Operation and Service Manual

Mainframe

SIM900

SRS Stanford Research Systems

Revision 1.3 ● August 8, 2003

Certification

Stanford Research Systems certifies that this product met its published specifications at the time of shipment.

Warranty

This Stanford Research Systems product is warranted against defects in materials and workmanship for a period of one (1) year from the date of shipment.

Service

For warranty service or repair, this product must be returned to a Stanford Research Systems authorized service facility. Contact Stanford Research Systems or an authorized representative before returning this product for repair.

Information in this document is subject to change without notice.

Stanford Research Systems, Inc.
1290–C Reamwood Avenue
Sunnyvale, CA 94089 USA
Phone: (408) 744-9040 • Fax: (408) 744-9049
www.thinkSRS.com • e-mail: info@thinkSRS.com

Printed in U.S.A.

Contents

General Information

Safety and Preparation for Use

WARNING! **Dangerous voltages, capable of causing injury or death, are present in this instrument. Use extreme caution whenever the instrument covers are removed. Do not remove the covers while the unit is plugged into a live outlet.**

Line Voltage

The universal input power supply of the SIM900 accommodates any voltage in the range 90 VAC to 260 VAC, with a frequency in the range 47 Hz to 63 Hz.

Line Fuse

The line fuse is internal to the SIM900 and may not be serviced by the user. If the *Standby* LED does not turn on when line power is provided, contact Stanford Research Systems.

Line Cord

The SIM900 has a detachable, three-wire power cord for connection to the power source and to a protective ground. The exposed metal parts of the instrument are connected to the outlet ground to protect against electrical shock. Always use an outlet which has a properly connected protective ground.

Service

Do not attempt to service or adjust this instrument unless another person, capable of providing first aid or resuscitation, is present.

Do not install substitute parts or perform any unauthorized modifications to this instrument. Contact the factory for instructions on how to return the instrument for authorized service and adjustment.

The SIM900 Mainframe is not intended for hot-swapping applications. Be certain to switch power to Standby before inserting or ejecting modules in the mainframe. Do not connect a module to the remote port while power is on.

Symbols you may Find on SRS Products

Symbol	Description
⏦	Alternating current
⚡	Caution - risk of electric shock
⊥	Frame or chassis terminal
⚠	Caution - refer to accompanying documents
⏚	Earth (ground) terminal
—|‖—	Battery
⏛	Fuse
|	On (supply)
○	Off (supply)

Notation

The following notation will be used throughout this manual:

- Front-panel indicators are set as *Overload*.
- Remote command names are set as *IDN?.
- Literal text other than command names is set as `OFF`.

Specifications

Performance Characteristics

Power Supplies	Voltages	±15 VDC, ±5 VDC, +24 VDC
	Regulation	±0.5 % (±15 V, ±5 V)
		±2 % (+24 V)
	Current limits	5 A max (+5 V)
		3 A max (all others)
	Power limit	70 W total, all voltages
Timebase	Internal timebase	10 MHz VCXO, ±10 ppm
	External connector	Rear panel BNC
	External input	10 MHz, 1 V to 5 V pp
	Capture range	±50 ppm (±500 Hz)
Interfaces	SIM ports	8 internal + 1 Remote DB–15 (female)
	Aux. RS-232	2; DB–9 (male) DTE
	Host interface	RS-232; DB–9 (female) DCE
		GPIB (optional)
	Eavesdrop	RS-232; DB–9 (female) DCE
Indicator Lights	Interface	RS-232, GPIB
	Timebase	Internal, External Lock, External Fault
	Activity	8 slots, Remote SIM, Aux A, Aux B, Mainframe, Data Send, Data Receive, Data Error
	Startup Script	Enabled
	Power	On, Standby, Overload, Trip
Operating	Temperature	0 °C to 40 °C, non-condensing
	Power	90 VAC to 260 VAC, 47 Hz to 63 Hz
		150 W max

General Characteristics

Weight	12.6 lbs
Dimensions	17.0″ W × 5.3″ H × 9.0″ D
Rack mount	O900RM (optional)

1 Operation

This chapter describes the operation of the SIM900 Mainframe.

In This Chapter

1.1 Introduction to the Instrument

The SIM900 Mainframe is the platform on which a SIM system is assembled. The mainframe provides power, computer interfaces, clock synchronization, and individual module status.

1.1.1 Front Panel

The SIM900 front panel consists of a power switch and a collection of indicator lamps (see Figures 1.1).

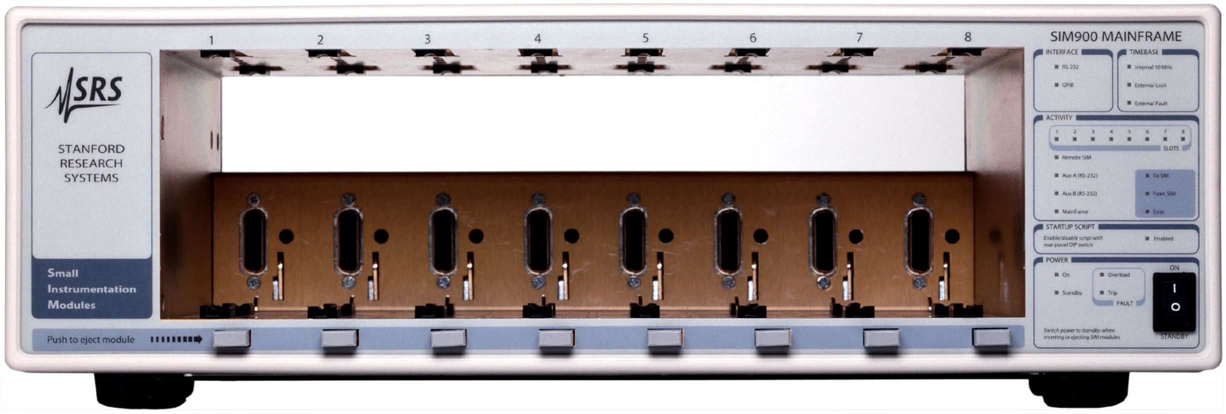

Figure 1.1: The SIM900.

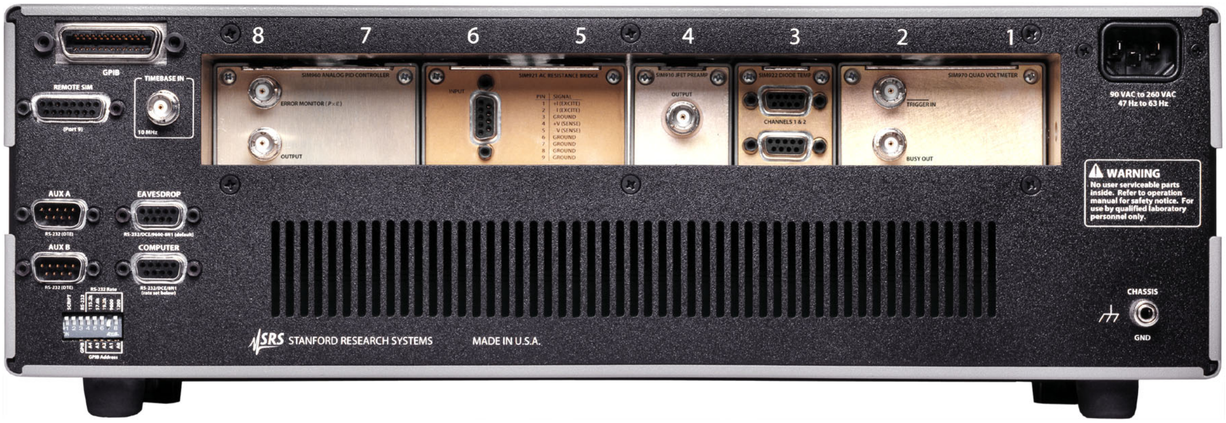
8
7
6
5
4
3
2
1
GPIB
REMOTE SIM
TIMEBASE IN
10 MHz
AUX A
AUX B
EAVESDROP
COMPUTER
90 VAC to 260 VAC 47 Hz to 63 Hz
WARNING
No user serviceable parts inside. Refer to operation manual for safety notice. For use by qualified laboratory personnel only.
CHASSIS
GND
SRS STANFORD RESEARCH SYSTEMS
MADE IN U.S.A.

Figure 1.2: The SIM900 rear panel (shown with a full complement of SIM modules).

1.1.2 Rear Panel

The SIM900 rear panel is shown in Figure 1.2. In addition to the host interface connector(s) and power entry module, the rear panel contains an external timebase reference input, auxiliary RS-232 interfaces, and a window for access to SIM module rear panels.

1.2 Getting Started

The SIM architecture does not support hot insertion or extraction of modules. Before installing or removing any SIM modules, the mainframe power must be switched to "Standby."

To install a module, align the back of the module with the black guide-ramps in the mainframe slot. Ease the module in until the connector begins to mate. Be careful to not apply pressure directly on any module display; it is better to push along the upper part of the edge of the module side covers until a positive "click" is heard. At this time, the module will be fully mated and locked in place.

To remove a module, press firmly on the ejection button below the module slot. When ejecting a double-wide module, the left-hand button must be pressed to eject.

When the arrangement of desired SIM modules is installed, turn on the mainframe power switch to begin operation.

1.3 Timebase

The SIM900 Mainframe provides a common 10 MHz clock reference to the SIM modules. By synchronizing clocks, low-frequency mixing products (beat tones) of independently running module clocks is avoided. A common timebase also allows precision time and frequency modules to be synchronized.

In a laboratory employing multiple SIM900's or where a high-precision clock reference is desired, this feature can be extended by synchronizing the mainframe(s) to an external 10 MHz reference. An auto-detect circuit senses the presence of an AC signal at the TIMEBASE IN connector on the rear panel, and attempts to phase-lock the internal oscillator to the applied signal. The TIMEBASE block of the SIM900 front panel (see Figure 1.3) indicates the clock status as one of three states:

Internal 10 MHz : No signal is detected at the timebase input, and the SIM900 internal oscillator is being used.

External Lock : The SIM900 detected an external clock reference and successfully phase-locked to it.

External Fault : An external clock signal was detected, but the SIM900 failed to phase-lock to it.

The phase-locked loop has a capture range of ±10 ppm (±100 Hz), and should lock reliably with input signals of 1 V to 5 V peak-to-peak amplitude.

1.4 Configuration Switches

The rear panel DIP switches (see Figure 1.3) provide basic configuration of the SIM900 Mainframe host interface. The switches are read only at power-up time, but may be changed at any time.

1.4.1 Baud Rate/GPIB Address

The rightmost five (5) switches program either the default baud rate for the "COMPUTER" RS-232 port, or the GPIB address.

