

Operation and Service Manual

RTD Temperature Monitor

SIM923A



Stanford Research Systems

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.

Copyright © Stanford Research Systems, Inc., 2004. All rights reserved.

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.



SIM923A RTD Temperature Monitor

Contents

General Information	iii
Service	iii
Symbols	iv
Notation	v
Specifications	vi
1 Getting Started	1-1
1.1 Introduction to the Instrument	1-2
1.2 Front-Panel Operation	1-2
1.3 Sensor Interface	1-4
1.4 SIM Interface	1-6
2 Remote Operation	2-1
2.1 Index of Common Commands	2-2
2.2 Alphabetic List of Commands	2-4
2.3 Introduction	2-6
2.4 Commands	2-7
2.5 Status Model	2-19

General Information

The SIM923A RTD Temperature Monitor, part of Stanford Research Systems' Small Instrumentation Modules family, consists of a single-channel sensor excitation and readout for precision low-noise resistive thermometry. Front-panel analog outputs provide both scaled (linear voltage proportional to temperature) and raw (buffered sensor voltage) signals. Two reversible DC current sources ($10\ \mu\text{A}$ & $1\ \text{mA}$) provide sensor excitation to the four-wire measurement circuit.

Service

Do not install substitute parts or perform any unauthorized modifications to this instrument.

The SIM923A is a single-wide module designed to be used inside the SIM900 Mainframe. Do not turn on the power until the module is completely inserted into the mainframe and locked in place.

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 buttons are set as [Button];
[Adjust ▲▼] is shorthand for “[Adjust ▲] & [Adjust ▼]”.
- 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

		Min	Typ	Max	Units	
Inputs	Number of inputs	1				
	Sensor type	Platinum RTD or other resistive sensor (either PTC or NTC)				
	Measurement type	4-wire				
	Excitation	constant current				
	(10 μ A setting)	9.999	10 ± 15	10.001	μ A DC ppm/ $^{\circ}$ C	
	(1 mA setting)	0.9999	1.0 ± 15	1.0001	mA DC ppm/ $^{\circ}$ C	
Sensor Characteristics	Sensor units	Ohms				
	sensor+lead resistance			1400	Ω (1 mA exc.)	
				140	k Ω (10 μ A exc.)	
	Calibration curves	built-in: IEC 751/DIN 43760				
1 user defined curve						
	Curve size			1024	points	
	Temperature range	1.4		873	K (typical)	
		sensor dependent				
Measurement	Measurement Rate	5 readings per second				
	Display resolution		4		digits	
	Interface resolution		7		digits	
	Measurement resolution		1.2			m Ω rms (1 mA exc.)
				120		m Ω rms (10 μ A exc.)
	Accuracy, (23 \pm 1) $^{\circ}$ C	5 m Ω + 0.01 %				(1 mA exc.)
		0.5 Ω + 0.01 %				(10 μ A exc.)
	Temperature coefficient	-5		+5	ppm/ $^{\circ}$ C	
Operating	Temperature	0		40	$^{\circ}$ C	
	Power	± 15 , +5			V DC	
	Supply current	50 (± 15 V), 250 (+5 V)			mA	

General Characteristics

Interface	Serial (RS-232) through SIM interface
Connectors	DB-9 (female)
	4-wire measurement + ground
	DB-15 (male) SIM interface
Weight	1.4 lbs
Dimensions	1.5" W \times 3.6" H \times 7.0" D

1 Getting Started

This chapter gives you the necessary information to get started quickly with the SIM923A RTD Temperature Monitor.

In This Chapter

1.1	Introduction to the Instrument	1-2
1.1.1	Overview	1-2
1.2	Front-Panel Operation	1-2
1.2.1	Excitation	1-2
1.2.2	Units	1-2
1.2.3	Reverse	1-3
1.2.4	Setpoint	1-4
1.2.5	Scaled Output	1-4
1.3	Sensor Interface	1-4
1.3.1	Four-wire measurement	1-4
1.3.2	Two-wire measurement	1-5
1.4	SIM Interface	1-6
1.4.1	SIM interface connector	1-6
1.4.2	Direct interfacing	1-6

1.1 Introduction to the Instrument

The SIM923A RTD Temperature Monitor provides excitation and readout for a single resistive temperature detector (RTD). Analog outputs, both proportional to temperature and the buffered sensor voltage, are available on the front panel.

1.1.1 Overview

The SIM923A uses a ratiometric design, where the excitation current is passed through both an internal, stable reference resistor and the user's resistive thermometer. The instrument measures the ratio of the voltage across the user's resistor to the voltage across the reference resistor, multiplied by the (known) value of the reference resistor, to obtain the user result.

Two separate excitation and reference circuits are included in the SIM923A, a 1 mA excitation appropriate for ~ 1 k Ω full-scale measurements, and a 10 μ A excitation useable for up to ~ 100 k Ω full-scale measurements. The excitation is also reversible, permitting the user to test for potential offset voltages in the setup.

A precision 24-bit analog-to-digital converter records the results. New sensor measurements are available at a rate of 5 conversions per second.

1.2 Front-Panel Operation

The front panel of the SIM923A (see Figure 1.1) provides a simple operator interface.

1.2.1 Excitation

The current source of the SIM923A RTD Temperature Monitor can be fully controlled from the front panel.

To select between 10 μ A and 1 mA briefly tap [Excitation]. To toggle the current on and off, *hold* [Excitation] for about 1 second, until the display switches between the numerical result and the word *OFF*.

The excitation can also be controlled with the EXCI and EXON remote interface commands.

