get-Buffer Routine (Priority-Sequential) treats all Receive Buffers as sequential buffers (as described above). It treats the first Transmit Buffer as a priority buffer. It always checks the priority Transmit Buffer first. If that buffer is not ready, it checks one of the remaining (sequential) buffers. (See Get-Buffer Routine: Version 3 on page 3-7 in Section 3.)

The use of priority buffers is described under Get-Buffer Routine: Version 2 on page 3-6.

Transmit Buffer Layout

BUFFER STATUS BYTE	TO-ADDRESS		FROM-ADDRESS		DATA LENGTH				WRITE FLAG	UNUSED (Multipoint control sec. ID)	
	1	2	3	4	5	6	7	8		10	11
0									9		

SYNCHRONIZATION CHARACTERS (SYN's)			STX OR SOH	SEQUENCE NUMBER		LENGTH CONTROL FIELD		DATA			
12	13	14	15	16	17	18	19	20		

Transmit Buffer Length

The first 20 positions of the Transmit Buffer (relative locations 0-19) are called the Buffer Prefix. The remainder of the buffer is filled by the UAP with data to be transmitted. When defining Transmit Buffers in the Initialization Record (core locations 585-601 in the System Mailbox), the user must allow enough space for the largest number of data characters to be sent plus the 20-character Buffer Prefix.

Data Block

The following information is included in the data block sent to the other computer.

CHARACTERS SENT	RELATIVE LOCATION IN TRANSMIT BUFFER	COMMENTS
SYNs	12-14	These characters are not received into core by the other computer.
STX or SOH	15	See "Transmit Buffer Fields" below
Sequence Number	16-17	
Length Control Field	18-19	
Data	20 through X+20	X = the Data Length specified by the UAP in the Data Length Field of the Transmit Buffer.
ETB or ETX	SCA Module work area	See "Transmit Buffer Fields" - Write Flag below. Note that this control character does not require a position in the transmit buffer.

Transmit Buffer Fields

Following is a description of the values of the fields in a Transmit Buffer. The fields shaded in the Transmit Buffer layout on page 4-23 are used by the SCA Module and must not be modified by the UAP.

<u>Relative Location</u>	<u>Definition</u>
0	<u>Buffer Status Flag</u>
	A - Available. The Write Routine posts an A in this field to indicate that the buffer was successfully transmitted and is now available for further use by the UAP.
	D - Ready. The buffer is ready to be transmitted. The user sets this flag when a buffer is full and ready to be transmitted. When the buffer has been transmitted, status will be set to A or E.
	E - Error. The buffer could <u>not</u> be transmitted or something unusual has happened. The Partition Status Byte will always be set to some code indicating what has occurred. If an EOT has been received, the Inhibit Switch will also be set (to Q) and control will pass to the ESR.
	V - The buffer is currently being transmitted. This buffer is being used by the SCA partition and <u>must not</u> be modified by any other partition.

The SCA changes the D to V before transmitting the buffer and then from V to A or E after the I/O operation is complete.

1-2 To Address

The number or mnemonic (CO) of the SCA partition.

3-4 From Address

The number or mnemonic of the partition from which the buffer is being passed.

NOTE: The To and From Addresses are included in the buffer prefix primarily because buffers may overlap the Mailbox. The SCA Module does not use these values; it merely exchanges them when an I/O operation is complete. If a user program wants these values to be meaningful, it is the user program's responsibility to control them.

5-8 Data Length

The UAP uses this field to tell the SCA Module how many data characters to transmit from the Data field of the transmit buffer.

9 Write Flag

This field is set by the UAP in order to request that the Write Routine send a special control character to the other computer. If this field is set to other than D,H,Q, or X, all data blocks sent will begin with an STX and end with an ETB. The field should be initialized to a value other than D,H,Q or X (V is recommended). (See Assembling the SCA Module in Section 5.) When the Write Routine detects a D,H,Q, or X in this field it sends the appropriate control character and then resets the field to V.

V - Normal. The Write Routine will send STX, data and ETB.

D - The Write Routine should send a DLE EOT instead of a data block. Control will then pass to the ESR.

H - The Write Routine should send an SOH instead of an STX at the beginning of this data block.

(Note: All blocks that start with an SOH will be terminated with an ETB.)

Q - The Write Routine should send an EOT instead of a data block. Control will then pass to the ESR.

X - The Write Routine should send an ETX instead of an ETB at the end of this data block.

10-11 Unused, except for Multipoint Control Selection I. D.

12-14 Synchronization Characters

SYN characters are used to get the hardware at the two locations into synchronization.

15 STX or SOH

STX will normally precede all data blocks sent. An SOH will precede the data if the UAP has posted an H in the Write Flag.

16-17 Sequence Number

This field is created and moved into the Transmit buffer by the Write Routine. The first data block sent has sequence number 00. The sequence number is incremented by one before being placed in each subsequent data block. After 99 it goes to 00. The program in the other computer can use the sequence number to make sure that no data blocks have been lost. If the program in the other computer is an SCA Module, the Sequence Number will automatically be checked. (See the definition of this field under Receive Buffer Fields.) It is not used by the 4310 Write routine.

18-19 Length Control Field

This field is created and moved into the transmit buffer by the Write Routine. The value of the Length Control Field = the low-order two digits of: the UAP-supplied Data Length (relative locations 5-8 of the transmit buffer) plus 6 (1 for the STX or SOH, 2 for the Sequence Number, 2 for the Length Control Field, and 1 for the ETB or ETX.) The program in the other computer can use the Length Control Field to make sure that no characters have

been picked up or lost during transmission of the data block. If the program in the other computer is an SCA Module, the Length Control Field will automatically be checked. (See the definition of this field under "Receive Buffer Fields". It is not used by the 4310 Write routine.

20 through Data
 19 + X (X = the number of characters received)

The data length specified in the Data Length Field of the transmit buffer. This field contains the data to be transmitted.

Receive Buffer Layout

Buffer Status Byte	To-Address		From-Address		Data Length				Unused		
	0	1	2	3	4	5	6	7	8	9	10

Unused			STX or SOH	Sequence Number		Length Control Field		Data		ETB or ETX
12	13	14	15	16	17	18	19	20		

Receive Buffer Length

When defining receive buffers in the Initialization Record (Core locations 602-618 for addresses and core locations 619-622 for length; see Initialization Record in this section.) The user must allow space for the following:

- The 20-character buffer prefix,
- the largest number of data characters to be received,
- the ETB or ETX (one position),
- and one additional position, which may be required by the hardware if a data block received is too large.

Receipt of Data

After data has been successfully read by the SCA Module, the data block received will be found in the Receive Buffer beginning in relative location 15. The SCA Module will have updated all of the other fields (relative locations 0-8) in the Receive Buffer.

Receive Buffer Fields

The following is a description of the fields in a Receive Buffer. The Buffer Status Byte is the only field that the UAP is required to update. (The UAP uses the Buffer Status Byte to indicate to the SCA Module that the buffer is available for the receipt of data.) The other fields in the Receive Buffer may be modified by the UAP after the SCA Module has indicated in the Buffer Status Byte that the buffer has been filled with error-free data.

<u>Relative Location</u>	<u>Definition</u>
0	<p><u>Buffer Status Byte</u></p> <p>A - Ready. The UAP sets the Buffer Status Byte to A to indicate that the buffer is ready to receive a block of data. If the Read Routine is unable to read a block of data, the Buffer Status Byte will still be set to A. The UAP must test the Partition Status Byte to see if an error has occurred and the Inhibit Switch to see if an EOT or DLE EOT has been received.</p> <p>E - Full. The SCA Module posts an E in the Buffer Status Byte to indicate to the UAP the buffer has been filled with error free data.</p> <p>V - The SCA Module is filling the buffer.</p>
1-2	<p><u>To Address</u></p> <p>The number or mnemonic (CO) of the SCA Partition.</p>
3-4	<p><u>From Address</u></p> <p>The number or mnemonic of the partition from which the buffer is being passed.</p>

Note: The To and From Addresses are included in the buffer prefix primarily because buffers may overlap the Mailbox. The SCA Module does not use these values; it merely exchanges them when an I/O operation is complete. If a user program wants these values to be meaningful, it is the user program's responsibility to control them.