The RS-232 default baud rate can be set to 1200, 9600, 19.2k, 57.6k, or 115.2k. Select *one* rate by setting that switch in the down position; if no switch is selected or more than one is down, the SIM900 defaults to 9600 baud and *Error* lights for several seconds after power-up. The host baud rate can be changed after power-up under remote program control (see the BAUD command), but will revert to the rear-panel default after power cycling or a Device Clear (RS-232 ⟨break⟩) signal.

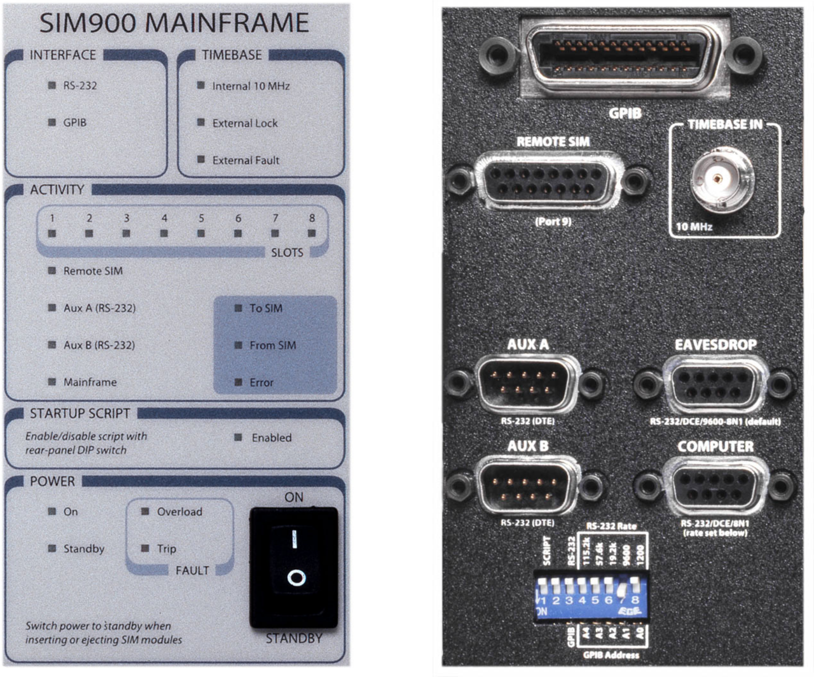

Figure 1.3: The SIM900 front and rear panels.

For GPIB, the 5 switches set the binary-encoded bus address for the SIM900. To add 2^n to the address, set switch An in the lower position.

For example, to set the GPIB address to 19 (= 16 + 2 + 1) set switches A0, A1, and A4 down (on).

1.4.2 Host Select

The next switch to the left of the Rate/Address field selects the mainframe host interface: up selects RS-232, down selects the (optional) GPIB interface.

1.4.3 Startup Script Enable

The SIM900 has a 4000 byte non-volatile memory to store a sequence of remote commands to execute upon power-up (the "Startup Script"). In order for a stored script to execute automatically after power-up, the left-most switch must be in the on (down) position.

When the script is enabled and executed, the front-panel indicator flashes and then remains lit.

See Section 2.5.7 for the remote commands to configure the startup script.

1.5 Activity Monitors

The ACTIVITY section of the front panel monitors data transfer to and from the mainframe. When bytes are received from any port, the corresponding port indicator (*1–8*, *Remote SIM*, *Aux A (RS-232)* or *Aux B (RS-232)*) flashes together with *From SIM*. When the mainframe transmits data to the host interface, *Mainframe* and *From SIM* both flash as well.

When data is received at the mainframe host interface or transmitted to one of the ports, *To SIM* flashes, along with the corresponding port indicator.

If a communication error is encountered, *Error* will flash briefly.

1.6 SIM Interface Connector

The DB–15 SIM Interface connector carries all the power and communications lines between the mainframe and SIM modules. The module-side of the interface is DB–15 male (plug), while the mainframe side is DB–15 female (socket). The connector signals are specified in Table 1.1

Pin	Signal	Direction Src ⇒ Dest	Description
1	SIGNAL_GND	MF ⇒ SIM	Ground reference for signal
2	−STATUS	SIM ⇒ MF	Status/service request (GND=asserted, +5V=idle)
3	RTS	MF ⇒ SIM	HW Handshake (+5 V=talk; GND=stop)
4	CTS	SIM ⇒ MF	HW Handshake (+5 V=talk; GND=stop)
5	−REF_10MHZ	MF ⇒ SIM	10 MHz reference
6	−5V	MF ⇒ SIM	Power supply (fast analog circuitry)
7	−15V	MF ⇒ SIM	Power supply (analog circuitry)
8	PS_RTN	MF ⇒ SIM	Power supply return
9	CHASSIS_GND		Chassis ground
10	TXD	MF ⇒ SIM	Async data (start bit="0"=+5 V; "1"=GND)
11	RXD	SIM ⇒ MF	Async data (start bit="0"=+5 V; "1"=GND)
12	+REF_10MHz	MF ⇒ SIM	10 MHz reference
13	+5V	MF ⇒ SIM	Power supply (digital & fast analog circuitry)
14	+15V	MF ⇒ SIM	Power supply (analog circuitry)
15	+24V	MF ⇒ SIM	Power supply (power circuitry)

Table 1.1: SIM Interface connector pin assignments, DB–15

Note that all SIM modules are specified to operate with or without the presence of the ±REF_10MHZ signals, so these lines are optional in any cabling interface between the mainframe REMOTE SIM port and a module.

2 Remote Programming

This chapter describes how to control the SIM900 Mainframe, and any connected SIMs or generic RS-232 devices, from a host computer.

In This Chapter

2.1 Index of Commands

symbol	definition
p	Port number
i,j	Integers
z	Literal token
b	Multi-byte (string) block
(*?*)	Required for queries; illegal for set commands
var	Parameter always required
{*var* }	Required parameter for set commands; illegal for queries
[*var*]	Optional parameter for both set and query forms

Communications

CONN *p*,*b*	2 – 15	Connect to Port
SEND *p*,*b* [,*i*]	2 – 15	Send Message to Port
SNDT *p*,*b* [,*i*]	2 – 15	Send Terminated Message to Port
ECHO? *b*	2 – 15	Echo Message back to Host
BRDC *b* [,*i*]	2 – 15	Broadcast Message to Ports
BRDT *b* [,*i*]	2 – 16	Broadcast Terminated Message to Ports
GETN? *p*,*i*	2 – 16	Get Bytes from Port
RAWN? *p*,*i*	2 – 16	Get Raw Bytes from Port

Configuration

NINP? *p*	2 – 16	Input Bytes Waiting
NOUT? *p*	2 – 16	Output Bytes Waiting
AINP? *p*	2 – 17	Input Spaces Available
AOUT? *p*	2 – 17	Output Spaces Available
DONE? [*p*]	2 – 17	Transmit Complete
BRER(?) [*p*,] {*i* }	2 – 17	Broadcast Enable
RDDR(?) [*p*,] {*i* }	2 – 17	Receive Data Disable
RPER(?) [*p*,] {*i* }	2 – 17	Receive Pass-Through Enable
MSGL(?) {*i* }	2 – 18	Maximum `MSG` Length
TMOT(?) *p*,{*i* }	2 – 18	Timeout
TERM(?) *p*,{*z* }	2 – 18	Message Termination
CEOI(?) {*z* }	2 – 19	Generate EOI on ⟨LF⟩
EOIX(?) {*z* }	2 – 19	EOI conversion during CONNect

Eavesdropping

VERB(?) [*i*],{*j* }	2 – 19	Verbosity
EAVS *b*	2 – 20	Echo Message to Eavesdrop
EIDN	2 – 20	Identify to Eavesdrop

Serial Configuration

PRTC(?) {*z* }	2 – 21	Port C Function
PRTD(?) {*z* }	2 – 21	Port D Function
BAUD(?) *p* {,*i* }	2 – 21	Baud Rate

FLOW(?) *p* {,*z* }	2 – 21	Flow Control
PARI(?) *p* {,*z* }	2 – 21	Parity
WORD(?) *p* {,*i* }	2 – 22	Word Length
SBIT(?) *p* {,*i* }	2 – 22	Stop Bits

Status

SSCR? [*p*]	2 – 22	SIM Status Condition
SSPT(?) [*p*,] {*i* }	2 – 22	SIM Status Positive Transition
SSNT(?) [*p*,] {*i* }	2 – 22	SIM Status Negative Transition
SSEV? [*p*]	2 – 23	SIM Status Event
SSEN(?) [*p*,] {*i* }	2 – 23	SIM Status Enable
CESR? [*p*]	2 – 23	Comm Error Status
CESE(?) [*p*,]{*i* }	2 – 23	Comm Error Status Enable
TOSR? [*p*]	2 – 23	Timeout Status
TOSE(?) [*p*,] {*i* }	2 – 23	Timeout Status Enable
IOSR? [*p*]	2 – 23	Input Overflow Status
IOSE(?) [*p*,] {*i* }	2 – 23	Input Overflow Status Enable
FCSR? [*p*]	2 – 24	Flow Control Status
FCSE(?) [*p*,] {*i* }	2 – 24	Flow Control Status Enable
CTCR? [*p*]	2 – 24	CTS Status Condition
CTSR? [*p*]	2 – 24	CTS Status
CTSE(?) [*p*,] {*i* }	2 – 24	CTS Status Enable
PDPR? [*p*]	2 – 24	Port Data Pending
PDPE(?) [*p*,] {*i* }	2 – 24	Port Data Pending Enable

Script

APSS *b*	2 – 25	Append to Script
ATSS *b*	2 – 25	Append Terminated to Script
ERSS	2 – 25	Erase Script
LKSS(?) {*z* }	2 – 25	Lock Script
LISS?	2 – 25	List Script
NBSS?	2 – 25	Bytes Used in Script
AVSS?	2 – 25	Space Available in Script
ENSS?	2 – 25	Script Enable
RNSS	2 – 26	Run Script

Housekeeping

TBIN?	2 – 26	Timebase Input Detect
VTBI?	2 – 26	Timebase Input Analog
PLLC(?) {*z* }	2 – 26	Timebase Control
LOCK?	2 – 26	Timebase Status
VLOC?	2 – 26	Timebase Status Analog
VVCO?	2 – 26	Timebase VCO
VMON?	2 – 26	Primary Voltage
IMON?	2 – 27	Primary Current

PMON?	2 – 27	Primary Power
UNDV?	2 – 27	Undervoltage
TICK?	2 – 27	Elapsed Time
DIPS? [*i*]	2 – 27	DIP Switch