1.2.2 Units

The SIM923A displays results either as raw sensor units (in ohms) or temperature (in kelvin). Pressing [Units] toggles between these two modes; the active units are indicated by the illuminated *K*, or Ω (or



Figure 1.1: The SIM923A front and rear panels.

k Ω). The display mode can also be controlled with the DISP remote interface command.

When temperature units are selected, a sensor calibration curve is required. The SIM923A is programmed with a standard curve for 100 Ω , 0.00385 alpha Pt RTD sensors (IEC 751). The SIM923A also has non-volatile memory to store a separate sensor curve with up to 1024 temperature-vs-resistance points. Which curve to use is set by the remote interface CURV command.

1.2.3 Reverse

The excitation current produced by the SIM923A can be reversed to check for possible offset potentials. Pressing [Reverse] toggles the current direction; when *REVERSE* is lit, the current is reversed (i.e., positive current flows from the *I-* terminal to the *I+* terminal). Note that when reversed, the raw sensor monitor output will also reverse, but the scaled analog output (and display result) will not. The current direction can also be controlled with the IPOL remote interface command.

1.2.4 Setpoint

The scaled analog output can produce a voltage proportional to either absolute temperature, or temperature deviation around some setpoint. [Setpoint] is used to review or set the setpoint temperature. Briefly tapping [Setpoint] toggles the display between showing the latest reading, and showing the setpoint temperature (indicated by the *SETPOINT* lamp).

Holding [Setpoint] for longer than ~1 second will update the setpoint temperature, as indicated by the word *SET* flashing on the display. There are two different behaviors that can occur, depending on the state of the numeric display.

- If the display is presently showing the setpoint (the *SETPOINT* lamp is lit), then long presses of [Setpoint] will toggle between commanding the setpoint temperature to 273.15 K and the most recent measurement result.
- If the display is presently showing the measurement result (the *SETPOINT* lamp is *not* lit), then long presses of [Setpoint] simply update the setpoint to the most recent measurement result.

To configure the SIM923A for an arbitrary setpoint, see the TSET remote interface command.

1.2.5 Scaled Output

[Scale] controls the gain for the scaled analog output. Gain is indicated by the temperature span corresponding to a full-scale output voltage (10 V). Three scales are available from the front panel, 10 K (1 V/K), 100 K (0.1 V/K), and 1000 K (0.01 V/K). To configure the SIM923A for an arbitrary analog scale, see the VKEL remote interface command.

[Rel] toggles the scaled output between two modes: absolute ($V \propto T$), and relative ($V \propto T - T_{\text{set}}$). Relative mode can also be controlled with the AMOD remote interface command.

1.3 Sensor Interface

The sensor interface on the SIM923A consists of a rear-panel DB-9/F connector (see Figure 1.1). The connector pinout is given in Table 1.1

1.3.1 Four-wire measurement

To avoid sensitivity to wiring lead resistance, the SIM923A is configured for four-wire measurements. The basic circuit for this wiring

Pin	Signal
1	I+
2	I-
3	ground
4	V+
5	V-
6-9	ground

Table 1.1: SIM923A Sensor Interface Connector Pin Assignments, DB-9

scheme is shown in Figure 1.2.

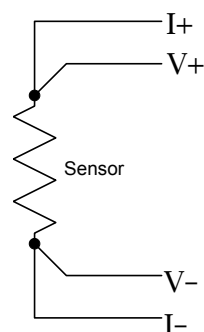


Figure 1.2: Wiring diagram for four-wire readout.

1.3.2 Two-wire measurement

If application-specific constraints limit the number of leads to the sensor, the SIM923A can be wired to measure the sensor resistance with a simple two-wire circuit, shown in Figure 1.3. Note that the lead resistance (past the junction points of the current and voltage leads) will add as a direct resistance error when measuring the sensor.

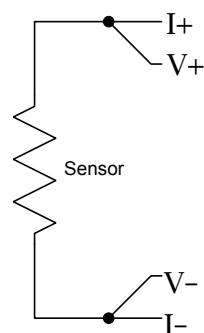


Figure 1.3: Wiring diagram for two-wire readout.

1.4 SIM Interface

The primary connection to the SIM923A RTD Temperature Monitor is the rear-panel DB-15 SIM interface connector. Typically, the SIM923A is mated to a SIM900 Mainframe via this connection, either through one of the internal Mainframe slots, or the remote cable interface.

It is also possible to operate the SIM923A directly, without using the SIM900 Mainframe. This section provides details on the interface.

1.4.1 SIM interface connector

The DB-15 SIM interface connector carries all the power and communications lines to the instrument. The connector signals are specified in Table 1.2

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 (optional connection)
6	-5V	MF ⇒ SIM	Power supply (No connection in SIM923A)
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 (optional connection)
13	+5V	MF ⇒ SIM	Power supply (digital circuitry)
14	+15V	MF ⇒ SIM	Power supply (analog circuitry)
15	+24V	MF ⇒ SIM	Power supply (No connection in SIM923A)

Table 1.2: SIM Interface Connector Pin Assignments, DB-15

1.4.2 Direct interfacing

The SIM923A is intended for operation in the SIM900 Mainframe, but users may wish to directly interface the module to their own systems without the use of additional hardware.

The mating connector needed is a standard DB-15 receptacle, such as Amp part # 747909-2 (or equivalent). Clean, well-regulated supply voltages of +5, ±15 VDC must be provided, following the pin-out specified in Table 1.2. Ground must be provided on pins 1 and 8,

with chassis ground on pin 9. The -STATUS signal may be monitored on pin 2 for a low-going TTL-compatible output indicating a status message.