5-8 Data Length

The SCA Module stores in this field the number of data characters read into the buffer.

9-14 Unused.

Note: When data is being transmitted, The Write Flag in location 9 of the Transmit Buffer is used to request that the SCA Module send a special control character. When receiving data, the Read Flag (\uparrow ECB+1) is used for this function. (See Communications Control Block in this section).

15 STX or SOH

STX is normally the first character in a data block. See SOH in Appendix A for information about the recognition and use of an SOH.

16-17 Sequence Number

This field is checked by the SCA Module. It need not be referenced by the UAP. If there is a sequence error, the SCA Module will post an F in the Partition Status Byte.

18-19 Length Control Field

This field is checked by the SCA Module. It need not be referenced by the UAP. If there is a length error, the SCA Module will post an E in the Partition Status Byte.

20 through Data

19 + X (X = the number of characters received, which can be found in the Data Length Field of the buffer.)

The data received.

20 + X ETB or ETX (X is defined under Data above.)

The receipt of an ETX is not recognized as a special condition by the SCA Module. (It is not posted to the UAP.) If the application requires that some special function be performed upon the receipt of an ETX, then the UAP should test this field.

IDLE MESSAGE BUFFER

The IDLE Message Buffer is used by the SCA Module to format IDLE Messages.

Idle Messages

When the Write routine is sending data to the other station there can be temporary delays in which the user program is not able to fill buffers. When this occurs the SCA Module formats and sends up to ten dummy IDLE messages to the other station in order to prevent it from timing out. It may take up to 3 minutes to transmit these IDLE messages. Non-serious problems should be recovered within that time. A UAP that is transmitting data will not be aware that IDLE messages are being sent unless all ten are sent before the UAP has been able to post a transmit buffer available. After sending ten IDLE messages the SCA Module sends an EOT, posts an R in the Inhibit Switch, and passes control to the ESR.

A UAP that is receiving data must be able to recognize IDLE messages if the data blocks being received are being transmitted by an SCA Module in the other computer. A UAP can recognize an IDLE message by testing for the characters "IDLE" in relative locations 20-23 of the Receive Buffer. It may ignore IDLE messages unless the application indicates that the 20 characters in relative locations 24-43 of the buffer are meaningful and should be processed by the receiving UAP.

The SCA Module formats the IDLE message in a 44-byte buffer at the label ↑IDLEB. The IDLE message buffer has the following format and values:

Note that the IDLE Message Buffer has the same format as the Transmit Buffers used by the UAP.

Field	Buffer Status Byte	Unused				Data Length				Write Flag	Unused	
Value	D					0	0	2	4	V		
Location	↑IDLEB	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11

Field	Synchronization Characters			STX	Sequence Number		Length Control Field		Data					
Value	SYN	SYN	SYN	STX			3	0	I	D	L	E	20 Data Characters	
Location	+12	+13	+14	+15	+16	+17	+18	+19	+20	+21	+22	+23	+24	+43

The values shown are moved by the SCA Module into the IDLE Message Buffer just before the IDLE message is transmitted. The Sequence Number is incremented normally, i.e., just as if the IDLE messages were regular data blocks. The 20 data characters are moved from the location ↑IMSG in Common to relative locations 24-43 of the IDLE message buffer. ↑IMSG begins at location 560C (the location of the Partition Status Bytes); the Partition Status Bytes will be moved into the IDLE Message Buffer and sent to the other station.

The contents of relative locations 12-43 of the IDLE Message Buffer are transmitted to the other station, followed by an ETB.

Location of ↑IMSG

↑IMSG begins at the location of the Partition Status Bytes because the UAP that is receiving data may want to determine the status of the partitions in the transmitting computer. If all active partitions post their status in their Partition Status Byte, the other computer may be able to determine, for example, that the disk is down in partition four. Using this tool it is theoretically possible to operate remote System Tens as line concentrators without operators and hence to monitor the status of the remote systems from a central computer.

If the user wishes that other meaningful data be sent in the IDLE message, he can change the address in the ORG card preceding the ↑IMSG card. The meaningful data can be moved into the field at execution time by the UAP or it can be placed in the field at assembly time by coding a constant in the DM statement of ↑IMSG.

Location of ↑IDLEB

↑IDLEB is at location 1000C. The user may move it to any other location in Common. If it is put at 0580C, the contents of the System Mailbox will be destroyed every time an IDLE Message is sent. The SCA Module will not test the status of the Mailbox before moving IDLE message fields into it. Therefore it is recommended that IDLEB be moved to another location in Common unless the SCA Module and its associated UAP are the only programs using the System Mailbox while data is being transmitted.

↑CWA and ↑PSWA

The Call Work Area (↑CWA) is used only in the Dial-Call environment. (See SCAM Dial User Application Guide for description of use.)

The Poll/Select Work Area (↑PSWA) is used only in the Multipoint-Tributary environment. (See the SCAM Multipoint User Application Programming Guide for a description of its use.)

Both these areas can be defined separately from the buffer, or they may overlap the IDLE Message Buffer.

Section 5

ASSEMBLING THE SCA PROGRAM

INTRODUCTION

CORE REQUIREMENTS

INITIAL VALUES OF COMMON

CHOOSING THE HOUSEKEEPING ROUTINE

ASSEMBLY CONSIDERATIONS

ASSEMBLING THE SCA PROGRAM

INTRODUCTION

The user creates an SCA module by selecting appropriate routines from the 20 released in the SCAM package, making modifications of Common constants as necessary to fit with his own UAP, and assembling the resulting SCA module. The module should be assembled using Singer Business Machines' Assembler I or Assembler II programs. (See manuals 40-029 or 40-279.)

CORE REQUIREMENTS

The table on the next page lists the 20 SCAM routines and core requirements for each:

TABLE 5-1: SCAM ROUTINE CORE REQUIREMENTS

Housekeeping and Constants

<u>Routine and version</u>	<u>Locations required</u>	<u>Description</u>
Housekeeping - Primary	80	ORG at 0000.
Housekeeping - Variation 1	60	ORG at 0300.
Housekeeping - Variation 2	80	ORG at 000; used with TATTLE.
SCAM Constants	190	Mandatory.
Common	Variable	Includes ↑BASE through Mailbox. Partition Status Byte: 20 locat- ions. Mailbox: 480 locations. ↑ECB: 18 locations. Add 44 locations for each of the following used: ↑IDLEB, ↑CWA, ↑PSWA.
TATTLE	111	Part in Common, part in a Monitor partition.

Communications Control System

Initialization - 1	390	One-time initialization.
Initialization - 2	390	Optional reinitialization.
Initialization - 3	380	Forced reinitialization.
Get Buffer - 1	260	Sequential
Get Buffer - 2	270	Priority
Get Buffer - 3	410	Priority/Sequential
Error - 1	280	Small error table.
Error - 2	580	Large error table.
Read - 1	860	Read data from a System Ten computer.

Read - 2	950	Read data from a 4310 MDRS.
Write - 1	1040	Write data to a System Ten computer.
Write - 2	1080	Write data to a 4310 MDRS.

TABLE 5-1: SCAM ROUTINE CORE REQUIREMENTS (CONTINUED)

Environment-Specific Routines

<u>Routine and version</u>	<u>Locations required</u>	<u>Description</u>
ESR - 1	390	Point to point; primary.
ESR - 2	400	Point to point; secondary.
ESR - 3	1120	Multipoint; control.
ESR - 4	890	Multipoint; tributary.
ESR - 5	1360	Dial; answering.
ESR - 6	1360	Dial; calling.