Interface

*RST	2 – 28	Reset
FLOQ	2 – 28	Flush Output Queue
SRST [*p*]	2 – 28	SIM Reset
FLSI [*p*]	2 – 28	Flush Port Input Buffers
FLSO [*p*]	2 – 28	Flush Port Output Queues
FLSH [*p*]	2 – 29	Flush Port Buffers
*IDN?	2 – 29	Identify
*TST?	2 – 29	Self Test
*CLS	2 – 29	Clear Status
*STB? [*i*]	2 – 29	Status Byte
*SRE(?) [*i*,] {*j* }	2 – 29	Service Request Enable
*ESR? [*i*]	2 – 29	Standard Event Status
*ESE(?) [*i*,] {*j* }	2 – 29	Standard Event Status Enable
*PSC(?) {*i* }	2 – 29	Power-on Status Clear
*OPC(?)	2 – 30	Operation Complete
*WAI	2 – 30	Wait to Continue
CONS(?) {*z* }	2 – 30	Console Mode
WAIT *i*	2 – 30	Wait
REQT(?) {*z* }	2 – 30	Announce REQT
REQF(?) {*z* }	2 – 30	Announce REQF
LEXE?	2 – 31	Execution Error
LCME?	2 – 31	Command Error
TOKN(?) {*z* }	2 – 32	Token Mode

2.2 Alphabetic List of Commands

★

*CLS	2 – 29	Clear Status
*ESE(?) [_i_,] {_j_ }	2 – 29	Standard Event Status Enable
*ESR? [_i_]	2 – 29	Standard Event Status
*IDN?	2 – 29	Identify
*OPC(?)	2 – 30	Operation Complete
*PSC(?) {_i_ }	2 – 29	Power-on Status Clear
*RST	2 – 28	Reset
*SRE(?) [_i_,] {_j_ }	2 – 29	Service Request Enable
*STB? [_i_]	2 – 29	Status Byte
*TST?	2 – 29	Self Test
*WAI	2 – 30	Wait to Continue

A

AINP? _p_	2 – 17	Input Spaces Available
AOUT? _p_	2 – 17	Output Spaces Available
APSS _b_	2 – 25	Append to Script
ATSS _b_	2 – 25	Append Terminated to Script
AVSS?	2 – 25	Space Available in Script

B

BAUD(?) _p_ {,_i_ }	2 – 21	Baud Rate
BRDC _b_ [,_i_]	2 – 15	Broadcast Message to Ports
BRDT _b_ [,_i_]	2 – 16	Broadcast Terminated Message to Ports
BRER(?) [_p_,] {_i_ }	2 – 17	Broadcast Enable

C

CEOI(?) {_z_ }	2 – 19	Generate EOI on ⟨LF⟩
CESE(?) [_p_,]{_i_ }	2 – 23	Comm Error Status Enable
CESR? [_p_]	2 – 23	Comm Error Status
CONN _p_,_b_	2 – 15	Connect to Port
CONS(?) {_z_ }	2 – 30	Console Mode
CTCR? [_p_]	2 – 24	CTS Status Condition
CTSE(?) [_p_,] {_i_ }	2 – 24	CTS Status Enable
CTSR? [_p_]	2 – 24	CTS Status

D

DIPS? [_i_]	2 – 27	DIP Switch
DONE? [_p_]	2 – 17	Transmit Complete

E

EAVS *b*	2 – 20	Echo Message to Eavesdrop
ECHO? *b*	2 – 15	Echo Message back to Host
EIDN	2 – 20	Identify to Eavesdrop
ENSS?	2 – 25	Script Enable
EOIX(?) {*z* }	2 – 19	EOI conversion during CONNect
ERSS	2 – 25	Erase Script

F

FCSE(?) [*p*,] {*i* }	2 – 24	Flow Control Status Enable
FCSR? [*p*]	2 – 24	Flow Control Status
FLOQ	2 – 28	Flush Output Queue
FLOW(?) *p* {,*z* }	2 – 21	Flow Control
FLSH [*p*]	2 – 29	Flush Port Buffers
FLSI [*p*]	2 – 28	Flush Port Input Buffers
FLSO [*p*]	2 – 28	Flush Port Output Queues

G

GETN? *p*,*i*	2 – 16	Get Bytes from Port

I

IMON?	2 – 27	Primary Current
IOSE(?) [*p*,] {*i* }	2 – 23	Input Overflow Status Enable
IOSR? [*p*]	2 – 23	Input Overflow Status

L

LCME?	2 – 31	Command Error
LEXE?	2 – 31	Execution Error
LISS?	2 – 25	List Script
LKSS(?) {*z* }	2 – 25	Lock Script
LOCK?	2 – 26	Timebase Status

M

MSGL(?) {*i* }	2 – 18	Maximum `MSG` Length

N

NBSS?	2 – 25	Bytes Used in Script
NINP? *p*	2 – 16	Input Bytes Waiting
NOUT? *p*	2 – 16	Output Bytes Waiting

P

PARI(?) *p* {,*z* }	2 – 21	Parity
PDPE(?) [*p*,] {*i* }	2 – 24	Port Data Pending Enable

PDPR? [*p*]	2 – 24	Port Data Pending
PLLC(?) {*z*}	2 – 26	Timebase Control
PMON?	2 – 27	Primary Power
PRTC(?) {*z*}	2 – 21	Port C Function
PRTD(?) {*z*}	2 – 21	Port D Function

R

RAWN? *p*,*i*	2 – 16	Get Raw Bytes from Port
RDDR(?) [*p*,] {*i*}	2 – 17	Receive Data Disable
REQF(?) {*z*}	2 – 30	Announce REQF
REQT(?) {*z*}	2 – 30	Announce REQT
RNSS	2 – 26	Run Script
RPER(?) [*p*,] {*i*}	2 – 17	Receive Pass-Through Enable

S

SBIT(?) *p* {,*i*}	2 – 22	Stop Bits
SEND *p*,*b* [,*i*]	2 – 15	Send Message to Port
SNDT *p*,*b* [,*i*]	2 – 15	Send Terminated Message to Port
SRST [*p*]	2 – 28	SIM Reset
SSCR? [*p*]	2 – 22	SIM Status Condition
SSEN(?) [*p*,] {*i*}	2 – 23	SIM Status Enable
SSEV? [*p*]	2 – 23	SIM Status Event
SSNT(?) [*p*,] {*i*}	2 – 22	SIM Status Negative Transition
SSPT(?) [*p*,] {*i*}	2 – 22	SIM Status Positive Transition

T

TBIN?	2 – 26	Timebase Input Detect
TERM(?) *p*,{*z*}	2 – 18	Message Termination
TICK?	2 – 27	Elapsed Time
TMOT(?) *p*,{*i*}	2 – 18	Timeout
TOKN(?) {*z*}	2 – 32	Token Mode
TOSE(?) [*p*,] {*i*}	2 – 23	Timeout Status Enable
TOSR? [*p*]	2 – 23	Timeout Status

U

UNDV?	2 – 27	Undervoltage

V

VERB(?) [*i*],{*j*}	2 – 19	Verbosity
VLOC?	2 – 26	Timebase Status Analog
VMON?	2 – 26	Primary Voltage
VTBI?	2 – 26	Timebase Input Analog
VVCO?	2 – 26	Timebase VCO

2.3 Introduction

The SIM900 Mainframe provides fully buffered multiplexed communications between the host computer and up to 9 SIM modules plus 2 (optionally as many as 4) external RS-232 devices. These SIM/RS-232 connections are generically called Ports here, and each port has a dedicated UART (universal asynchronous receiver & transmitter) with hardware input and output FIFO buffers. The host computer (typically a PC) communicates with the mainframe through the host interface, which can be either RS-232 or (optionally) GPIB. The active interface is selected with rear-panel DIP switches at power-on.

No protocol requirements are placed on the communications across the ports—any sequence of bytes can be transmitted to or received from any port. Simultaneous ongoing communications with multiple ports can be maintained using a packet-message style of command (see SEND, BRDC, GETN?, RPER commands below). Simple host-to-port communications are provided with the CONN command.

2.3.1 Device Clear

If the host interface is GPIB, the IEEE–488 DCL (Device Clear) or SDC (Selected Device Clear) interface messages will cause the mainframe to flush the host input buffer and output queue, and reset the parser to the idle state.

RS-232 ⟨break⟩

If the host interface is RS-232, the same action is initiated by the RS-232 ⟨break⟩ signal (space level (0) for at least one full character frame). This single “out-of-band” signal allows the host to reset the mainframe interface to a known state, independent of the current operating mode.

In particular, a Device Clear event (either from DCL, SDC, or RS-232 ⟨break⟩) will cause the mainframe to abandon an active connect session (see CONN command).

2.3.2 Queues and buffers

Each port is separately buffered with a port input buffer and port output queue, while the host interface is buffered with the host input buffer and host output queue. All queues and buffers are 512 bytes deep.

Data is initially received from the host interface into the host input buffer. If the mainframe is not currently in connect mode, then bytes from the host input buffer are read by the parser until a valid command is found. Command Errors detected by the parser are reported through the CME flag in the ESR register. Mainframe-directed commands and queries are then handled directly, and responses (if any) transferred to the host output queue for the host computer to read.

2.4 Port Communications

2.4.1 Message-based communication

Port-directed messages SEND,SNDT and BRDC,BRDT are parsed for syntax, and then given to the Message Handler for delivery. The message payload is stripped out of the command, and copied to the appropriate port output queue(s) for delivery. If the port output queue is full (because the SIM or external RS-232 device has asserted flow control, or simply because of data rate mismatch), the mainframe will wait up to TMOT milliseconds until there is sufficient room in the port output queue for the data. In the meantime, commands and queries from the host will simply accumulate in the input buffer until that fills as well. At that point, flow control on the host interface should hold off any further transmissions from the host until the buffers clear up.

Data received from the ports is initially stored in the corresponding port input buffer. If the corresponding bit in the Receive Pass-through Enable Register (RPER) is set and there is sufficient room in the host output queue, then the data is wrapped into a MSG unit and transferred to the output queue for delivery to the host. If the output queue was too full, the message will be sent as soon as sufficient space becomes available. If the corresponding bit in RPER is clear, then the corresponding bit in the Port Data Pending Register (PDPR) is set.

2.4.2 Connection-based communication

If the mainframe is connected to a port via the CONN command, the situation is somewhat different. Bytes received from the host accumulate in the input buffer, where they are scanned for matching the escape string provided with CONN. Non-matching bytes are directly transferred to the port output queue. A partial match is held off until an unambiguous complete match or non-match is present. On a successful complete match, the connect mode is terminated and the mainframe is ready for new commands.

Concurrently, bytes received in the port input buffer are transferred directly to the host output queue. Data received at the unconnected ports will be held in their port input buffers (causing the corresponding bit(s) in the PDPR to be set).

2.4.3 Port map

The SIM900 Mainframe ports are defined in the following table:

Port	Type	Description
1	SIM	Slot 1 SIM port
2	SIM	Slot 2 SIM port
3	SIM	Slot 3 SIM port
4	SIM	Slot 4 SIM port
5	SIM	Slot 5 SIM port
6	SIM	Slot 6 SIM port
7	SIM	Slot 7 SIM port
8	SIM	Slot 8 SIM port
9	SIM	Remote SIM port (DB-15-F back panel connector)
A	RS-232	Aux–1 DTE (DB–9/M back panel connector)
B	RS-232	Aux–2 DTE (DB–9/M back panel connector)
C	RS-232	Eavesdrop DCE (DB–9/F back panel connector)
D	RS-232	COMM DCE (DB–9/F back panel connector)

Ports 1 through B are always available as user ports, with A & B as generic RS-232 ports. After power-on, ports 1–B default to 9600 baud, 8-bits, no parity, and 1 stop bit.