The SIM923A has no internal protection against reverse polarity, missing supply, or overvoltage on the power supply pins.

1.4.2.1 Direct interface cabling

If the user intends to directly wire the SIM923A independent of the SIM900 Mainframe, communication is usually possible by directly connecting the appropriate interface lines from the SIM923A DB-15 plug to the RS-232 serial port of a personal computer.¹ Connect RXD from the SIM923A directly to RD on the PC, TXD directly to TD, and similarly RTS→RTS and CTS→CTS. In other words, a null-modem style cable is *not* needed.

To interface directly to the DB-9 male (DTE) RS-232 port typically found on contemporary personal computers, a cable must be made with a female DB-15 socket to mate with the SIM923A, and a female DB-9 socket to mate with the PC's serial port. Separate leads from the DB-15 need to go to the power supply, making what is sometimes know as a "hydra" cable. The pin-connections are given in Table 1.3.

DB-15/F to SIM923A	Name
	DB-9/F
3 ↔ 7	RTS
4 ↔ 8	CTS
10 ↔ 3	TxD
11 ↔ 2	RxD
5	Computer Ground
	to P/S
7 ↔	-15 VDC
14 ↔	+15 VDC
13 ↔	+5 VDC
8,9 ↔	Ground (P/S return current)
1 ↔	Signal Ground (separate wire to Ground)

Table 1.3: SIM923A Direct Interface Cable Pin Assignments

¹ Although the serial interface lines on the DB-15 do not satisfy the minimum voltage levels of the RS-232 standard, they are typically compatible with desktop personal computers

1.4.2.2 Serial settings

The initial serial port settings at power-on are: 9600 Baud, 8–bits, no parity, 1 stop bit, and RTS/CTS flow control. These may be changed with the **BAUD**, **FLOW**, or **PARI** commands.

The maximum *standard* baud rate that the SIM923A supports is 38400. The minimum baud rate is 110. Above 38400, the SIM923A can be set to the following (non-RS-232-standard) baud rates: 62500, 78125, 104167, 156250. Note that these rates are typically not accessible on a standard PC RS-232 port, but can be used between the SIM923A and the SIM900 Mainframe.

2 Remote Operation

This chapter describes operating the SIM923A over the serial interface.

In This Chapter

2.1	Index of Common Commands	2-2
2.2	Alphabetic List of Commands	2-4
2.3	Introduction	2-6
2.3.1	Power-on configuration	2-6
2.3.2	Buffers	2-6
2.3.3	Device Clear	2-6
2.4	Commands	2-7
2.4.1	Command syntax	2-7
2.4.2	Notation	2-8
2.4.3	Readout commands	2-9
2.4.4	Setpoint and analog output commands	2-10
2.4.5	Excitation commands	2-11
2.4.6	Display & configuration commands	2-11
2.4.7	Sensor calibration commands	2-12
2.4.8	Serial communication commands	2-13
2.4.9	Status commands	2-14
2.4.10	Interface commands	2-15
2.5	Status Model	2-19
2.5.1	Status Byte (SB)	2-20
2.5.2	Service Request Enable (SRE)	2-21
2.5.3	Standard Event Status (ESR)	2-21
2.5.4	Standard Event Status Enable (ESE)	2-21
2.5.5	Communication Error Status (CESR)	2-22
2.5.6	Communication Error Status Enable (CESE)	2-22
2.5.7	Overload Status (OVCR)	2-23
2.5.8	Overload Status (OVSR)	2-23
2.5.9	Overload Status Enable (OVSE)	2-23

2.1 Index of Common Commands

symbol	definition
<i>i,j</i>	Integers
<i>f,g</i>	Floating-point values
<i>z</i>	Literal token
<i>s</i>	Arbitrary character sequence (no “,” or “;”)
(?)	Required for queries; illegal for set commands
<i>var</i>	Parameter always required
{ <i>var</i> }	Required parameter for set commands; illegal for queries
[<i>var</i>]	Optional parameter for both set and query forms

Readout

RVAL? [<i>n</i>]	2 – 9	Resistance Value
TVAL? [<i>n</i>]	2 – 9	Temperature Value
TDEV? [<i>n</i>]	2 – 9	Temperature Deviation Value
SOUT	2 – 9	Stop Streaming

Setpoint/Analog Output

TSET(?) { <i>f</i> }	2 – 10	Temperature Setpoint
VKEL(?) { <i>f</i> }	2 – 10	Temperature Scale Factor
AMOD(?) { <i>z</i> }	2 – 10	Analog Output Mode
AOUT(?) { <i>f</i> }	2 – 10	Analog Output Voltage

Excitation

EXON(?) { <i>z</i> }	2 – 11	Excitation On/Off
EXCI(?) { <i>z</i> }	2 – 11	Current Source High/Low
IPOL(?) { <i>z</i> }	2 – 11	Current Source Polarity

Display & Configuration

DISX(?) { <i>z</i> }	2 – 11	Display Enable/Disable
DISP(?) { <i>z</i> }	2 – 11	Display Mode
FPLC(?) { <i>i</i> }	2 – 11	Frequency of Power Line Cycle

Sensor Calibration

CINI(?) { <i>z,s</i> }	2 – 12	Initialize Sensor Calibration
CAPT <i>f,g</i>	2 – 12	Add User Curve Point
CAPT? <i>j</i>	2 – 13	Query User Curve Point
CURV(?) { <i>z</i> }	2 – 13	Select Sensor Curve