INITIAL VALUES OF COMMON

The following areas are defined in the distributed Common deck and must be located in Common. The first six (\uparrow BASE through \uparrow K101) are referenced only by the SCA module; the rest are referenced by both the UAP and the SCA module.

The Common deck must be included when assembling the SCA module; if the UAP and SCA module are assembled separately, the Common deck must be included with each. (To avoid doubly loading the Common areas in the case of separate assembles, set COMTXT=YES when assembling the SCA module and COMTXT=NO when assembling the UAP.)

The user can alter the location of certain of these areas and set their initial values by making changes to the Common deck before assembling.

- Buffer Prefix Displacements: includes fields defined by labels \uparrow BASE, \uparrow B1, \uparrow B5, \uparrow B9, \uparrow B10 through \uparrow K101. These are used by the SCA module and must not be changed. They require no Common core space allocation.
- Communications Control Block: 18 characters starting at \uparrow ECB. Relocatable by changing the ORG card, but must be in Common.

- Partition Status Bytes: 20 characters (one for each partition), starting at ↑STAT. Must start at 0560C, according to Common Core conventions.
- IDLE Message Contents: 20 characters, starting at ↑IMSG. May overlay ↑STAT (i.e., may be ORGed at 0560C).
- System Mailbox: 420 characters starting at ↑MAIL. Must start at 0580C by Common Core standards.
- IDLE Message Buffer: 44 characters, starting at ↑IDLEB. Released to be at 1000C, but may be relocated in Common as desired by changing the ORG card. Used only in Point-to-Point and Dial.
- Call Work Area: 44 characters, starting at ↑CWA. Used only by Dial Call ESR, may be omitted otherwise. Relocatable by changing the ORG card. (May overlap ↑IDLEB.)
- Poll Select Work Area: 44 characters, starting at ↑PSWA. Used only by Multipoint Tributary Handshaking, may be omitted otherwise. Relocatable by changing the ORG card.
- Transmit Buffer(s): Released as 100 characters each, starting at address specified by ↑SEND (in Mailbox). Size may be specified by UAP. Relocatable by changing the ORG card.
- Receive Buffer(s). Released as 102 characters each, starting at address specified in ↑SINK (in Mailbox). Size may be specified by UAP. Relocatable by changing the ORG card.

TABLE 5-2: COMMON FIELDS

<u>Label</u>	<u>Core Required</u>	<u>Location in Common</u>	<u>Use</u>	<u>Initial Value (at Load Time)</u>
↑BASE		0000C	Displacement	Not applicable
↑B1		0001C	Displacement	Not applicable
↑B5		0005C	Displacement	Not applicable
↑B9		0009C	Displacement	Not applicable
↑B10		0010C	Displacement	Not applicable
↑K101		0101C	Displacement	Not applicable
↑ECB	18	Relocatable (0400C)	CCB	zeroes
↑STAT	20	0560C	PSBs	all blanks or zeroes
↑MAIL	420	0580C	System Mailbox	first character = A if no other parti- tions are using the mailbox; do not initialize if other partitions are us- ing the mailbox.
↑IDLEB	44	Relocatable (1000C)	Idle Buffer	any value
↑IMSG	20	Relocatable (0560C)	Idle Message	any value
↑CWA	44	Relocatable (1000C)	Call work area	first character = 0 or blank
↑PSWA	44	Relocatable (1000C)	Poll/select work area	first character = 0 or blank

CHOOSING THE HOUSEKEEPING ROUTINE

Table 5-3 shows three versions of the Housekeeping routine: Primary, Variation 1, and Variation 2. The user should choose one routine, depending on whether the SCA module is to be loaded at location 0000 in the SCA partition, and whether TATTLE is being used in Common, or whether the SCA module is to be loaded at location 0300 of the SCA partition.

All three versions are distributed in the SCA package, with asterisks in column 1 for versions 2 and 3. If either version 2 or version 3 is to be used, the user should repunch the deck, removing the first column asterisks in the desired version, and removing the other versions from his deck.

Version 1: this version of the Housekeeping routine should be used if the loader used gives control to the loaded SCA partition at 0000p.

Version 2: this version of the Housekeeping routine should be used if the first 300 or more locations of the SCA partition are reserved for indirect use by the Conversational Loader. The user should repunch the ORG 300 card if the loader to be used needs more than 300 locations.

Version 3: this version of the Housekeeping routine should be used if the loader used gives control to the loaded SCA partition at 0000p and the routine TATTLE has been loaded into Common at the time the SCA partition becomes active.

TABLE 5-3: HOUSEKEEPING ROUTINES

Version 1

	ORG	0000	
↑ORG	M	47P92),↑KU(1)	FORM 5 TIMES PARTITION NUMBER
	A	↑KU(3),↑RD8+1(4)	ADD TO 101C FOR B-REGISTER
	BC	50(5)	
	BC	0(0),30(8)	MACHINE CHECK - LOOP
	ORG	50	
	MC	30(10),0	MOVE LOOP INSTRUCTION TO 0
	MC	47P(2),↑ECB+6	SET PARTITION EXECUTING
	S	47P(2),↑STATW	FORM PARTITION STATUS ADDRESS

Version 2

*	ORG	300	
*	BC	0(8)	MACHINE CHECK - LOOP
*↑ORG	M	47P(2),↑KU(1)	FORM 5 TIMES PARTITION NUMBER
*	A	↑KU(3),↑RD8+1(4)	ADD TO 101C FOR B-REGISTER
*	MC	↑ORG-10(10),)	MOVE LOOP INSTRUCTION TO 0
*	S	47P(2),↑STATW	FORM PARTITION STATUS ADDRESS
*	MC	47P(2),↑ECB+6	SET PARTITION EXECUTING

Version 3

*	ORG	0	
↑ORG	BC	TATTLE(5),+10(0)	
*	M	47P(2),↑KU(1)	FORM 5 TIMES PARTITION NUMBER
*	A	↑KU(3),↑RD8+1(4)	ADD TO 101C FOR B-REGISTER
*	BC	60(5)	
*	ORG	50	
*	BC	0(0),50(8)	MACHINE CHECK - LOOP
*	MC	56(4),06	MOVE RETURN FROM TATTLE
*	S	47P(2),↑STATW	FORM PARTITION STATUS ADDRESS
*	MC	47P(2),↑ECB+6	SET PARTITION EXECUTING
*			
*			*END OF SELF INITIALIZATION
*			*NEXT ROUTINE MUST BE AN ESR
*			
*			*THE NEXT INSTRUCTIONS MUST BE IN A MONITOR PARTITION
*			
*	C	FREE,MSGQ	IS MSGQ EMPTY?
*	BC	*+40(2)	IF YES, GO TO NEXT ROUTINE
*	W	MSGQ(0),10(1)	IF NO, WRITE MESSAGE
*	W	CARRET(0),1(3)	CARRIAGE RETURN
*	MC	FREE(10),MSGQ	CLEAR MSGQ
*			
*			*THE NEXT INSTRUCTIONS MUST BE IN COMMON
*			

*TATTLE	C	FREE,MSGQ	IS MESSAGE QUEUE EMPTY?
*	BC	*+10(2),*-10(8)	IF NO, BRANCH AND SWITCH
*	MC	40P(10),MSGQ	IF YES, PLACE ERROR BUFFER IN MSGQ
*	MC	6P(4),GOOUT+6	ADJUST NEXT BRANCH TO PROPER ADDRESS
*GOOUT	BC	GOOUT(0),GOOUT(8)	BRANCH TO PARTITION BEGINNING
*FREE	DM	C' '	
*MSGQ	DM	C10' '	BLOW-UP MESSAGE QUEUE

ASSEMBLY CONSIDERATIONS

A complete SCA program consists of the UAP and the SCA module.
A proper SCA module consists of the following segments:

1. Common deck
2. SCA partition program, containing:
 - a. one Housekeeping routine
 - b. one ESR
 - c. one Initialization routine
 - d. one Get-Buffer routine
 - e. one Error routine
 - f. Read routine (unless configuration allows it to be omitted -- see p. 3-10. If not being used, include ↑READ, ↑RDEX, ↑RD8 for proper assembly.)
 - g. Write routine (unless configuration allows it to be omitted -- see p. 3-11. If not being used, include ↑WRITE and ↑WRTEX for proper assembly.)
 - h. SCA partition constants

As was stated in section 3, the user selects the appropriate routines from the 20 provided, and normally groups his selection together with the Common deck and the UAP to assemble an SCA program. If the UAP and SCA module are assembled together, and the SCA partition program contains only one version of each of the above routines, the user need not insert any additional coding to prevent assembly errors.