Port C (Eavesdrop) is normally dedicated to monitoring communications between the mainframe and host, but can be remapped as an additional general purpose port with the PRTC command. At power-on, this port defaults to 9600/8/N/1. Baud rate, parity, word size, and stop bits can be reconfigured by command after power-on, regardless of whether Port C is used for eavesdrop or general communications.

Port D (COMM) is normally dedicated as the RS-232 interface to the host computer. At power-on, this port defaults to the baud rate selected by the rear-panel DIP switches, or 9600 if the DIP settings are invalid (8/N/1). If RS-232 is NOT the active host interface, then Port D is normally inactive, but can be remapped as an additional general purpose port with the PRTD command. Baud rate, parity, word size, and stop bits can be reconfigured by command after power-on, regardless of whether Port D is used for the host interface or general communications.

When Port C or D are not reconfigured for general port communications, the corresponding PC and/or PD bits in the RPER, BER, and PDPR registers are undefined; they can be written or read, but will have no effect.

2.5 Commands

All commands for the SIM900 Mainframe originate at the host computer[1], and are sent to the mainframe via the host interface. The commands are organized according to functional groups, beginning with commands that directly control communications with the SIM modules. Other groups of commands configure the mainframe communications hardware, status reporting mechanism, startup script, internal housekeeping, and host interface.

2.5.1 Command syntax

All command names are 4-characters long and are case-insensitive. IEEE–488.2 defined commands begin with the "*" character followed by 3 letters, while SIM900-specific commands are composed of 4 letters.

The four letter mnemonic (shown in CAPS) in each command sequence specifies the command. The rest of the sequence consists of parameters.

Commands may take either *set* or *query* form, depending on whether the "?" character follows the mnemonic. *Set only* commands are listed without the "?", *query only* commands show the "?" after the mnemonic, and *optionally query* commands are marked with a "(?)".

Parameters shown in { } and [ ] are not always required. Parameters in { } are required to set a value, and are omitted for queries. Parameters in [ ] are optional in both set and query commands. Parameters listed without any surrounding characters are always required.

Do *not* send ( ) or { } or [ ] as part of the command.

The command buffer is limited to 255 bytes, with multi-byte block parameter bytes separately stored in an independent 255 byte buffer—any command that exceeds this size will generate a command error and be discarded.

If the host interface is RS-232, commands are terminated by either ⟨CR⟩ (ASCII 13) or ⟨LF⟩ (ASCII 10) characters that are outside any protected binary block or string. If the host interface is GPIB, then commands are terminated by either ⟨LF⟩ or ⟨EOI⟩ or ⟨LF-EOI⟩. Execution of the command does not begin until the command terminator is received.

Unlike most SIM modules*, no multi-command messages (i.e., ";" separated commands) are allowed for the SIM900.*

[1] or from the startup script

The following table summarizes the notation used in the command descriptions:

symbol	definition
p	Port number (`1–9`, `a–d`, `A–D`, but see below)
i,j	Integers
z	Literal token
b	Multi-byte (string) block
(*?*)	Required for queries; illegal for set commands
var	parameter always required
{*var*}	required parameter for set commands; illegal for queries
[*var*]	optional parameter for both set and query forms

2.5.1.1 Ports

Port parameters can be given as either simple decimal integers, or a single-letter hexadecimal value (*without* any leading `0x`).

Many of the commands to set/query a register accept an optional port parameter. In these cases, if the optional parameter *p* is given, then the command only sets/queries the single bit corresponding to the binary weight 2^p. Typically, this bit represents Port *p*, but in a few cases additional flag bits are packed into the register. For these additional flag bits, the optional *p* still restricts the command to the single bit, but it no longer corresponds to a port. Thus, it is possible in these cases for *p* to be `E`, despite the maximum port value of `D`.

2.5.1.2 Integers

Integer parameters follow "C-language" style. Simple decimal integers are indicated by beginning with a non-zero digit (`1–9`). Octal integers are represented with a leading zero digit (`0`). Hexadecimal integers are given by a leading `0x` or `0X`.

For example, `26`, `032`, `0x1A` all refer to the integer value 26.

2.5.1.3 Tokens

Tokens are listed here as word–integer pairs, such as `AUTO 2`. For set commands, token parameters must either be the exact text word indicated (case-insensitive), or the corresponding decimal integer code. For example, to set the response termination sequence to ⟨CR⟩+⟨LF⟩, the following two commands are equivalent:

TERM `CRLF` —or— TERM `3`

For queries that return token values, the return format (keyword or integer) is specified with the TOKN command.

2.5.1.4 Blocks

Multi-byte block parameters can follow one of 3 formats (4 on GPIB).

Quote-delimited strings : An arbitrary byte sequence bounded by either `"` or `'` characters. All characters (including control characters) are allowed. The quoting character itself (either `"` or `'`) can be included by escaping with an additional quote. For example,

```
"It is a ""good"" quote"
```

is identical to

```
'It is a "good" quote'.
```

Hex-formatted binary : `#Hxx xx xx`, where `xx` are hexadecimal bytes (00 through ff). Whitespace is ignored.

Definite-length arb. : `#abbbrrrr`, where `a` is a single non-zero digit equal to the digit count in `bbb`, `bbb` is a decimal integer count of the number of data bytes to follow, and `rrrr` are the raw data bytes.

Indefinite-length arb. : `#0rrrr`⟨LF-EOI⟩, where `rrrr` is the raw data block, and ⟨LF-EOI⟩ is the newline character (ASCII 10) with the GPIB-EOI line simultaneously asserted. (only on GPIB)

2.5.2 Communication commands

These commands provide the actual communication, through the SIM900 mainframe, between the host computer and the SIM modules or external RS-232 devices. Stream-style communications is provided with the CONN command, while packet messaging is supported with the remainder of the commands in this section.

CONN *p*,*b*

Connect to Port

The CONN command establishes a stream-style connection to Port *p*, with escape string *b*.

Executing the CONN command automatically clears the RPER register.

SEND *p*,*b* [,*i*]

Send Message to Port

The SEND command transfers the message *b* to Port *p*. If present, *i* contains a checksum for *b*.

The optional checksum is calculated as the unsigned integer sum of the ASCII value of each byte in *b* (excluding wrapping characters such as `#H`).

SNDT *p*,*b* [,*i*]

Send Terminated Message to Port

The SNDT command transfers the message *b* followed by the ⟨term⟩ sequence to Port *p*. If present, *i* contains the checksum value for *b* (see SEND).

See the TERM command for more about the ⟨term⟩ sequence.

ECHO? *b*

Echo Message back to Host

The ECHO command transfers the message *b*+⟨term⟩ back to the host output queue.

Note this command does not control character echoing back to a console terminal; see the CONS (console) command.

BRDC *b* [,*i*]

Broadcast Message to Ports

The BRDC command transfers the message *b* to multiple ports. If present, *i* is the checksum on *b*.

BRDC acts just like SEND, except instead of indicating a specific port in the command, all ports enabled in the BRER register receive a copy of the message *b*.

BRDT *b* [,*i*]

Broadcast Terminated Message to Ports

The BRDT command transfers the message *b*+⟨term⟩ to multiple ports. If present, *i* is the checksum on *b*.

GETN? *p*,*i*

Get Bytes from Port

The GETN command retrieves up to *i* bytes from Port *p* and transfers them to the host output queue.

Data is formatted as a definite-length arbitrary block, `#3aaabbbbbb`, where `aaa` = number of bytes actually retrieved from Port *p*. Response message may be up to 7 bytes longer than *i*, for the `#3aaa` header + ⟨term⟩.

RAWN? *p*,*i*

Get Raw Bytes from Port

The RAWN command retrieves *exactly* *i* bytes from Port *p* and transfers them to the host output queue.

No header or ⟨term⟩ characters are added. If fewer than *i* bytes are available, no bytes are transferred and the EXE flag is set with the ESR register.

2.5.3 Configuration commands

The first five Configuration commands query the current state of the port I/O buffers, while the remaining commands are used to set or query registers and other parameters controlling communications with the SIMs.

See the Register Model section for more about the mainframe registers.

NINP? *p*

Input Bytes Waiting

Query bytes waiting in Port *p* input buffer. Returns the integer number of bytes waiting to be read by the host.

NOUT? *p*

Output Bytes Waiting

Query bytes waiting in Port *p* output queue. Returns the integer number of bytes waiting to be transmitted to the SIM or RS-232 device.

AINP? *p*

Input Spaces Available

Query spaces available in Port *p* input buffer. Returns the integer number of additional bytes that can be written by the host before overflowing the port input buffer.

AOUT? *p*

Output Spaces Available

Query spaces available in Port *p* output queue. Returns the integer number of additional bytes that can be written by the SIM or RS-232 device before overflowing the port output queue.

DONE? [*p*]

Transmit Complete

DONE? returns `1` if there are no bytes remaining to be transferred, and `0` if any bytes remain in either port output buffers or in the UART output FIFO buffers.

If *p* is given, only the buffers for Port *p* are checked; otherwise all port buffers are checked.

BRER(?) [*p*,] {*i* }

Broadcast Enable

Set (query) the Broadcast Enable Register [Port *p* bit] {to *i*}. The Broadcast Enable Register (BER) selects which ports will receive broadcast messages via the BRDC and BRDT commands. If *p* is given, only the bit corresponding to Port *p* is set (queried); otherwise the entire register is indicated.

RDDR(?) [*p*,] {*i* }

Receive Data Disable

Set (query) the Receive Data Disable Register [Port *p* bit] {to *i*}. Bits within the Receive Data Disable Register (RDDR) are used to inhibit the serial receiver circuitry for the corresponding ports. If *p* is given, only the bit corresponding to Port *p* is set (queried); otherwise the entire register is indicated.

RPER(?) [*p*,] {*i* }

Receive Pass-Through Enable

Set (query) Receive Pass-Through Enable Register [Port *p* bit] {to *i*}. Bits within the Receive Pass-Through Enable Register (RPER) are used to enable spontaneous delivery of port data messages to the host output queue as `MSG` packets. If *p* is given, only the bit corresponding to Port *p* is set (queried); otherwise the entire register is indicated.

MSGL(?) {*i*}

Maximum `MSG` Length

Set (query) the maximum overall `MSG` length to *i* bytes. The data portion of MSG packets will have 10 or 11 fewer bytes than the maximum data block length *i*, corresponding to the characters `MSG p,#2yy` or `MSG p,#3yyy` that precede the data block. The command terminator (either ⟨CR⟩⟨LF⟩ or ⟨LF-EOI⟩) is not included in *i*.