Serial Communications

BAUD(?) { <i>i</i> }	2 – 13	Baud Rate
FLOW(?) { <i>z</i> }	2 – 13	Flow Control

PARI(?) {z}	2-13 Parity
-------------	-------------

Status

*CLS	2-14 Clear Status
*STB? [i]	2-14 Status Byte
*SRE(?) [i,] {j}	2-14 Service Request Enable
*ESR? [i]	2-14 Standard Event Status
*ESE(?) [i,] {j}	2-14 Standard Event Status Enable
CESR? [i]	2-14 Communication Error Status
CESE(?) [i,]{j}	2-14 Communication Error Status Enable
OVCR? [i]	2-14 Overload Condition
OVSr? [i]	2-15 Overload Status
OVSE(?) [i,]{j}	2-15 Overload Status Enable
PSTA(?) {z}	2-15 Pulse -STATUS Mode

Interface

*RST	2-15 Reset
CONS(?) {z}	2-15 Console Mode
*IDN?	2-16 Identify
*OPC(?)	2-16 Operation Complete
LEXE?	2-16 Execution Error
LCME?	2-17 Device Error
LBTN?	2-17 Button
TOKN(?) {z}	2-17 Token Mode
TERM(?) {z}	2-18 Response Termination

2.2 Alphabetic List of Commands

★	
*CLS	2 – 14 Clear Status
*ESE(?) [<i>i</i>] [<i>j</i>]	2 – 14 Standard Event Status Enable
*ESR? [<i>i</i>]	2 – 14 Standard Event Status
*IDN?	2 – 16 Identify
*OPC(?)	2 – 16 Operation Complete
*RST	2 – 15 Reset
*SRE(?) [<i>i</i>] [<i>j</i>]	2 – 14 Service Request Enable
*STB? [<i>i</i>]	2 – 14 Status Byte

A	
AMOD(?) { <i>z</i> }	2 – 10 Analog Output Mode
AOUT(?) { <i>f</i> }	2 – 10 Analog Output Voltage

B	
BAUD(?) { <i>i</i> }	2 – 13 Baud Rate

C	
CAPT <i>f,g</i>	2 – 12 Add User Curve Point
CAPT? <i>j</i>	2 – 13 Query User Curve Point
CESE(?) [<i>i</i>] [<i>j</i>]	2 – 14 Communication Error Status Enable
CESR? [<i>i</i>]	2 – 14 Communication Error Status
CINI(?) { <i>z,s</i> }	2 – 12 Initialize Sensor Calibration
CONS(?) { <i>z</i> }	2 – 15 Console Mode
CURV(?) { <i>z</i> }	2 – 13 Select Sensor Curve

D	
DISP(?) { <i>z</i> }	2 – 11 Display Mode
DISX(?) { <i>z</i> }	2 – 11 Display Enable/Disable

E	
EXCI(?) { <i>z</i> }	2 – 11 Current Source High/Low
EXON(?) { <i>z</i> }	2 – 11 Excitation On/Off

F	
FLOW(?) { <i>z</i> }	2 – 13 Flow Control
FPLC(?) { <i>i</i> }	2 – 11 Frequency of Power Line Cycle

I	
IPOL(?) { <i>z</i> }	2 – 11 Current Source Polarity

L

LBTN?	2-17	Button
LCME?	2-17	Device Error
LEXE?	2-16	Execution Error

O

OVCR? [i]	2-14	Overload Condition
OVSE(?) [i,]{j}	2-15	Overload Status Enable
OVSr? [i]	2-15	Overload Status

P

PARI(?) {z}	2-13	Parity
PSTA(?) {z}	2-15	Pulse -STATUS Mode

R

RVAL? [n]	2-9	Resistance Value
-----------	-----	------------------

S

SOUT	2-9	Stop Streaming
------	-----	----------------

T

TDEV? [n]	2-9	Temperature Deviation Value
TERM(?) {z}	2-18	Response Termination
TOKN(?) {z}	2-17	Token Mode
TSET(?) {f}	2-10	Temperature Setpoint
TVAL? [n]	2-9	Temperature Value

V

VKEL(?) {f}	2-10	Temperature Scale Factor
-------------	------	--------------------------

2.3 Introduction

Remote operation of the SIM923A is through a simple command language documented in this chapter. Both set and query forms of most commands are supported, allowing the user complete control of the amplifier from a remote computer, either through the SIM900 Mainframe or directly via RS-232 (see Section 1.4.2.1).

See Table 1.2 for specification of the DB-15 SIM interface connector.

2.3.1 Power-on configuration

The settings for the remote interface are 9600 baud with no parity and hardware flow control, and local echo disabled (CONS OFF).

Most of the SIM923A instrument settings are stored in non-volatile memory, and at power-on the instrument returns to the state it was last in when power was removed. Exceptions are noted in the command descriptions.

Reset values of parameters are shown in **boldface**.

2.3.2 Buffers

Incoming data from the host interface is stored in a 32-byte input buffer. Characters accumulate in the input buffer until a command terminator (either <CR> or <LF>) is received, at which point the message is parsed and executed. Query responses from the SIM923A are buffered in a 32-byte output queue.

If the input buffer overflows, then all data in *both* the input buffer and the output queue are discarded, and an error is recorded in the CESR and ESR status registers.