If, however, the UAP and the SCA module are assembled separately, care must be used in handling the Common deck to prevent assembly errors. The Common deck may be used in both assemblies.

This will produce object decks with the Common deck in each (or object text files, if disc is specified as the object medium during assembly), unless the COMTXT option (described earlier in this section) is used. If the COMTXT option has not been used, it is important to remove the duplicate Common deck from the UAP prior to loading, leaving only one set of Common definitions in the load deck. Alternately, if assembled separately, the UAP may use a different set of labels at the same locations as the labels in the Common deck provided the COMTXT=NO option is used during assembly.

If there are to be two or more SCA partitions, SCA program object code must be produced for each partition; it may be the same program for each partition, or it may be different. However, the user must make modifications to the Common deck, as indicated below:

- The following labels may be used by the UAP and are used by all SCA partition programs, and may remain unchanged in their distributed form:

↑BASE ↑STAT

↑B1 ↑MAIL

↑B5 ↑SEND

↑B9 ↑SINK

↑B10 ↑CALL

↑K101 ↑IDEN

CARRET (if TATTLE is being used)

- The following labels must be redefined in Common (using the same labels, but at different locations) for each additional SCA partition program:

↑ECB

↑IDLEB (only if using Point-to-Point or Dial)

↑CWA (only if using Dial-call)

↑PSWA (only if using Multipoint tributary)

Transmit buffer(s)

- Because there will thus be more than one ↑ECB, etc. each SCA partition program will have to be assembled separately, with the appropriate Common deck consisting of the labels specific to the partition program, plus the shared Common labels (↑BASE through ↑IDEN and CARRET, if used).
- The following labels should be redefined in Common for each additional SCA partition program, as above; they may, however, be shared by all partition programs:

↑IMSG

up to 4 receive buffers

Section 6

LOADING THE SCA PROGRAM

USING THE DMF MULTIPARTITION LOADER

USING CARD LOADERS



LOADING THE SCA PROGRAM

A standard Singer multipartition loader should be used to load the SCA program and UAP. This section will discuss using the DMF Multipartition Loader (see manual 40-308), the Multipartition Loader -- Card Version (see manual 40-309).

The user should keep in mind two major considerations when deciding how and where to load the SCA program:

1. The SCA partition program and Common deck must be loaded and operating before the UAP is loaded. This is because the first thing the SCA partition program does is post its partition number in \uparrow ECB. If recommended procedures were followed in the design of the UAP, \uparrow ECB will be checked as the first action of the UAP, and if no partition number is found there, an error condition will result.
2. The SCA partition is necessarily "blind." The SCA uses the IOC, thus preventing any slow-speed IOC devices from being attached to that partition. Errors and other information to be reported from that partition must thus pass through the mailbox or other buffers in Common on their way to being reported via devices attached to other partitions.

USING THE DMF MULTIPARTITION LOADER

Use of the DMFMPL requires certain operational considerations:

1. Version 2 of the Housekeeping routine must be used because the DMF Conversational Loader uses the lower 300 locations of a partition; version 2 is the only version that does not Origin at location 0 in a partition. A minus sign (-) in column 20 of the MPL card will allow loading below 300.
2. The mailbox must be available to DMF, and must not have initialization information stored there at load time; if DMF finds information stored there, it will report an error message.
3. Neither the UAP nor the SCA module may be loaded into Partition 0, since the DMF low core module and the DMFMPL normally reside there, and the MPL will not load partition 0. The conversational loader may be used to load P0.

The user is referred to the DMF Multipartition Loader Reference Manual (40-308) for specific instructions in the use of this loading program. The user will need to prepare control statements for the SCA module and for the UAP.

- Since the SCA module must be loaded first, the MPL control card must have an L in the positions (1-19) relative to the partition(s) where it is to reside.
- The UAP is loaded after the SCA module, using an MPL control card with an L in the position relative to the partition where it is to reside or with conversational loader if PO.

NOTE: Remember to insert Start cards at the end of the object decks for both SCA and UAP or if using Assembler II use

END LABEL

feature where LABEL is at the starting address of that partition.