After reset, the default is MSGL `64`. The maximum value is 128.

TMOT(?) *p*,{*i*}

Timeout

Set (query) the timeout value for Port *p* {to *i* milliseconds}. If TMOT is non-zero, the mainframe will wait up to *i* milliseconds while attempting to transfer output to a port output queue that is full (either due to flow control or simple transfer-rate mismatch). If the timeout expires before the mainframe is able to transfer the message, an error is recorded in the Timeout Status Register (TOSR).

If TMOT is set to zero (`0`), the timeout feature is disabled for Port *p*, and the mainframe will wait indefinitely when attempting to transfer a message to a full port output queue.

After reset, TMOT defaults to `1000` (1 second) for all ports.

TERM(?) *p*,{*z*}

Message Termination

Set (query) the ⟨term⟩ sequence for Port *p* {to *z*=`CR 0`, **`LF 1`**, `CRLF 2`, `LFCR 3`, `NONE 4`}.

The ⟨term⟩ sequence is appended to port messages sent with the SNDT or BRDT commands, and is constructed of ASCII character(s) 13 (carriage return) and 10 (line feed). The token mnemonic gives the sequence of characters. When the host interface is RS-232, then TERM `D` (the host port) also determines the termination for all mainframe-generated query responses (for GPIB host interface, query responses always terminate in ⟨LF-EOI⟩).

At power-on, ports default to TERM *p*,`LF`. After *RST, ports are reset to TERM *p*,`CR`. When the host interface is RS-232, Port D defaults to TERM `D,CRLF` both at power-on and *RST.

CEOI(?) {*z* }

Generate EOI on ⟨LF⟩

Set (query) the connect-mode EOI-on-⟨LF⟩ {to *z*=(**OFF 0**, ON 1)}.

CEOI controls whether the ⟨EOI⟩ signal is generated on the GPIB host interface whenever a ⟨LF⟩ character is received from a port during connect mode. This command has no effect when the host interface is RS-232.

Set *z* to OFF (0) to disable, ON (1) to enable.

After reset, the default is CEOI OFF.

EOIX(?) {*z* }

EOI conversion during CONNect

Set (query) the EOI-conversion-mode {to *z*=(**OFF 0**, ON 1)}.

In connect mode, EOIX controls whether the receipt of the ⟨EOI⟩ signal *from* the GPIB host interface causes a new ⟨LF⟩ character to transmitted to the connected port. This command has no effect when the host interface is RS-232.

Set *z* to OFF (0) to disable, ON (1) to enable.

After reset, the default is EOIX OFF.

2.5.4 Eavesdropping commands

By default, Port C is an eavesdropping monitor port. This port can be configured to provide real-time monitoring of communications and internal state changes in the SIM900 Mainframe, and is intended to help users build and debug their SIM applications.

VERB(?) [*i*],{*j* }

Verbosity

Set (query) the eavesdropping verbosity control [bit i] {to j}.

The verbosity control is an 8-bit register providing on/off control of separate eavesdropping monitor functions. The bit definitions are:

Weight	Bit	Flag
1	0	MFERRORS
2	1	LONGERRS
4	2	IOMON
8	3	MFMON
16	4	STATMON
32	5	FROMHOST
64	6	TOHOST
128	7	RTMON

MFERRORS : Error conditions in the mainframe will be reported.

LONGERRS : The long-form (English text) of error messages will be used. This flag is only functional if MFERRORS is also set.

IOMON : I/O status messages (device-clear, buffer overrun, . . .) will be reported.

MFMON : Internal state changes in the mainframe (such as timebase settings) will be reported. Also, if MFMON is set, any bytes *sent to Port C* will be interpreted as though sent by the mainframe, and processed as remote commands.

STATMON : Any (enabled) status register changes will be reported.

FROMHOST : All data received by the mainframe from the host interface is echoed to Port C.

TOHOST : All data sent by the mainframe to the host interface is echoed to Port C.

RTMON : A summary message of mainframe housekeeping data is reported once a second. The message is formatted as:

```
<vtbi=#, vloc=#, vvoc=#; vmon=#, imon=#>
```

where the value of each field corresponds with the query command of the same name (i.e., see VTBI?, VLOC?, . . .).

After reset, the default is VERB 5 (MFERRORS and IOMON).

EAVS *b* Echo Message to Eavesdrop

Send message *b* to the eavesdrop port.

EIDN Identify to Eavesdrop

Send the identify message (see *IDN) to the eavesdrop port.

2.5.5 Serial commands

The Serial commands control the configuration of the serial port hardware in the SIM900 Mainframe, such as baud rate and parity. Optional re-mapping of Port C and Port D is also provided with the PRTC and PRTD commands.

PRTC(?) {*z* }

Port C Function

Set (query) Port C function {to *z*=(**EAVS 0**, PORT 1)}.

Ordinarily, Port C is dedicated to eavesdropping on communications with the host computer. Using PRTC, it is possible to override this function and make Port C available as a general purpose RS-232 port. With PRTC PORT, general communications with Port C are performed using CONN, SEND, BRDC, etc., just as with other ports.

Set *z* to EAVS (0) for the default eavesdropping function, or PORT (1) for the general purpose port.

After reset, the default is PRTC EAVS.

PRTD(?) {*z* }

Port D Function

Set (query) Port D function {to *z*=(**COMM 0**, PORT 1)}.

Ordinarily, Port D is dedicated to communications with the host computer (when configured for host RS-232). If the host interface is GPIB, Port D is normally idle. Using the PRTD command with the GPIB interface, it is possible to make Port D available as a general purpose RS-232 port.

Set *z* to COMM 0 for the default host function, or PORT 1 for the general purpose port. Note that PRTD PORT will generate an execution error if the host interface is RS-232.

After reset, the default is PRTD COMM.

BAUD(?) *p* {,*i* }

Baud Rate

Set (query) Port *p* baud rate {to *i*}.

At power-on, all baud rates default to 9600.

FLOW(?) *p* {,*z* }

Flow Control

Set (query) Port *p* flow control {to *z*=(NONE 0, RTS 1, XON 2)}.

At power-on, all ports default to FLOW RTS flow control.

PARI(?) *p* {,*z* }

Parity

Set (query) Port *p* parity {to *z* = (NONE 0, ODD 1, EVEN 2, MARK 3, SPACE 4)}.

At power-on, all ports default to PARI NONE.

WORD(?) *p* {,*i* }

Word Length

Set(query) Port *p* word length {to *i* bits (5, 6, 7, or 8)}.

WORD sets the RS-232 word length to 5–8 bits. The set command is only valid for ports A-D, and will generate an execution error if *p*<A.

At power-on, all ports default to WORD 8.

SBIT(?) *p* {,*i* }

Stop Bits

Set(query) Port *p* stop bits {to *i* bits (1, or 2)}.

SBIT selects 1 or 2 stop bits for the RS-232 ports. The set command is only valid for ports A-D, and will generate an execution error if *p*<A.

If WORD 5 (5-bit word length) is set, then SBIT 2 corresponds to 1.5 stop bits.

At power-on, all ports default to SBIT 1.

2.5.6 Status commands

The Status commands query and configure registers associated with status reporting of SIMs and the mainframe.

See Section 2.6.2 for more about the mainframe status registers.

SSCR? [*p*]

SIM Status Condition

Query SIM Status Condition Register [for Port *p* bit].

SSCR? returns the present value of the STATUS signal.

SSPT(?) [*p*,] {*i* }

SIM Status Positive Transition

Set (query) SIM Status Positive Transition Register [Port *p*] {to *i*}.

SSNT(?) [*p*,] {*i* }

SIM Status Negative Transition

Set (query) SIM Status Negative Transition Register [Port *p*] {to *i*}.

SSPT and SSNT together define the enabled events (positive and negative transitions, respectively) that generate SIM Status Events.

At power-on, SSPT and SSNT are both cleared.

SSEV? [*p*]

SIM Status Event

Query SIM Status Event Register [for Port *p* bit].

Upon executing an SSEV? query, the returned bit(s) of the SSEV register are cleared.

SSEN(?) [*p*,] {*i* }

SIM Status Enable

Set(query) SIM Status Enable Register [for Port *p* bit] {to *i*}.

CESR? [*p*]

Comm Error Status

Query Comm Error Status Register [for Port *p* bit].

Upon executing a CESR? query, the returned bit(s) of the CESR register are cleared, with the exception of the TOSB bit (see the Register Model section for details).

CESE(?) [*p*,]{*i* }

Comm Error Status Enable

Set (query) Comm Error Status Enable Register [for Port *p* bit] {to *i*}

TOSR? [*p*]

Timeout Status

Query Timeout Status Register [for Port *p* bit].

Upon executing a TOSR? query, the returned bit(s) of the TOSR register are cleared.

TOSE(?) [*p*,] {*i* }

Timeout Status Enable

Set (query) Timeout Status Enable Register [for Port *p* bit] {to *i*}.

IOSR? [*p*]

Input Overflow Status

Query input Overflow Status Register [for Port *p* bit].

Upon executing a IOSR? query, the returned bit(s) of the IOSR register are cleared.

IOSE(?) [*p*,] {*i* }

Input Overflow Status Enable

Set (query) input Overflow Status Enable Register [for Port *p* bit] {to *i*}.

FCSR? [*p*]

Flow Control Status

Query Flow Control Status Register value [for Port *p* bit].

Upon executing a FCSR? query, the returned bit(s) of the FCSR register are cleared, with the exception of the CTSB bit (see the Register Model section for details).

FCSE(?) [*p*,] {*i* }

Flow Control Status Enable

Set (query) Flow Control Status Enable Register [Port *p* bit] {to *i*}.

CTCR? [*p*]

CTS Status Condition

Query CTS Status Condition Register [for Port *p* bit].

CTCR? returns the present value of the CTS signal [from Port *p*].

CTSR? [*p*]

CTS Status

Query CTS Status Register [for Port *p* bit].

Upon executing a CTSR? query, the returned bit(s) of the CTSR register are cleared.

CTSE(?) [*p*,] {*i* }

CTS Status Enable

Set (query) CTS Status Enable Register [for Port *p* bit] {to *i*}.

PDPR? [*p*]

Port Data Pending

Query Port Data Pending Register value [for Port *p* bit].

Upon executing a PDPR? query, the returned bit(s) of the PDPR register are cleared.

PDPE(?) [*p*,] {*i* }

Port Data Pending Enable

Set (query) Port Data Pending Enable Register [bit for Port *p*] {to *i*}.

2.5.7 Script commands

The Startup Script is a non-volatile block of memory that stores a series of commands for optional execution by the mainframe upon power-on. Any legal mainframe command, except for APSS, ATSS, ERSS, LKSS, and RNSS, can be included within the Startup Script.

Automatic execution of the Startup Script on power-on is controlled by the Startup Script Enable dip switch, on the rear panel of the mainframe.