2.3.3 Device Clear

The SIM923A host interface can be asynchronously reset to its power-on configuration by sending an RS-232-style <break> signal. From the SIM900 Mainframe, this is accomplished with the SRST command; if directly interfacing via RS-232, then use a serial break signal. After receiving the Device Clear, the interface is reset to 9600 baud and CONS mode is turned OFF. Note that this *only* resets the communication interface; the basic function of the SIM923A is left unchanged; to reset the instrument, see *RST.

The Device Clear signal will also terminate any streaming outputs from the SIM923A due to a TVAL?, TDEV?, or RVAL? query of multiple conversions.

2.4 Commands

This section provides syntax and operational descriptions for remote commands.

2.4.1 Command syntax

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.

Multiple parameters are separated by commas. Multiple commands may be sent on one command line by separating them with semicolons (;) so long as the input buffer does not overflow. Commands are terminated by either <CR> or <LF> characters. Null commands and whitespace are ignored. Execution of command(s) does not begin until the command terminator is received.

tokens Token parameters (generically shown as *z* in the command descriptions) can be specified either as a keyword or integer value. Command descriptions list the valid keyword options, with each keyword followed by its corresponding integer value. 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.

2.4.2 Notation

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

symbol	definition
<i>i,j</i>	Integers
<i>f,g</i>	Floating-point values
<i>z</i>	Literal token
<i>s</i>	Arbitrary character sequence (no “,” or “;”)
(?)	Required for queries; illegal for set commands
<i>var</i>	Parameter always required
{ <i>var</i> }	Required parameter for set commands; illegal for queries
[<i>var</i>]	Optional parameter for both set and query forms

2.4.3 Readout commands

RVAL? [<i>n</i>]	<p>Resistance Value</p> <p>Query the sensor resistance.</p> <p>The result is formatted as <code>+#.#####E+##</code>, where <code>+</code> indicates sign ("<code>+</code>" or "<code>-</code>"), and the value following <code>E</code> is a power-of-ten that multiplies the preceding value. The value is always in ohms.</p> <p>If the optional parameter <i>n</i> is provided, then <i>n</i> sequential conversion results are returned to the host. If <i>n</i>=0, the conversion results continue indefinitely. To terminate the stream before <i>n</i> results (or when <i>n</i>=0), issue the <code>SOUT</code> command.</p> <p>Note that omitting <i>n</i> is equivalent to <i>n</i>=1.</p>
TVAL? [<i>n</i>]	<p>Temperature Value</p> <p>Query the sensor temperature value.</p> <p>The result is formatted identically to <code>RVAL</code>, above.</p> <p>If the optional parameter <i>n</i> is provided, then <i>n</i> sequential conversion results are returned to the host. If <i>n</i>=0, the conversion results continue indefinitely. To terminate the stream before <i>n</i> results (or when <i>n</i>=0), issue the <code>SOUT</code> command.</p> <p>Note that omitting <i>n</i> is equivalent to <i>n</i>=1.</p>
TDEV? [<i>n</i>]	<p>Temperature Deviation Value</p> <p>Query the sensor temperature minus setpoint value.</p> <p>The result is formatted identically to <code>RVAL</code>, above.</p> <p>If the optional parameter <i>n</i> is provided, then <i>n</i> sequential conversion results are returned to the host. If <i>n</i>=0, the conversion results continue indefinitely. To terminate the stream before <i>n</i> results (or when <i>n</i>=0), issue the <code>SOUT</code> command.</p> <p>Note that omitting <i>n</i> is equivalent to <i>n</i>=1.</p>
SOUT	<p>Stop Streaming</p> <p>Turn off streaming output.</p>

2.4.4 Setpoint and analog output commands

TSET(?) {f}	<p>Temperature Setpoint</p> <p>Set (query) the temperature setpoint {to <i>f</i> kelvin}. This is the “offset” value used for the scaled analog output when in AMOD REL mode.</p>
VKEL(?) {f}	<p>Temperature Scale Factor</p> <p>Set (query) the temperature scale factor {to <i>f</i>}, in volts per kelvin. When set to 1.0, 0.1, or 0.01, the corresponding front panel scale indicator (10 V = 10 K, 100 K, or 1000 K) is lit; when set to any other value, all three indicators are dark.</p>
AMOD(?) {z}	<p>Analog Output Mode</p> <p>Set (query) the analog output mode {to <i>z</i>=(ABS 0, REL 1, MAN 2)}.</p> <p>In AMOD ABS, the scaled output is given by</p> $V = T \times VKEL.$ <p>In AMOD REL, the output is given by</p> $V = (T - T_{\text{set}}) \times VKEL.$ <p>In AMOD MAN, the output is controlled by AOUT.</p>
AOUT(?) {f}	<p>Analog Output Voltage</p> <p>Set (query) the manual analog output {to <i>f</i>}, in volts.</p>

2.4.5 Excitation commands

EXON(?) {z}	Excitation On/Off Set (query) the excitation current {to z=(OFF 0, ON 1)}.
EXCI(?) {z}	Current Source High/Low Set (query) the excitation current {to z=(LOW 0, HIGH 1)}. Setting EXCI LOW commands the current source to 10 μ A, while EXCI HIGH sets it to 1 mA. Note that the EXCI setting does <i>not</i> alter the EXON setting; if the current source is off, switching between 10 μ A and 1 mA still leaves the source off.
IPOL(?) {z}	Current Source Polarity Set (query) the excitation current polarity {to z=(POSITIVE 0, NEGATIVE 1)}. Setting IPOL NEGATIVE reverses the excitation; IPOL POSITIVE sets it to nominal.