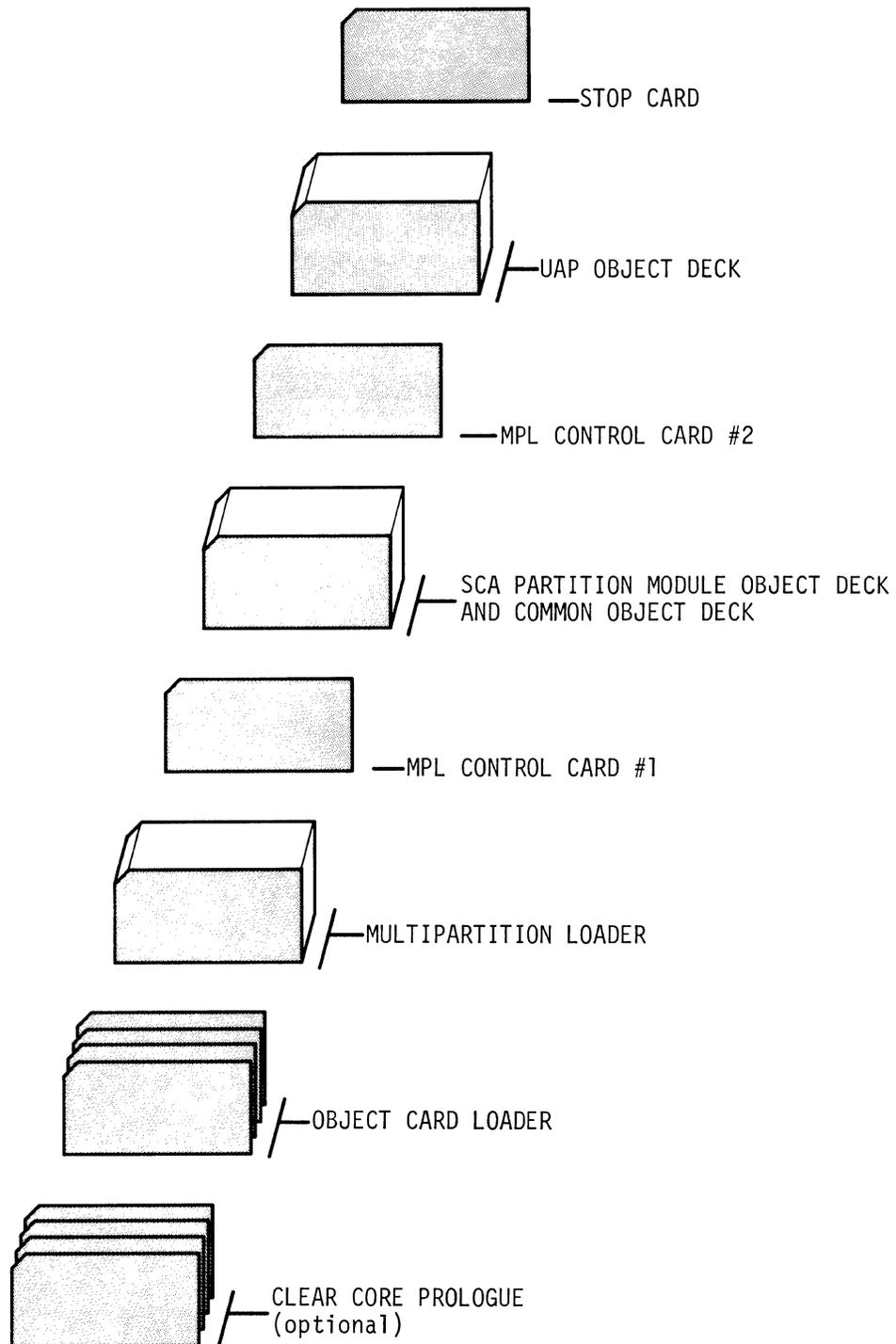


FIGURE 6-1: DECK SETUP FOR MPL RUN WHERE UAP TO RESIDE IN PARTITION OTHER THAN 0

USING CARD LOADERS

If using the card version of the Multipartition Loader, the user should see the Multipartition Loader (Cards) Reference Manual (40-309) for specific instructions in the use of this program.

To use the card MPL, the loading program itself will have to be loaded, using the Clear Core Prologue and the Object Card Loader (see the IOC Loading Package manual, 524-701015-M47). The Prologue and the Object Card Loader are bootstrapped in from the card reader, using the instruction I001010290 entered from the workstation.

One limitation to the use of the card MPL is the fact that it will not load a program into Partition 0. This should be no problem for loading the SCA module, since the latter's channel requirements mean that the partition in which it resides is "blind" (no slow-speed devices), an impossible condition for Partition 0 if DMF is being used at all. If, however, the UAP is to be loaded into Partition 0, the Clear Core Prologue and Object Card Loader should be used for the UAP's loading.

As with using the DMFMPL, the SCA module and Common deck should be loaded first (see deck order in Figure 6-1), with the MPL control card having an L in the position(s) relative to the Partition(s) where the SCA program is to reside.

If the UAP is to reside in a partition other than 0, it may next be loaded by the MPL, with the MPL control card having an L in the position(s) relative to the partition(s) where the UAP is to reside.

If the UAP is to be loaded into Partition 0, the MPL run should be terminated after loading the SCA module, and the user should enter the Clear Core Prologue and Object Card Loader, followed by the object cards for the UAP, through the card reader, after entering the bootstrap instruction on the workstation.

Appendix A

LINE CONTROL CHARACTERS



LINE CONTROL CHARACTERS

INTRODUCTION

Line control characters are used to establish and maintain the data link for message transmission. When control characters are received, they are converted to internal characters of the System Ten computer as shown in Table A-1. When in memory, control characters appear as System Ten data characters. They are distinguishable from data characters because of their position in a data message, or because of the point of time within a transmission that they are sent or received.

The SCA Module checks all control characters received, and, except for message length and sequence number, makes no check of the data between them. The user has no direct control over when the control characters ACK0, ACK1, ENQ, NAK, SYN or WACK are sent. However, through the use of the Read and Write flags (see section 4), the user can directly control when SOH, STX, ETB, ETX, TI, EOT, or DLE EOT are sent.

LINE CONTROL CHARACTERS

TABLE A-1 SCA LINE CONTROL CHARACTERS

<u>CONTROL CHARACTERS</u>	<u>SYSTEM TEN INTERNAL CHARACTERS(S)</u>	<u>ASCII BIT STRUCTURE</u>
		7 5 4 3 2 1
ACK0	P0	(1 1 0 0 0 0 (0 1 0 0 0 0
ACK1	P1	(1 1 0 0 0 0 (0 1 0 0 0 1
SOH	A	1 0 0 0 0 1
STX	B	1 0 0 0 1 0
ETX	C	1 0 0 0 1 1
EOT	D	1 0 0 1 0 0
ENQ	E	1 0 0 1 0 1
NAK	U	1 1 0 1 0 1
SYN	V	1 1 0 1 1 0
ETB	W	1 1 0 1 1 1
WACK	P;	(1 1 0 0 0 0 (0 1 1 0 1 1
TI	P<	(1 1 0 0 0 0 (0 1 1 1 0 0
DLE EOT	PD	(1 1 0 0 0 0 (1 0 0 1 0 0
DLE	P	1 1 0 0 0 0

CHARACTER DESCRIPTIONS

Following is a description of when and for what purpose control characters are sent or received by the SCA Module.

ENQ Enquiry

ENQ is used to request a response from another station.

1. It is used by both Point-to-Point Primary and Point-to-Point Secondary ESRs to bid for master status. If the station sending the ENQ gets an ACK0 in response, it can begin sending data. If it does not get an ACK0 in response, no data is sent (master status has not been achieved). The ENQ 'bid' is sent again.
2. It is used by the Multipoint-Control ESR as part of the polling or selection sequence. This routine sends ID characters, followed by an ENQ, to the tributary station as an addressing sequence.
3. It is used by the Dial-Call ESR as part of the initial information exchange. After sending its own ID characters (I AM) Dial-Call sends an ENQ (interpreted as "tell me who you are"). The normal response from the answering station is ID-ACK0. If no ID characters are exchanged only ENQ is sent as a readiness query. ("Are you ready?")
4. It is used by the SCA module to request a repeat of the last control character transmission from the other station, usually because the last control characters received were uninterpretable.
5. It is used to respond to a WACK sent by the SCA module. If the other station responds with an ENQ, the SCA is free to either repeat the WACK, or to indicate to the other station that the reason for waiting has passed (i.e., respond with the proper ACK to the 'last' data message received because a receive buffer is now available.) If the response is anything other than an ENQ, the receiving SCA module stops transmission and "hangs up."

ACKn Acknowledgement N

ACKn is used as an affirmative response to the other station. The SCA Module always sends ACK0 first (when the link is being established) and always alternates ACK0 and ACK1 in response to data. ACK0 and ACK1 are alternated in order to insure that no data messages are lost.

- ACK0 is used:
 1. as an affirmative response to data.
 2. by Point-to-Point Control and Tributary as acceptance of the other station's bid (ENQ) for master status.
 3. by Dial-Answer as acceptance of the other station's (ID) ENQ.
 4. by Multipoint Tributary as a response to a selection (ID ENQ) message from the control station.
- ACK1 is used only as an affirmative response to data.

The SCA Module never expects to receive a literal ACK0 or ACK1. Instead, it expects to receive DLE0 or DLE1, which are interpreted as ACK0 and ACK1. Note that whenever a DLE is received by the SCA hardware, CC=3 is set on, and the DLE character does not come into memory.

NAK Negative Acknowledgement

NAK is used as a negative response to the other station.

1. The Read routine sends NAK when it did not receive the last data block correctly.
2. When the Write routine receives a NAK, it automatically resends the last data block (up to 8 successive times). See Partition Status Byte code=J.

WACK Wait Acknowledgement

This is a communications control character used as a request for a short period of idling before accepting another logical record. WACK is treated as an abnormal response in that control passes to the Error routine.

It is used by the Read routine when the Get Buffer routine indicates no buffers are available to receive data; a WACK is then sent to indicate that no more data can be handled at the moment.