APSS *b*

Append to Script

Append block *b* to Startup Script macro.

This command can only be executed after LKSS OFF.

ATSS *b*

Append Terminated to Script

Append block *b*+⟨LF⟩ to Startup Script macro.

This command can only be executed after LKSS OFF.

ERSS

Erase Script

Erase Startup Script macro.

This command can only be executed after LKSS OFF.

LKSS(?) {*z* }

Lock Script

Set (query) Startup Script write/erase lockout {to *z*=(OFF 0, **ON 1**)}.

After LKSS ON has been executed, the Startup Script is protected against any modification or erasure. To reprogram the Startup Script, issue LKSS OFF.

After reset, the default is LKSS ON.

LISS?

List Script

List Startup Script contents (#4aaaaxxxxx... format).

NBSS?

Bytes Used in Script

Query number of bytes used in Startup Script macro.

AVSS?

Space Available in Script

Query number of spaces available in Startup Script memory.

ENSS?

Script Enable

Query Startup Script Enable token value (ON 1, OFF 0) from rear-panel DIP switch.

RNSS

Run Script

Run Startup Script now (independent of DIP switch setting).

2.5.8 Housekeeping commands

The Housekeeping commands provide status information on the SIM900 Mainframe hardware, such as the 10 MHz timebase and the total power consumption.

TBIN?

Timebase Input Detect

Query external Timebase Input detected token (`TRUE 1`, `FALSE 0`).

VTBI?

Timebase Input Analog

Query external Timebase Input detection circuit voltage, in millivolts.

PLLC(?) {*z* }

Timebase Control

Set (query) Timebase PLL Control {to *z*=(`OFF 0`, `ON 1`, **`AUTO 2`**)}.

When PLLC `AUTO` is set, the SIM900 Mainframe will automatically activate the timebase phase-locked loop (PLL) whenever TBIN? `TRUE`. To force the PLL to remain either off (free-running clock) or on (always attempting to lock), set PLLC appropriately.

LOCK?

Timebase Status

Query Timebase status token value (`FREE 0`, `LOCKING 1`, `LOCKED 2`, `FAULT 3`).

While the PLL is off, LOCK? returns `FREE`. For 50 ms after activating the PLL, LOCK? will return `LOCKING`; after that, either `LOCKED` or `FAULT` is returned, indicating whether the loop is successfully tracking the external timebase input.

VLOC?

Timebase Status Analog

Query lock detector voltage, in millivolts.

VVCO?

Timebase VCO

Query the VCO control voltage, in millivolts.

VMON?

Primary Voltage

Query the SIM900 Mainframe primary power supply voltage, in millivolts (nominally 24000).

IMON?

Primary Current

Query the SIM900 Mainframe primary power supply current, in milliamps.

PMON?

Primary Power

Query the SIM900 Mainframe primary power supply power, in milliwatts. Equal to (VMON?)×(IMON?)/1000.

UNDV?

Undervoltage

Query output undervoltage detect.

While an output undervoltage condition exists, UNDV? returns `1`; otherwise it returns `0`. An undervoltage occurs if any of the SIM900 Mainframe output voltages droop below their nominal value, indicating an overload.

TICK?

Elapsed Time

Query time elapsed since power-on (each count is 50 ms).

DIPS? [*i*]

DIP Switch

Query rear panel dip switch [bit *i*] value.

2.5.9 Interface commands

The interface commands include required IEEE–488.2 common commands, along with additional commands for configuring the interface between the SIM900 Mainframe and the host computer. Additionally, commands for flushing the port input buffers and output queues are also provided.

*RST

Reset

Reset the mainframe to default configuration.

The following register and command values are established immediately following a *RST:

cmd	value
BRER	0
RDDR	0
RPER	0
TERM[†]	0
TMOT[†]	1000
MSGL	64
CEOI	0 OFF
EOIX	0 OFF
PRTC	0 EAVS
PRTD	0 COMM
PLLC	2 AUTO
CONS	0 OFF
VERB	5(MFERRORS and IOMON)
TOKN	0 OFF
LKSS	1 ON
REQT	0 OFF
REQF	0 OFF

[†] TERM and TMOT are reset for all ports

FLOQ

Flush Output Queue

Flush the host output queue.

SRST [*p*]

SIM Reset

Sends the SIM Reset signal [to Port *p*] (default is all SIM ports).

SRST causes a ⟨break⟩ signal (MARK level) to be asserted for 100 milliseconds, either to Port *p* or to all SIM ports. Upon receiving the ⟨break⟩ signal, any connected SIM should flush its internal input buffer, reset its command parser, and default to 9600 baud communications.

FLSI [*p*]

Flush Port Input Buffers

Flushes port input buffers [associated with Port *p*].

FLSO [*p*]

Flush Port Output Queues

Flushes port output queues [associated with Port *p*].

FLSH [*p*]

Flush Port Buffers

Flushes port input buffers & output queues [associated with Port *p*].

---

*IDN?

Identify

Read the mainframe device identification string.

The identification string is formatted as:
`Stanford␣Research␣Systems,SIM900,s/n******,ver#.#`
where `******` is the 6-digit serial number, and `#.#` is the firmware revision level.

---

*TST?

Self Test

Perform mainframe self-test (currently no-op, returns 0).

---

*CLS

Clear Status

*CLS immediately clears the SSEV, ESR, CESR, FCSR, PDPR, TOSR, IOSR, and CTSR registers.

---

*STB? [*i*]

Status Byte

Reads the serial poll Status Byte register [bit *i*].

---

*SRE(?) [*i*,] {*j* }

Service Request Enable

Set (query) the Service Request Enable register [bit *i*] {to *j*}.

---

*ESR? [*i*]

Standard Event Status

Reads the Standard Event Status Register [bit *i*].

Upon executing *ESR?, the returned bit(s) of the ESR register are cleared.

---

*ESE(?) [*i*,] {*j* }

Standard Event Status Enable

Set (query) the Standard Event Status Enable Register [bit *i*] {to *j*}.

---

*PSC(?) {*i* }

Power-on Status Clear

Set (query) the Power-on Status Clear flag {to *j*}. The Power-on Status Clear flag is stored in non-volatile memory in the SIM900 Mainframe, and thus maintains its value through power-cycle events.

If the value of the flag is `0`, then the Service Request Enable and Standard Event Status Enable Registers (*SRE, *ESE) are stored in non-volatile memory, and retain their values through power-cycle events. If the value of the flag is non-zero, then these two registers are cleared upon power-cycle.

The initial factory default is *PSC 1, causing *SRE 0 and *ESE 0 to be set at power-on.

---

*OPC(?)

Operation Complete

Operation Complete. Sets the OPC flag in the ESR register.

The query form *OPC? writes a 1 in the output queue when complete, but does not affect the ESR register.

---

*WAI

Wait to Continue

Wait to Continue. Equivalent to a no-op.

CONS(?) {*z*}

Console Mode

Set (query) the host port Console mode {to *z*=(**OFF 0**, ON 1)}.

CONS only has an effect when host interface is RS-232, and causes each character received at the host input buffer to be copied to the host output queue.

After reset, the default is CONS OFF.

WAIT *i*

Wait

Wait *i* milliseconds before processing more commands from the host.

This command can be especially useful programming the Startup Script.

REQT(?) {*z*}

Announce REQT

Set (query) the REQT announce mode {to *z*=(**OFF 0**, ON 1)}.

When REQT ON is set, any event which causes a new service request condition will cause the characters <reqt>+⟨term⟩ to be spontaneously written to the host output buffer.

On reset, the default is REQT OFF.

REQF(?) {*z*}

Announce REQF

Set (query) the REQF announce mode {to *z*=(**OFF 0**, ON 1)}.

When REQF ON is set, any event which causes the Master Summary Status message to become false will cause the characters <reqf>+⟨term⟩ to be spontaneously written to the host output buffer.

After reset, the default is REQF OFF.

LEXE?

Execution Error

Query the last Execution Error code. Valid codes are:

Value	Definition
0	No execution error since power-on
1	Invalid port
2	Invalid token
3	Command failed
4	Timeout
5	Invalid bit
6	Invalid value
7	Checksum failed
8	Invalid host interface

LCME?

Command Error

Query the last Command Error code. Valid codes are:

Value	Definition
0	No parser error since power-on
1	Illegal first character
2	Illegal name
3	Undefined command
4	Extra questio mark
5	No query allowed
6	Only query allowed
7	Missing parameter(s)
8	No parameters allowed
9	Premature command terminator
10	Message buffer overflow
11	Illegal half-byte in hex parameter
12	Command buffer overflow
13	Illegal extra string parameter
14	Illegal extra hex parameter
15	Illegal extra binary parameter
16	Illegal byte-digits count
17	Illegal bytes count
18	Null parameter
19	Extra parameter(s)
20	Illegal port
21	Illegal short integer
22	Illegal long integer
23	Illegal token integer
24	Unknown token
25	Illegal string parameter
26	Illegal hex parameter
27	Illegal binary parameter
28	⟨EOI⟩ without ⟨LF⟩ on indef. arb. block

TOKN(?) {*z*}

Token Mode

Set (query) the Token Query mode {to *z*=(**OFF 0**, ON 1)}.

If TOKN ON is set, then queries to the SIM900 mainframe that return tokens will return the text keyword; otherwise they will return the decimal integer value.

Thus, the only possible responses to the TOKN? query are ON and 0.

After reset, the default is TOKN OFF.

2.6 Register Model

Registers in the SIM900 Mainframe are divided into 2 broad categories: control registers and status registers. Control registers govern the operation of the mainframe (specifically, controlling the automatic routing of port data), while status registers (and their associated configuration registers) govern the monitoring and reporting of status conditions within the mainframe.

The registers are represented as either 8-bit or 16-bit unsigned integer values. Individual flags within a register correspond to an integer weight of 2^n. Most commands for setting or reading the registers have an optional parameter to select a single bit within the register. Using this optional parameter, all flag values are either `0` or `1`. In the default (whole-register) form, the commands treat the register as a single integer value by summing the weighted values of the individual flags.

2.6.1 Control registers

2.6.1.1 Receive Data Disable (RDDR)

This is a 16-bit wide register that controls transfers of data from the ports to the port input buffers.

Weight	Bit	Flag
1	0	undef (0)
2	1	P1-disable
4	2	P2-disable
8	3	P3-disable
16	4	P4-disable
32	5	P5-disable
64	6	P6-disable
128	7	P7-disable
256	8	P8-disable
512	9	P9-disable
1024	10	PA-disable
2048	11	PB-disable
4096	12	PC-disable
8192	13	PD-disable
16384	14	undef (0)
32768	15	undef (0)

If the bit corresponding to a particular port is set in the RDDR, and data from that port arrives at the mainframe, it is immediately discarded. If the corresponding bit is cleared, then data flows as normal from the port hardware into the port input buffer.