2.4.6 Display & configuration commands

DISX(?) {z}	Display Enable/Disable Set (query) the front panel display {to z=(OFF 0, ON 1)}. The DISX setting is <i>not</i> stored in non-volatile memory. At power-on, the SIM923A returns to DISX ON.
DISP(?) {z}	Display Mode Set (query) the display mode {to z=(OHMS 0, TEMP 1, or TSET 2)}. If DISP OHMS, then results are displayed in ohms or milliohms; if DISP TEMP, results are displayed in kelvin. If DISP TSET, the display shows the setpoint temperature.
FPLC(?) {j}	Frequency of Power Line Cycle Set (query) the power line rejection frequency {to j=(50, 60)}, in Hz.

2.4.7 Sensor calibration commands

In addition to the built-in curve, the SIM923A has a dedicated 1024-point non-volatile memory for storing user calibration data. Once loaded, this curve is retained by the SIM923A through power cycles.

CINI(?) {z,s}

Initialize Sensor Calibration

Initialize sensor calibration curve.

The set form of the command, CINI z,s, erases the old curve. The parameter z specifies the curve format, as one of:

	z	meaning
LINEAR	0	ohms, kelvin
SEMILOGT	1	ohms, log ₁₀ (kelvin)
SEMILOGR	2	log ₁₀ (ohms), kelvin
LOGLOG	3	log ₁₀ (ohms), log ₁₀ (kelvin)

The second parameter s is an arbitrary identification string for this sensor calibration curve. This string can consist of any non-blank characters *except* the comma “,” or semicolon “;”, and can be up to 15 characters in length.

If CURV USER was active when CINI is executed, the SIM923A reverts to CURV STAN, and records an execution error (EXE bit in the ESR) of “uninitialized curve.”

The query form of the command, CINI?, returns the following response:

⟨format⟩, ⟨serial⟩, n

where ⟨format⟩ is the calibration curve format (same as z above), ⟨serial⟩ is the identification string (s above), and n is the number of points currently stored in the curve.

CAPT f,g

Add User Curve Point

Add a new point to the user curve. f is the raw sensor value (in either ohms or log₁₀(ohms), depending on curve format), and g is the corresponding temperature value (in either kelvin or log₁₀(kelvin), again depending on curve format).

Note that curve points *must* be added in increasing order of sensor value f.

Temperature values must be in the range $1 \text{ mK} \leq T \leq 9999.499 \text{ K}$.

CAPT? <i>j</i>	<p>Query User Curve Point</p> <p>Query the value of the user curve, entry point <i>j</i>.</p> <p>The response is ⟨sensor⟩, ⟨temperature⟩, where ⟨sensor⟩ is the raw sensor value (in either ohms or log₁₀(ohms), depending on curve format), and ⟨temperature⟩ is the corresponding temperature value (in either kelvin or log₁₀(kelvin), again depending on curve format).</p>
CURV(?) { <i>z</i> }	<p>Select Sensor Curve</p> <p>Set (query) the sensor curve selection {to <i>z</i>=(STAN 0, USER 1)}.</p> <p>The built-in standard curve for Pt RTDs (IEC 751) is selected by <i>z</i>=STAN. When CURV USER is selected, the user calibration curve (previously loaded with CINI and CAPT) is used.</p>

2.4.8 Serial communication commands

For further discussion of serial communications, including limitations on settings, see Section 1.4.2.2.

BAUD(?) { <i>i</i> }	<p>Baud Rate</p> <p>Set (query) the baud rate {to <i>i</i>}.</p> <p>At power-on, the baud rate defaults to 9600.</p> <p>Actual baud rate settings depend on implementation details of the SIM923A, and are limited to values of 312500/<i>n</i>, where <i>n</i> is a positive integer. As a result, queries of BAUD? will in general be slightly different from the set values. For example, after setting BAUD 9600, the query BAUD? will respond 9470. The functional requirement for successful asynchronous serial communication is no greater than ~ 5 % mismatch in baud rates.</p>
FLOW(?) { <i>z</i> }	<p>Flow Control</p> <p>Set (query) flow control {to <i>z</i>=(NONE 0, RTS 1, XON 2)}.</p> <p>At power-on, the SIM923A defaults to FLOW RTS flow control.</p>
PARI(?) { <i>z</i> }	<p>Parity</p> <p>Set (query) parity {to <i>z</i> = (NONE 0, ODD 1, EVEN 2, MARK 3, SPACE 4)}.</p> <p>At power-on, the SIM923A defaults to PARI NONE.</p>

2.4.9 Status commands

The Status commands query and configure registers associated with status reporting of the SIM923A. See Section 2.5 for more details.

*CLS	Clear Status *CLS immediately clears the ESR, CESR, and the OVSR.
*STB? [<i>i</i>]	Status Byte Reads the Status Byte register [bit <i>i</i>]. Execution of the *STB? query (without the optional bit <i>i</i>) always causes the –STATUS signal to be deasserted. Note that *STB? <i>i</i> will <i>not</i> clear –STATUS, even if bit <i>i</i> is the only bit presently causing the –STATUS signal.
*SRE(?) [<i>i</i> ,] { <i>j</i> }	Service Request Enable Set (query) the Service Request Enable register [bit <i>i</i>] {to <i>j</i> }.
*ESR? [<i>i</i>]	Standard Event Status Reads the Standard Event Status Register [bit <i>i</i>]. Upon executing *ESR?, the returned bit(s) of the ESR register are cleared.
*ESE(?) [<i>i</i> ,] { <i>j</i> }	Standard Event Status Enable Set (query) the Standard Event Status Enable Register [bit <i>i</i>] {to <i>j</i> }.
CESR? [<i>i</i>]	Communication Error Status Query Communication Error Status Register [for bit <i>i</i>]. Upon executing a CESR? query, the returned bit(s) of the CESR register are cleared.
CESE(?) [<i>i</i> ,] { <i>j</i> }	Communication Error Status Enable Set (query) Communication Error Status Enable Register [bit <i>i</i>] {to <i>j</i> }.
OVCR? [<i>i</i>]	Overload Condition Query Overload Condition Register [for bit <i>i</i>].