Whenever there is a need to wait, the SCA Module sends a WACK. It never sends and never tests for receipt of a WABT (Wait Before Transmission). WACK implies a positive acknowledgement to the last block received, while WABT does not imply acknowledgement. Hence, when the reason for waiting has passed (an ACKn is sent instead of another WACK), the ACKn is an acknowledgement for the 'last' block and transmission may continue with the 'next' block. However, if

WABT is used as the wait character, there is some ambiguity as to the status of the 'last' block (perhaps it should be retransmitted). In order to eliminate the ambiguity and not duplicate records received, the SCA Module uses WACK instead of WABT.

Caution: The Read routine does not count the number of WACKs it sends. If it is sending WACKs to an SCA Write routine, the Write routine counts the number of WACKs received, and after 8 posts an S in the Partition Status Byte. The UAP may allow the wait to continue, or may direct the Write routine to hang up (see Write flag). However, if the Read routine is communicating with a non-SCA Module, and that module does not limit the number of WACKs received before hanging up, it is possible to get into an infinite loop.

DLE EOT Mandatory Disconnect

This control character pair indicates that transmission must cease immediately. Whenever it is transmitted or received, the SCA Module disconnects the line. See Read and Write flags (page 4-25) for a discussion of how a UAP may send a DLE EOT.

TI Transmission Interrupt

This is a pair of communications control characters used to indicate that the user wishes to interrupt receipt of data in order to send one or more priority data blocks of its own.

The SCA module recognizes BTAM's RVI (Reverse Channel Interrupt) as a TI, and the SCA sends a TI to accomplish the same function as an RVI. A TI is sent by a UAP that is receiving data when it wishes to interrupt the receipt of data so that it can send one or more priority messages to the other station. An SCA Module that is receiving data will send a TI character when the UAP has posted a 3 in the Read flag in the Communications Control Block. After sending a TI, the SCA module will continue to attempt to read. An SCA module that is sending data will post the receipt of a TI to the UAP by moving a T to the Partition Status Byte and an E to the Buffer Status Byte of the transmit buffer. After posting a T in the Partition Status Byte, the SCA module will not continue writing until the PSB has been cleared. A transmitting UAP may receive priority messages by directing the Write routine (in the Write flag) to send an EOT and then going to its read logic.

Following is an example of the UAP logic that might be used when a receiving station wishes to interrupt the receipt of data so that it can send a single high-priority message and then resume receiving data.

UAP in Receiving Station

1. Post a 3 in the Read Flag of the CCB.
(The SCA Module will send a TI.)
7. Detect that EOT has been received (W in the Inhibit Switch).
8. Move the high-priority message to a Transmit Buffer.
9. Post the buffer ready to transmit.
10. Reinitialize if necessary.
11. Request that data be written (post a D in the Operation Code of the CCB).
12. Clear the Inhibit Switch (set it to zero).

UAP in Transmitting Station

2. Detect that a TI has been received (T in the PSB).
3. Post all Transmit Buffers not available.
4. Request that an EOT be sent (move a Q to the Write Flag and a D to the Buffer Status Byte of a Transmit Buffer).
5. Reinitialize if necessary.
6. Prepare to receive data; post all receive buffers available.

UAP in Receiving Station

UAP in Transmitting Station

13. Detect an A in the Buffer Status Byte of the Transmit Buffer. (The data has been transmitted.)

14. Receive and process the high-priority message.

15. Request that an EOT be sent (post a D in the Write Flag of the Transmit Buffer).

16. Detect that an EOT has been received (W in the Inhibit Switch).

17. Prepare to resume receiving data (reinitialize if necessary, post Receive Buffers ready).

18. Prepare to resume transmitting data (reinitialize if necessary, prepare Transmit Buffers and post them ready, move a D to the Operation Code of the CCB).

EOT End of Transmission

This is a communications control character used to indicate the end of a transmission. It may be sent as a result of the user correctly posting the Read or Write flag with a '1' or 'Q' respectively.

It is used by the Dial routines to turn around the direction of transmission on a line, without disconnecting the line. If a sending station sends an EOT, it means there is nothing more to transmit; the receiving station then has the option of sending an ENQ, if it has data to send, or an EOT, which means that the calling station may then hang up. (Neither station has any data to send.)

SYN Synchronous Idle

This is a communications control character used to achieve and maintain character synchronization. All SYNs are deleted by the hardware receiving them, and hence never appear in memory.

The following should be kept in mind about transmitting SYNs:

1. When the line is idle, the SCA hardware sends a continuous stream of SYNs.
2. When data is being transmitted, the SCA hardware sends two SYNs every second.

3. The SCA Module sends three SYN's before every data block.

SOH Start of Heading

This is a communications control character used to denote the beginning of a message heading. It is not normally used by the SCA module; it can be generated if the UAP posts an H in the Write flag. It or an STX are used in position 15 of the Transmit Buffer.

If an SOH is sent, an ETB will be sent at the end of the block.

The receipt of an SOH has no special significance to the SCA module. It is accepted as a valid first character of any data block. When the Read routine detects an SOH in the first position of a data block, it branches to ↑HDRA. If the user has not modified the distributed package, ↑HDRA will be a branch back to normal Read routine processing. The receipt of the SOH will not be posted to the UAP.

If an application requires that header messages be processed, the user has two options:

1. Coding can be added at ↑HDRA in the Read routine to process the header records. The user coding should begin at the label ↑HDRA and should end with a branch back to normal Read routine processing.
2. The user's header processing routine can be included in the UAP. (The distributed package need not be changed.) The UAP will have to test location 15 of the Receive Buffer for an SOH to determine if a header record has been received.

Caution: If an SOH is received and an STX follows the header in the same block, there is no way with the System Ten to distinguish the STX from a 'B' in the data, unless the header is of fixed length, i.e., the STX can be identified by its position in the data record. For this reason, the SCA Write routine never sends an STX in a block started with SOH.

STX Start of Transmission

This is a communications control character used to delimit the beginning of a transmission block. All data blocks sent by the SCA Module start with either STX or SOH.

ETB End of Text Block

This is a communications control character used to terminate a data block.

It will be generated, at the end of all data blocks unless the Write flag = X.

ETX End of Text

This is a communications control character used to terminate a logical group of data blocks, each of which are terminated by ETB's.

It will be generated when the Write flag is posted with an X.

Appendix B
4310 SCA MODULE

4310 SCA MODULE

INTRODUCTION

This appendix should be used as reference when creating an SCA Module or designing and writing a user application program (UAP) to communicate with a 4310 series Magnetic Data Recording System (MDRS) terminal.

Since 4310 MDRS SCA Module usage is identical to standard SCA Module usage in most respects, the MDRS SCA Module user should understand the material presented in the earlier portions of this manual and the SCA User Application Guide dealing with the configuration to be used. This appendix covers the differences between the MDRS SCA Module and the standard SCA Module. It covers the following topics:

- The differences between the standard SCA Module routines and their 4310 SCA Module counterparts.
- Modifications to the interpartition interface conventions.
- Line control conventions.
- Remote device selection.
- Remote device control.

The following 4300 series and 4310 series manuals may also be of interest to the 4310 MDRS SCA Module user:

PUBLICATION NO.	TITLE
15-146	4310 Series MDRS Communications Operator Instructions
15-147	Super Dup Reference Manual
15-045	Series 4310 Communications Reference Manual

Two device-specific routines are provided -- a 4310 Read Routine and a 4310 Write Routine. These are designed to be used in place of the standard SCA Read and Write Routines and are compatible with any versions of the standard SCA Handshaking, Initialization, Get-Buffer, and Error Routines. The 4310 Read and Write Routines allow the user to fully utilize the capabilities of a 4310 series MDRS terminal. In this document an SCA Module composed of 4310 Read and/or Write Routines and standard SCA Handshaking, Initialization, Get-Buffer, and Error Routines will be referred to as a "4310 SCA Module".

4310 Read Routine

The 4310 Read Routine receives data from the tape drive of the 4310 series MDRS terminal.