If a bit is set in RDDR while data is already in the corresponding port input buffer, that data remains available to the host computer, but no further data from the port will be accepted until the RDDR bit is cleared. *No flow control signals are asserted to stop the port from transmitting data.*

At power-on, this register is cleared.

2.6.1.2 Receive Pass-Through Enable (RPER)

This is a 16-bit wide register that controls transfers of data from the ports to the mainframe output queue.

Weight	Bit	Flag
1	0	undef (0)
2	1	P1-passthrough
4	2	P2-passthrough
8	3	P3-passthrough
16	4	P4-passthrough
32	5	P5-passthrough
64	6	P6-passthrough
128	7	P7-passthrough
256	8	P8-passthrough
512	9	P9-passthrough
1024	10	PA-passthrough
2048	11	PB-passthrough
4096	12	PC-passthrough
8192	13	PD-passthrough
16384	14	undef (0)
32768	15	undef (0)

If the bit corresponding to a particular port is set in the RPER when data from that port arrives at the mainframe, it is immediately encapsulated into `MSG` packet(s) by the mainframe and transferred to the output queue without further host intervention (i. e., no query command is needed).

Messages are formatted as: `MSG` *p*,*b*

where *p* is the port the message came from, and *b* is a definite-length arbitrary binary data block (see section 2.5.1.4 for format).

Since the mainframe imposes no protocol requirements on the data transferred to or from the SIMs and external RS-232 devices, the `MSG` packets are divided at arbitrary points in the data stream. To correctly reconstruct the byte stream from a particular port, the host must concatenate the contents of all packets from that port in the order received.

The data block *b* of the `MSG` packets are guaranteed not to exceed the byte limit set by the `MSGL` command. If fewer bytes are received from a port, then a `MSG` packet is generated after a timeout of approximately 5 serial-byte times (5 or 6 ms at 9600 baud).

When the mainframe is switched to connect mode to a port (via CONN), the RPER is cleared to all zeros. This prevents data from any of the unconnected ports from being interspersed in the output queue with data bytes from the connected port. Upon leaving connect mode the RPER remains cleared, so the host must reprogram RPER if needed.

At power-on, this register is cleared.

2.6.1.3 Broadcast Enable (BER)

This is a 16-bit wide register that selects ports to receive broadcast messages (BRDC, BRDT) from the mainframe.

Weight	Bit	Flag
1	0	undef (0)
2	1	P1–broadcast
4	2	P2–broadcast
8	3	P3–broadcast
16	4	P4–broadcast
32	5	P5–broadcast
64	6	P6–broadcast
128	7	P7–broadcast
256	8	P8–broadcast
512	9	P9–broadcast
1024	10	PA–broadcast
2048	11	PB–broadcast
4096	12	PC–broadcast
8192	13	PD–broadcast
16384	14	undef (0)
32768	15	undef (0)

If the bit corresponding to a particular port is set in the BER, then any broadcast messages sent from the host to the mainframe will be transferred to that port. Note that the Port C and Port D bits are only effective if PRTC `PORT` (or PRTD `PORT`) have been set.

This register is cleared on power-on.

2.6.2 Status registers

The SIM900 Mainframe status registers follow the hierarchical IEEE–488.2 format. A block diagram of the entire status register array is given in Figure 2.1.

There are four broad categories of registers in the status model of the mainframe:

Condition Registers : These read-only registers correspond to the real-time condition of some underlying physical property being monitored. Queries return the latest value of the property, and have no further side effects. Condition register names end with CR.

Transition Selection Registers : These read/write registers define specific transition events (such as 0 → 1 or 1 → 0). The event is then defined by the selected transition in the value of the underlying condition register. Transition register names end with PT or NT.

Event Registers : These read-only registers record the occurrence of defined events within the mainframe. If the event occurs, the corresponding bit is set to 1. Upon querying an event register, any set bits within it are cleared[2]. These are sometimes known as "sticky bits," since once set, a bit can only be cleared by reading its value. Event register names typically end with SR.

Enable Registers : These read/write registers define a bitwise mask for their corresponding event register. If any bit position is set in an event register while the same bit position is also set in the enable register, then the corresponding summary bit message is set. Enable register names typically end with SE.

[2] Except for any summary bit messages.

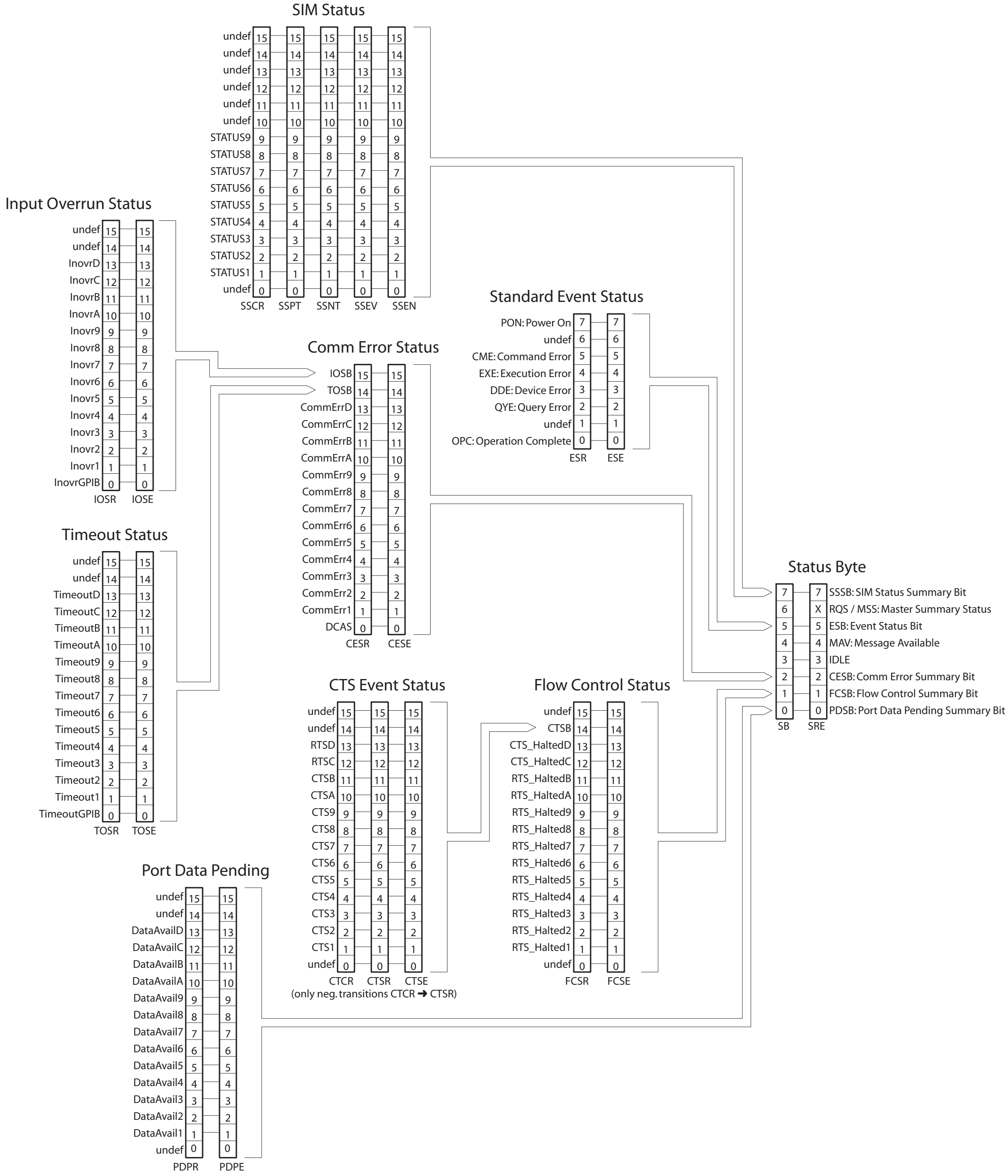

Figure 2.1: Status Register Model for the SIM900 Mainframe.

2.6.2.1 Status Byte (SB)

This is an 8-bit wide register defined by IEEE–488.2. It can be read either by a GPIB serial poll, or through the *STB? command. The Status Byte is the top-level summary of the SIM900 Mainframe status model.

Weight	Bit	Flag
1	0	PDSB
2	1	FCSB
4	2	CESB
8	3	IDLE
16	4	MAV
32	5	ESB
64	6	RQS/MSS
128	7	SSSB

Bit 6 returns the RQS message during a GPIB serial poll, and MSS when queried with *STB?

PDSB : Port Data Pending Summary Bit. Indicates whether one or more of the enabled flags in the Port Data Pending Status Register has become true.

FCSB : Port Flow Control Summary Bit. Indicates whether one or more of the enabled flags in the Port Flow Control Status Register has become true.

CESB : Comm Error Summary Bit. Indicates whether one or more of the enabled flags in the Comm Error Status Register has become true.

IDLE : Indicates that the input buffer is empty and the command parser is idle. Can be used to help synchronize mainframe query responses.

MAV : Message Available. Indicates whether or not the output queue has any data pending for the host.

ESB : Event Status Bit. Indicates whether one or more of the enabled events in the Standard Event Status Register is true.

RQS : IEEE–488 Request Service message, indicating this device (the SIM900 Mainframe) requested service.

MSS : Master Summary Status. Indicates whether one or more of the enabled status messages in the Status Byte register is true.

SSSB : SIM Status Summary Bit. Indicates whether one or more of the enabled event flags in the SIM Status Event Register has become true.

Bits in the Status Byte are *not* cleared by the *STB? query. These bits are only cleared by reading the underlying event registers, or by clearing the corresponding enable registers.

2.6.2.2 Service Request Enable (SRE)

This is an 8-bit wide register defined by IEEE–488.2. Each bit in the SRE corresponds one-to-one with a bit in the SB register, and acts as a bitwise AND of the SB flags to generate MSS/RQS. Bit 6 of the SRE is undefined - setting it has no effect, and reading it always returns 0. This register is set and queried with the *SRE(?) command.

At power-on, this register is cleared if *PSC is non-zero. It retains its value if *PSC 0.

2.6.2.3 Standard Event Status (ESR)

This is an 8-bit wide register defined by IEEE–488.2. These event flags are all "sticky bits" that are set by the corresponding event, and cleared only by reading or with the *CLS command. Reading a single bit (with the *ESR? *i* query) clears only bit *i*.

Weight	Bit	Flag
1	0	OPC
2	1	undef (0)
4	2	QYE
8	3	DDE
16	4	EXE
32	5	CME
64	6	undef (0)
128	7	PON

OPC : Operation Complete. Set by the *OPC command.

QYE : Query Error. Indicates either (1) an attempt to read data when no output is present or pending (only valid for GPIB), or (2) data in the output queue has been lost.

DDE : Device Dependent Error. Indicates a mainframe power supply undervoltage.

EXE : Execution Error. Indicates an error in a command that was successfully parsed. Out-of-range parameters are an example. The error code can be queried with LEXE?.