OVSR? [i]	<p>Overload Status</p> <p>Query Overload Status Register [for bit <i>i</i>].</p> <p>Upon executing a OVSR? query, the returned bit(s) of the OVSR register are cleared.</p>
OVSE(?) [i,]{j}	<p>Overload Status Enable</p> <p>Set (query) Overload Status Enable Register [bit <i>i</i>] {to <i>j</i>}.</p>
PSTA(?) {z}	<p>Pulse –STATUS Mode</p> <p>Set (query) the Pulse –STATUS Mode {to <i>z</i>=(OFF 0, ON 1)}.</p> <p>When PSTA ON is set, any new service request will only <i>pulse</i> the –STATUS signal low (for a minimum of 1 μs). The default behavior is to latch –STATUS low until a *STB? query is received.</p> <p>At power-on, PSTA is set to OFF.</p>

2.4.10 Interface commands

*RST	<p>Reset</p> <p>Reset the SIM923A to default configuration.</p> <p>The following commands are internally executed upon *RST:</p> <ul style="list-style-type: none"> • DISX ON • EXON ON • EXCI LOW • CURV STAN • DISP TEMP • AMOD ABS • VKEL 1
CONS(?) {z}	<p>Console Mode</p> <p>Set (query) the Console mode {to <i>z</i>=(OFF 0, ON 1)}.</p> <p>CONS causes each character received at the Input Buffer to be copied to the Output Queue.</p> <p>At power-on and Device-Clear, CONS is set to OFF.</p>

***IDN?** Identify

Read the device identification string.

The identification string is formatted as:
 Stanford_Research_Systems,SIM923A,s/n*****,ver#.##
 where ***** is the 6-digit serial number, and #.## is the firmware revision level.

***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.

LEXE? Execution Error

Query the last execution error code. Valid codes are:

Value	Definition
0	No execution error since last LEXE?
1	Illegal value
2	Wrong token
3	Invalid bit
16	Uninitialized curve
17	Curve full
18	Curve point out-of-order
19	Illegal temperature value
20	No Excitation

LCME?

Device Error

Query the last command error code. Valid codes are:

Value	Definition
0	No command error since last LCME?
1	Illegal command
2	Undefined command
3	Illegal query
4	Illegal set
5	Missing parameter(s)
6	Extra parameter(s)
7	Null parameter(s)
8	Parameter buffer overflow
9	Bad floating-point
10	Bad integer
11	Bad integer token
12	Bad token value
13	Bad hex block
14	Unknown token

LBTN?

Button

Query the last button-press code. Valid codes are:

Value	Definition
0	no button pressed since last query
1	[Rel]
2	[Scale]
3	[Setpoint]
4	[Units]
5	[Reverse]
6	[Excitation]

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 SIM923A that return tokens will return the text keyword; otherwise they return the decimal integer value.

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

At power-on, TOKN OFF is set.

TERM(?) {z}**Response Termination**

Set (query) the <term> sequence {to z=(NONE 0, CR 1, LF 2, **CRLF 3**, LFCR 4)}.

The <term> sequence is appended to all query responses sent by the module, and is constructed of ASCII character(s) 13 (carriage return) and/or 10 (line feed).

At power-on, TERM CRLF is set.

2.5 Status Model

The SIM923A status registers follow the hierarchical IEEE-488.2 format. A block diagram of the status register array is given in Figure 2.1.

There are three categories of registers in the SIM923A status model:

- 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 other effect. Condition register names end with CR.
- Event Registers : These read-only registers record the occurrence of defined events. When the event occurs, the corresponding bit is set to 1. Upon querying an event register, any set bits within it are cleared. These are sometimes known as “sticky bits,” since once set, a bit can only be cleared by reading its value. Event register names 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 end with SE.

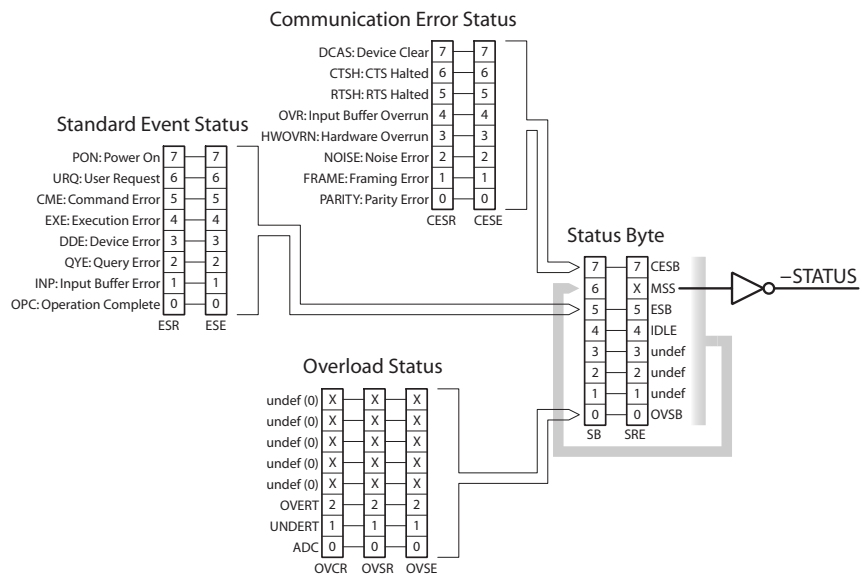


Figure 2.1: Status Register Model for the SIM923A RTD Temperature Monitor.