Because the 4310 terminal does not generate sequence and length fields in the data block it transmits, the 4310 Read Routine does not include the standard SCA Module Read Routine's sequence and length checks. This removal of sequence and length checking has no effect on UAP design. In all other respects the 4310 Read Routine is identical to the standard Read Routine.

4310 Write Routine

The 4310 Write Routine transmits data to the tape drive or the line printer of a 4310 series MDRS terminal. It differs from the standard SCA Module Write Routine in three respects:

1. Sequence and Length Fields

Because the 4310 terminal cannot perform sequence and length checking, the logic that generates sequence and length fields has been removed from the 4310 Write Routine. The removal of this logic has no effect on UAP design.

2. Temporary Text Delay

The 4310 Write Routine sends a Temporary Text Delay (TTD) line control sequence (STX ENQ) instead of an IDLE message when no Transmit Buffers are ready for transmission. The 4310 is expected to send a NAK in response to the TTD. When the 4310 Write Routine has sent a TTD (and received a NAK response) one hundred consecutive times without finding a Transmit Buffer ready to transmit, it "gives up": it sends an EOT, posts an R in the Inhibit Switch, and returns to the Handshaking Routine.

This difference does not affect the design of the UAP. The UAP need never be aware that TTDs are being sent until one hundred have been sent and an R has been posted in the Inhibit Switch. The R has the same meaning to a 4310 UAP as to a UAP operating with a standard SCA Module, even though the line control procedures preceding the posting of the R are not the same.

3. Remote Device Selection

The 4310 Write Routine allows the UAP to select the device to which it will transmit (4310 tape drive or line printer). See Remote Device Selection in this appendix for a description of the method by which the UAP requests device selection.

Core Requirements

ROUTINE	CORE REQUIRED
4310 Read Routine	950
4310 Write Routine	1080

Creating A 4310 MDRS SCA Module

To create a 4310 MDRS SCA Module, the user must assemble the 4310 Read Routine or the 4310 Write Routine or both, with the appropriate versions of the standard SCA Module Handshaking, Initialization, Get-Buffer, and Error Routines. The selection of versions of the standard routines is determined by the application as in standard SCA Module use. Once the appropriate routines have been selected, the assembly of the 4310 SCA Module is the same as the assembly of a standard SCA Module.

INTERPARTITION INTERFACE CONVENTIONS

The five areas of Common used by the SCA Module for communication with the UAP remain unchanged in basic physical definition and use. There are, however, minor modifications to the interpretation and use of some of the individual fields within these areas. Only these modifications are discussed here. See Interpartition Interface in Section 4 for a complete description of the standard use of fields in Common.

Communications Control Block

The definition and use of the CCB is unchanged except that the Inhibit Switch code R has a slightly different meaning. This difference does not change the functional significance of the code R

to the UAP. See Temporary Text Delay above for an understanding of the Inhibit Switch code R posted by a 4310 SCA Module.

Partition Status Byte

Three of the possible PSB codes are not applicable when communicating with a 4310 terminal and will not occur. They are:

- E -- length error,
- F -- sequence error,
- T -- transmission interrupt.

All other previously defined PSB codes may occur when communicating with a 4310 MDRS terminal.

System Mailbox

System Mailbox definition and use during initialization are unchanged.

Transmit Buffers

Since the 4310 Write Routine does not generate sequence number and length control fields, relative locations 12 - 19 of the Transmit Buffer Prefix have been changed. The definition and use of the Transmit Buffer by the UAP is unchanged.

Standard Transmit Buffer Format:

Buffer Status Byte	To-Address		From-Address		Data Length				Write Flag	Unused		Synchronization Characters (SYNs)			STX or SOH	Sequence Number		Length Control Field		Data
	1	2	3	4	5	6	7	8		10	11	12	13	14		16	17	18	19	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

See Transmit and Receive Buffers in Section 4 for a description of the fields of the standard Transmit Buffer.

4310 Transmit Buffer Format:

Buffer Status Byte	To-Address		From-Address		Data Length				Write Flag	Unused					Synchronization Characters (SYNs)			STX or SOH	Data	
	1	2	3	4	5	6	7	8		10	11	12	13	14	15	16	17			18
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

(Changed)

Three additional codes may be posted in the Write Flag by the UAP in order to request remote device selection. These codes are covered under Remote Device Selection in this appendix.

The data area of the 4310 Transmit Buffer may contain device control sequences or data or both. When transmitting to the 4310, the number of data characters plus the number of control sequence characters in a single data block may not exceed 200. (Note: If the RECORD LENGTH switch on the 4310 terminal is set at less than 200, then the maximum number of characters that can be sent is correspondingly reduced.)

Receive Buffers

Since the 4310 terminal does not send sequence and length control fields, relative locations 15 - 19 of the Receive Buffer prefix have been changed. The definition and use of the Receive Buffer by the UAP is unchanged.

Standard Receive Buffer Format:

Buffer Status Byte	To-Address		From-Address		Data Length				Unused						STX or SOH	Sequence Number		Length Control Field		Data		ETB or ETX
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		

See Transmit and Receive Buffers in Section 4 for a description of the fields of the standard Receive Buffer.

4310 Receive Buffer Format:

Buffer Status Byte	To-Address		From-Address		Data Length				Unused						STX	Data		ETB or ETX				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		

(Changed)

Relative location 19 of the 4310 Receive Buffer will always contain an STX after a data block has been received from the 4310 terminal. The 4310 cannot send an SOH.

Idle Message Buffer

Although the 4310 Write Routine does not send IDLE Messages, the definition of the IDLE Message Buffer must be retained for proper assembly of the 4310 SCA Module. The area defined as the IDLE Message Buffer is available to the UAP for its own use (the SCA Module will not access or modify it), unless Dial-Out Handshaking has been selected, in which case the UAP must use the buffer to pass telephone numbers to the SCA Module.

LINE CONTROL CONVENTIONS

Normal SCA line control characters and conventions are discussed in Appendix A. The line control characters and conventions of the 4310 terminal are discussed in Publication No. 15-045. Only differences which affect UAP design are discussed here.

SOH -- Start of Header

The 4310 terminal cannot send an SOH. If it receives an SOH, it treats it as an STX. Therefore there is no reason for the UAP to request in the Write Flag that a data block be preceded by an SOH instead of an STX.

ETX -- End of Text

The 4310 terminal sends an ETB (End of Text Block) character after each regular data block. When it has detected end of file on the tape, it sends a null data block (no data characters) terminated by an ETX instead of an ETB. The UAP can recognize this null data block by detecting a zero in the Data Length field of the Receive Buffer Prefix. The UAP should post the buffer available, but it can otherwise ignore the null data block. The 4310 terminal will then send an EOT, and the SCA Module will post a W in the Inhibit Switch. This will indicate end of file to the UAP as in standard SCA Module usage.

When transmitting data to a 4310 terminal, the UAP must send an ETX before sending an EOT to indicate end of transmission. The ETX can be sent with a null data block after the last regular data block has been sent, or it can be sent with the last regular data block. The UAP requests that an ETX be sent in place of the standard ETB by moving an X to the Write Flag of the Transmit Buffer Prefix before posting the buffer ready. After the ETX has been transmitted, the UAP should request in the Write Flag that an EOT be sent.

Failure to send the ETX prior to sending an EOT will cause the 4310 terminal to indicate an error condition instead of lighting its End of File (EOF) lamp.

TI -- Transmission Interrupt (Called RVI -- Reverse Channel Interrupt in Publication No. 15-045)

When the System Ten is receiving data from the 4310 tape drive, the UAP can stop the 4310 tape in order to transmit to the 4310 printer by sending the 4310 terminal a TI. The UAP can request that a TI be sent by moving a 3 to the Read Flag in the CCB. The 4310 will stop transmitting and send an EOT. The UAP can then transmit data to the

4310 printer (if the printer is ready). When the UAP has completed its transmission to the printer (when it has requested in the Write Flag that an EOT be sent and the SCA Module has sent the EOT), the 4310 will continue transmission of its tape file.