CME : Command Error. Indicates a parser-detected error. The error code can be queried with LCME?.

PON : Power On. Indicates that an off-to-on transition has occurred.

2.6.2.4 Standard Event Status Enable (ESE)

This is an 8-bit wide register defined by IEEE–488.2. It acts as a bitwise AND with the ESR register to produce the single bit ESB message in the Status Byte Register (SB). It can be set and queried with the *ESE(?) command.

At power-on, this register is cleared if *PSC is non-zero. It retains its value if *PSC 0.

2.6.2.5 SIM Status Condition (SSCR)

This 16-bit wide register monitors the −STATUS line from the 9 SIM ports (1–9). There is no corresponding signal for the RS-232 ports. SSCR is a read-only register.

Weight	Bit	Flag
1	0	undef (0)
2	1	Status1
4	2	Status2
8	3	Status3
16	4	Status4
32	5	Status5
64	6	Status6
128	7	Status7
256	8	Status8
512	9	Status9
1024	10	undef (0)
2048	11	undef (0)
4096	12	undef (0)
8192	13	undef (0)
16384	14	undef (0)
32768	15	undef (0)

The −STATUS output signal from a SIM idles high, indicating no special status information to report, and set or pulsed to low to indicate some (device-dependent) status message. The SSCR records the complement of the signal, so −STATUS True (low) appears as a 1 in the SSCR, while −STATUS False (high) appears as a 0. Reads to the SSCR (via SSCR?) return the present value of the −STATUS signal for port *n* as bit Status*n*.

2.6.2.6 SIM Status Positive/Negative Transition (SSPT/SSNT)

These two 16-bit wide registers control the mapping of transitions in the SSCR to setting flags in the SSEV register. For any particular SIM port, if the corresponding bit is set in SSPT, then a 0 → 1 transition in the SSCR causes the bit to be set in the SSEV. Likewise, if a bit is set in SSNT, then a 1 → 0 transition in the SSCR causes the bit to be set in the SSEV.

All combinations of SSPT and SSNT settings for the 9 SIM ports are valid. At power-on, both SSPT and SSNT are cleared.

2.6.2.7 SIM Status Event (SSEV)

This 16-bit wide register monitors selected events in the SSCR, based on transitions selected in SSPT and SSNT. When the selected transition(s) occur, the corresponding bit is set. Reading the register clears it (reading a single bit clears only that bit). This register is cleared by the *CLS command.

2.6.2.8 SIM Status Enable (SSEN)

This is a 16-bit wide register that masks the SSEV register. The logical OR of the bitwise AND of SSEV and SSEN produces the SSSB message in the Status Byte register (SB).

2.6.2.9 Communications Error Status (CESR)

This is a 16-bit wide register that monitors communications errors on the ports.

Weight	Bit	Flag
1	0	DCAS
2	1	CommErr1
4	2	CommErr2
8	3	CommErr3
16	4	CommErr4
32	5	CommErr5
64	6	CommErr6
128	7	CommErr7
256	8	CommErr8
512	9	CommErr9
1024	10	CommErrA
2048	11	CommErrB
4096	12	CommErrC
8192	13	CommErrD
16384	14	TOSB
32768	15	IOSB

DCAS : Device Clear Active State. Set by the mainframe receiving a Device Clear event (either from DCL, SDC, or RS-232 ⟨break⟩.

CommErr*p* : Communication Error for Port *p*. Set by the mainframe detecting a serial error (such as parity violation or framing error, or an input buffer overflow) on the corresponding port input hardware.

TOSB : Timeout Summary Bit message. TOSB indicates the logical OR of the bitwise AND of TOSR and TOSE (see below).

IOSB : Input Overflow Summary Bit message. IOSB indicates the logical OR of the bitwise AND of IOSR and IOSE (see below).

This register (with the exception of the TOSB & IOSB bits) is cleared either by reading, or with the *CLS command.

2.6.2.10 Communications Error Status Enable (CESE)

This is a 16-bit wide register that masks the CESR register. The logical OR of the bitwise AND of CESR and CESE produces the CESB message in the Status Byte register (SB).

At power-on, this register is cleared.

2.6.2.11 Timeout Status (TOSR)

This is a 16-bit wide register that monitors timeout errors.

Weight	Bit	Flag
1	0	TimeoutGPIB
2	1	Timeout1
4	2	Timeout2
8	3	Timeout3
16	4	Timeout4
32	5	Timeout5
64	6	Timeout6
128	7	Timeout7
256	8	Timeout8
512	9	Timeout9
1024	10	TimeoutA
2048	11	TimeoutB
4096	12	TimeoutC
8192	13	TimeoutD
16384	14	undef (0)
32768	15	undef (0)

If an attempt to write to a port output queue fails due to a timeout error, the corresponding bit in the TOSR is set. (This can also be thought of as an output overflow error.) The register is cleared either by reading, or with the *CLS command.

2.6.2.12 Timeout Status Enable (TOSE)

This is a 16-bit wide register that masks the TOSR register. The logical OR of the bitwise AND of TOSR and TOSE produces the TOSB message in the Communications Error Status Register (CESR).

2.6.2.13 Input Overflow Status (IOSR)

This is a 16-bit wide register that monitors input overflow errors.

Weight	Bit	Flag
1	0	InoverGPIB
2	1	Inovr1
4	2	Inovr2
8	3	Inovr3
16	4	Inovr4
32	5	Inovr5
64	6	Inovr6
128	7	Inovr7
256	8	Inovr8
512	9	Inovr9
1024	10	InovrA
2048	11	InovrB
4096	12	InovrC
8192	13	InovrD
16384	14	undef (0)
32768	15	undef (0)

When data is received by the mainframe at a port, it is initially stored in the 512-byte port input buffer, on a first-in, first-out (FIFO) basis. If an input buffer overflows, an error condition is recorded in the IOSR, and the corresponding input buffer is flushed. If the host input buffer overflows, then the host output queue is also flushed (as though a Device Clear had occurred).

The register is cleared either by reading, or with the *CLS command.

2.6.2.14 Input Overflow Status Enable (IOSE)

This is a 16-bit wide register that masks the IOSR register. The logical OR of the bitwise AND of IOSR and IOSE produces the IOSB message in the Communications Error Status Register (CESR).

2.6.2.15 Flow Control Status (FCSR)

This 16-bit wide register monitors the flow-control hardware of the ports. If the mainframe stops the incoming flow of data from a port (by deasserting RTS), then the corresponding bit in the FCSR is set. Whether or not this means that data was actually lost depends on the implementation of the particular SIM or RS-232 device sourcing data to the mainframe.

Since Ports C and D (EAVS and COMM) are DCE ports rather than DTE ports, the hardware flow-control outputs are actually CTS for these two ports.

Note if FLOW is set to XON or NONE, the FCSR will not be set by flow control events. Only RTS flow control is monitored by FCSR.

Weight	Bit	Flag
1	0	undef (0)
2	1	RTS–Halted1
4	2	RTS–Halted2
8	3	RTS–Halted3
16	4	RTS–Halted4
32	5	RTS–Halted5
64	6	RTS–Halted6
128	7	RTS–Halted7
256	8	RTS–Halted8
512	9	RTS–Halted9
1024	10	RTS–HaltedA
2048	11	RTS–HaltedB
4096	12	CTS–HaltedC
8192	13	CTS–HaltedD
16384	14	CTSB
32768	15	undef (0)

CTSB is the CTS Summary Bit status message, and is the logical OR of the bitwise AND of CTSR and CTSE (see below).

This register (with the exception of the CTSB bit) is cleared either by reading. All bits are cleared by *CLS.

2.6.2.16 Flow Control Status Enable (FCSE)

This is a 16-bit wide register that masks the FCSR register. The logical OR of the bitwise AND of FCSR and FCSE produces the FCSB message in the Status Byte register (SB).

At power-on, this register is cleared.

2.6.2.17 CTS Status Condition (CTCR)

This 16-bit wide register monitors the CTS line of all ports (1–D).

Weight	Bit	Flag
1	0	undef (0)
2	1	CTS1
4	2	CTS2
8	3	CTS3
16	4	CTS4
32	5	CTS5
64	6	CTS6
128	7	CTS7
256	8	CTS8
512	9	CTS9
1024	10	CTSA
2048	11	CTSB
4096	12	RTSC
8192	13	RTSD
16384	14	undef (0)
32768	15	undef (0)

The CTS output signal from a SIM or RS-232 device is typically used for hardware flow control, and is asserted high (1) to indicate the mainframe is Clear To Send new bytes to the device, and deasserted (0) to stop the flow.

Since Ports C and D (EAVS and COMM) are DCE ports instead of DTE ports, the hardware flow-control inputs for these two ports are actually RTS rather than CTS.

SIM ports (1–9) each have a pull-down resistor wired to the CTS input, so unconnected slots will show CTCR[p] = 0. The RS-232 ports each have a pull-up resistor wired to the flow-control input, so unconnected RS-232 ports will show CTCR[p]=1.

Regardless of the current FLOW setting for a port, the CTCR always reflects the real-time value of CTS (RTS for Ports C & D).

2.6.2.18 CTS Status (CTSR)

This is another 16-bit wide register that monitors the flow control hardware of the ports. A 1 → 0 transition in the CTCR will cause the corresponding bit in the CTSR to be set. Thus, only Negative Transitions generate the CTS Status events. When FLOW is RTS, this indicates the SIM or RS-232 device has halted the flow of data from the mainframe.

This register is cleared either by reading, or with the *CLS command.

2.6.2.19 CTS Status Enable (CTSE)

This is a 16-bit wide register that masks the CTSR register. The logical OR of the bitwise AND of CTSR and CTSE produces the CTSB message in the Flow Control Status register (FCSR).

At power-on, this register is cleared.

2.6.2.20 Port Data Pending (PDPR)

This is a 16-bit wide register that monitors incoming data from the ports to the mainframe.

Weight	Bit	Flag
1	0	undef (0)
2	1	DataAvail1
4	2	DataAvail2
8	3	DataAvail3
16	4	DataAvail4
32	5	DataAvail5
64	6	DataAvail6
128	7	DataAvail7
256	8	DataAvail8
512	9	DataAvail9
1024	10	DataAvailA
2048	11	DataAvailB
4096	12	DataAvailC
8192	13	DataAvailD
16384	14	undef (0)
32768	15	undef (0)

A bit for a given port in the PDPR is set if any bytes arrive from that port while it is NOT mapped to the mainframe output queue (either by the CONN command or through the RPER register). Reading the register clears it (reading a single bit clears only that bit).

This register is cleared by the *CLS command.

2.6.2.21 Port Data Pending Enable (PDPE)

This is a 16-bit wide register that masks the PDPR register. The logical OR of the bitwise AND of PDPR and PDPE produces the PDSB message in the Status Byte register (SB).

At power-on, this register is cleared.