2.5.1 Status Byte (SB)

The Status Byte is the top-level summary of the SIM923A status model. When masked by the Service Request Enable register, a bit set in the Status Byte causes the -STATUS signal to be asserted on the rear-panel SIM interface connector.

Typically, -STATUS remains asserted (low) until a *STB? query is received, at which time -STATUS is deasserted (raised)¹. After clearing the -STATUS signal, it will only be re-asserted in response to a *new* status-generating condition.

Weight	Bit	Flag
1	0	OVSB
2	1	undef (0)
4	2	undef (0)
8	3	undef (0)
16	4	IDLE
32	5	ESB
64	6	MSS
128	7	CESB

OVSB : Overload Status Summary Bit. Indicates whether one or more of the enabled flags in the Overload 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 SIM923A query responses.

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

MSS : Master Summary Status. Indicates whether one or more of the enabled status messages in the Status Byte register is true. Note that while -STATUS is released by the *STB? query, MSS is only cleared when the underlying enabled bit message(s) are cleared.

CESB : Communication Error Summary Bit. Indicates whether one or more of the enabled flags in the Communication Error Status 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.

¹ but see the PSTA command

2.5.2 Service Request Enable (SRE)

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 the MSS bit in the SB and the -STATUS signal. 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.

This register is cleared at power-on.

2.5.3 Standard Event Status (ESR)

The Standard Event Status register consists of 8 event flags. 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	INP
4	2	QYE
8	3	DDE
16	4	EXE
32	5	CME
64	6	URQ
128	7	PON

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

INP : Input Buffer Error. Indicates data has been discarded from the Input Buffer.

QYE : Query Error. Indicates data in the Output Queue has been lost.

DDE : Device Dependent Error. Not used in the SIM923A.

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?`.

URQ : User Request. Indicates a front-panel button was pressed.

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

2.5.4 Standard Event Status Enable (ESE)

The ESE 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.

This register is cleared at power-on.

2.5.5 Communication Error Status (CESR)

The Communication Error Status register consists of 8 event flags; each of which is set by the corresponding event, and cleared only by reading or with the *CLS command. Reading a single bit (with the CESR? *i* query) clears only bit *i*.

Weight	Bit	Flag
1	0	PARITY
2	1	FRAME
4	2	NOISE
8	3	HWOVRN
16	4	OVR
32	5	RTSH
64	6	CTSH
128	7	DCAS

PARITY : Parity Error. Set by serial parity mismatch on incoming data byte.

FRAME : Framing Error. Set when an incoming serial data byte is missing the STOP bit.

NOISE : Noise Error. Set when an incoming serial data byte does not present a steady logic level during each asynchronous bit-period window.

HWOVRN : Hardware Overrun. Set when an incoming serial data byte is lost due to internal processor latency. Causes the Input Buffer to be flushed, and resets the command parser.

OVR : Input Buffer Overrun. Set when the Input Buffer is overrun by incoming data. Causes the Input Buffer to be flushed, and resets the command parser.

RTSH : RTS holdoff. The host asserted the RTS hardware flow control line to stop the SIM923A from sending more data.

CTSH : CTS holdoff. The SIM923A asserted the CTS hardware flow control line to stop the host from sending more data.

DCAS : Device Clear. Indicates the SIM923A received the Device Clear signal (an RS-232 <break>). Clears the Input Buffer and Output Queue, and resets the command parser.

2.5.6 Communication Error Status Enable (CESE)

The CESE acts as a bitwise AND with the CESR register to produce the single bit CESB message in the Status Byte Register (SB). It can

be set and queried with the CESE(?) command.

This register is cleared at power-on.

2.5.7 Overload Status (OVCR)

The Overload Condition Register consists of 7 single-bit monitors of condition events within the SIM923A. Bits in the OVCR reflect the real-time values of their corresponding signals. Reading the entire register, or individual bits within it, does not affect the OVCR.

Weight	Bit	Flag
1	0	ADC
2	1	UNDERT
4	2	OVERT
8	3	undef (0)
16	4	undef (0)
32	5	undef (0)
64	6	undef (0)
128	7	undef (0)

ADC : Analog-to-Digital overload. The digitizer input was overloaded.

UNDERT : Calibration curve underflow ($R < R_{\min}$).

OVERT : Calibration curve overflow ($R > R_{\max}$).

2.5.8 Overload Status (OVSR)

The Overload Status Register consists of (latching) event flags that correspond one-to-one with the bits of the OVCR (see above). Upon the transition $0 \rightarrow 1$ of any bit within the OVCR, the corresponding bit in the OVSR becomes set.

Bits in the OVSR are unaffected by the $1 \rightarrow 0$ transitions in the OVCR, and are cleared only by reading or with the *CLS command. Reading a single bit (with the OVSR? *i* query) clears only bit *i*.

2.5.9 Overload Status Enable (OVSE)

The OVSE acts as a bitwise AND with the OVSR register to produce the single bit OVSB message in the Status Byte Register (SB). It can be set and queried with the OVSE(?) command.

This register is cleared at power-on.