The 4310 terminal cannot send a TI.

REMOTE DEVICE SELECTION

Data sent from the System Ten to the 4310 terminal normally is routed to the 4310 line printer or tape drive or both, depending on the setting of the PRINT/DUAL/WRITE TAPE switch on the terminal. The PRINT setting routes data to the line printer only; the WRITE TAPE setting routes data to the tape drive only; and the DUAL setting routes data to both simultaneously.

The UAP can override the PRINT or WRITE TAPE setting of the switch by requesting device selection. The UAP requests device selection by moving a code 1, 2, or 3 to the Write Flag of the Transmit Buffer of the first data block of the transmission. The Write Flag codes have the following meanings:

CODE	MEANING
1 or 3	Route this data to the line printer, even if the PRINT/DUAL/WRITE TAPE switch is set to WRITE TAPE.
2	Route this data to the tape drive, even if the PRINT/DUAL/WRITE TAPE switch is set to PRINT.

It is not possible to select both the line printer and the magnetic tape drive from the System Ten. However, if the PRINT/DUAL/WRITE TAPE switch is set to DUAL, data will be routed to both the printer and the tape drive, regardless of any device selection requested by the UAP.

Once a device has been selected, all data will be routed to that device until the System Ten sends an EOT to the 4310 terminal.

If the selected device is not ready, the 4310 terminal will send an EOT response. The 4310 Write Routine will post a Q in the Inhibit Switch in the CCB and return to the Handshaking Routine. The following conditions will cause this to happen:

- An attempt to select the tape drive when the tape drive is not ready or when the 4310 terminal is in the DATA SEND mode.
- An attempt to select the line printer when it is not in an ON-LINE condition.

REMOTE DEVICE CONTROL

The System Ten programmer can remotely control vertical positioning of forms in the 4310 line printer by inserting control sequences (device control flags and device control characters) in data blocks sent to the terminal. The forms control functions are performed immediately when transmitting to the 4310 line printer. When transmitting to the tape drive, the control sequences are recorded, as received, on the magnetic tape; the control functions are not actually performed until off-line printing of the data from the tape.

The control sequences used for line printer forms control are the same for remote transmission as for local (off-line) recording. The valid control sequences are shown in figure 6-1.

Each control sequence begins with the flag character \ (reverse slash). The reverse slash always indicates the beginning of a control sequence. Therefore it cannot be used as a data character in a data block that is to be printed.

The 4310 line printer automatically advances forms one line at the end of each data block, and after encountering and printing 132 consecutive data characters (no control sequences) in a data block. However, in either of these cases, if an "after printing" control sequence is imbedded in the data characters of the last line printed, the control function is performed instead of the automatic single-line feed.

The control sequences do not print on the line printer, but they do occupy positions in the 4310 buffer. Therefore they reduce the number of data characters that can be sent to the 4310 in a single data block. Data block size restrictions are discussed under Transmit Buffers in this appendix.

LINE CONTROL SEQUENCE (INTERNAL SYSTEM TEN REPRESENTATION)	4310 KEYBOARD OR KEYPUNCH * EQUIVALENT	USASCII EQUIVALENT	EXTENDED CHARACTER SET EQUIVALENT	ACTION
\N	0-8-2 - N	LF	J	New Line: space one line immediately.
\K	0-8-2 - K	FF	L	Form Feed: skip to top of next page immediately.
\A	0-8-2 - A	ESC A	[A	Skip to top of form: after printing.
\Q	0-8-2 - Q	ESC Q	[Q	Single Space: after printing.
\R	0-8-2 - R	ESC R	[R	Double Space: after printing.
\S	0-8-2 - S	ESC S	[S	Triple Space: after printing.

* - is equivalent to the multi-punch 11-7-8 on the keypunch machine.

Figure B-1. System Ten 4310 Device Control Sequences

Although data received from the 4310 terminal may contain device control sequences, the 4310 SCA Module cannot recognize them. They may, however, have meaning to the UAP.

Column 1 of Figure B-1 shows the internal System Ten representation of the control sequences. However, if the Extended Character Set feature on the terminal is enabled, they will appear in System Ten memory as shown under Extended Character Set Equivalent in the figure. If the Extended Character Set feature on the terminal is enabled, a 200-character block beginning with the control sequence N will be a 199-character block beginning with the character J after it has been transmitted from the terminal to System Ten memory.

If control sequences are to be transmitted from the terminal to the System Ten, it is recommended that the Extended Character Set feature be disabled in order to facilitate the recognition of the control sequences by the UAP. When transmitting from the System Ten to the 4310 terminal, the Extended Character Set feature has no effect.

Figure B-2 shows sample data blocks containing data to be printed on a 4310 terminal line printer and control sequences to format the data. Figure B-3 shows the data as it will print on the 4310 line printer.

The data blocks shown in figure B-2 are moved to the Data area (positions 20 - 219) of the Transmit Buffer and then are transmitted as in standard SCA Module usage. They can be transmitted directly to the 4310 line printer, or they can be transmitted to the 4310 tape drive and then later printed off-line from the tape. The printout will be the same in either case.

Notes: Figures B-2 and B-3

- 1 - No device control action is requested at the beginning of the first data block. It is assumed that the paper is already positioned at top of form.
- 2-3 \ - Double space immediately.
- 4 \- 5 The 4310 hardware will automatically cause single spacing at the end of the data block.
- 6 \- 7-10 \ - Double space immediately.
- 11 \<^R - Double space instead of performing the automatic single spacing described in Note 5.
- 12-15 \ - Double space immediately.
- 16 \<^R - Same as Note 11, except that the control sequence precedes the data to be printed.
- 17 \ - Triple space immediately.
- 18 \ - Double space immediately.
- 19 \<^A - Skip to the top of the next form instead of performing automatic single spacing.

① WAREHOUSE 1 ORDER PREPARATION LIST										② PAGE 0099 ORDER NO. 767202 : SHEET 1 OF 1										09/21/71																													
③ FLOOR AISLE BIN PART NO. ORDERED										④ FILLED										⑤ -----																													
⑥ 3 B 11 98-972										⑦ 10										⑧ 342 43-466 100										399																			
92-980										⑨ F 287 76-412										⑩ 23										201 49-001 1										⑪ R									
2 B 34 97-760										⑫ 54										⑬ D 216 02-743 32										119 19																			
-642										⑭ F 3 27-307										⑮ 150										⑯ 1 C 57 62-901 29																			
G 10 33-333										⑰ 47										⑱ * * * END OF ORDER * * *										MOVE TO TRUCK LOADING DOC																			
K NO. 3										⑲																																							

Figure B-2 Data Blocks Containing Data and Control Sequences for Printing on a 4310 Terminal

①	WAREHOUSE 1 ORDER PREPARATION LIST										PAGE 0099									
②	ORDER NO. 767202 : SHEET 1 OF 1										09/21/71									
③	FLOOR AISLE BIN PART NO. ORDERED										FILLED									
④	-----																			
⑤	3	B	11	98-972	10															
⑦					342	43-466	100													
⑧					399	92-980	8													
⑨			F	287	76-412	23														
⑩					201	49-001	1													
⑪	2	B	34	97-760	54															
⑫					D	216	02-743	32												
⑬					119	19-642	5													
⑭			F	3	27-307	150														
⑮	1	C	57	62-901	29															
⑯			G	10	33-333	47														
⑰	* * * END OF ORDER * * *																			
⑱	MOVE TO TRUCK LOADING DOCK NO. 3																			

Figure B-3 Printout of Data Blocks shown in Figure B-2

