Operation and Service Manual

Diode Temperature Monitor

SIM922

Stanford Research Systems

Revision 1.5 • September 17, 2010

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.

Stanford Research Systems, Inc.
1290–D 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. Document number 9-01556-903

Contents

General Information

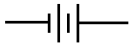
The SIM922 Diode Temperature Monitor, part of Stanford Research Systems' Small Instrumentation Modules family, consists of four channels of sensor excitation and readout for precision low-noise diode thermometry. Independent 10 μA DC current sources provide sensor excitations to the four-wire measurement circuits.

Service

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

The SIM922 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		4		
	Sensor type		Silicon diode		
	Measurement type		4–wire		
	Excitation		constant current		
		9.999	10	10.001	μA DC, ±5 ppm/°C
Sensor Characteristics	Sensor units		Volts		
	Input Range	0		2.5	V
	sensor+lead resistance			1400	Ω
	Calibration curves		1 built-in		
			4 user defined curves		
	Curve size (each)			256	points
	Temperature range	1.4		475	K (typical)
			sensor dependent		
Measurement	Display resolution		4		digits
	Interface resolution		1		μV
			1		mK
	Measurement resolution		1.2		μV rms
	Measurement accuracy, (23 ± 1)°C		20 μV + 0.01 %		
	Temperature coefficient	−5		+5	ppm/°C
Operating	Temperature	0		40	°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	2 DB–9 (female)
	4–wire measurement + ground (Ch 1 & 2)
	4–wire measurement + ground (Ch 3 & 4)
	DB–15 (male) SIM interface
Weight	1.4 lbs
Dimensions	1.5″ W × 3.6″ H × 7.0″ D

1 Getting Started

This chapter gives you the necessary information to get started quickly with the SIM922 Diode Temperature Monitor.

In This Chapter

1.1 Introduction to the Instrument

The SIM922 Diode Temperature Monitor monitors up to four (4) silicon diode thermometers using a precision 10 μA DC current excitation.

1.1.1 Overview

Each channel has an independent, precision 10 μA current source, so the excitation to the user's sensor remains steady as the SIM922 cycles between measurements. Disabling a channel switches that channel's current source off.

A precision 24–bit analog-to-digital converter is cycled between the channels with a rate of 4 conversions per second. With all channels enabled, a complete cycle is completed every second; disabling some channels yields a corresponding increase in overall cycle rate.

1.2 Front-Panel Operation

The front panel of the SIM922(see Figure 1.1) provides a simple operator interface.

Figure 1.1: The SIM922 front and rear panels.

1.2.1 Excitation

The four channels of the SIM922 Diode Temperature Monitor can be independently enabled or disabled from the front panel. Disabling a channel turns off the 10 μA excitation current to that sensor, and speeds the readout rate for the remaining (enabled) channels.

To toggle the excitation for a channel, first briefly press [Excitation]. One of the four channel displays will appear highlighted, indicating which channel is selected. Briefly tap [Excitation] to advance the highlighted selection until the desired channel is selected. Then depress *and hold* [Excitation] until the display switches between the numerical value and the word OFF.

1.2.2 Units

The SIM922 displays the sensor results either as voltage (in millivolts) or temperature (in kelvin). Pressing [Units] toggles between these two modes; the active units are indicated by the illuminated *K* or *mV*.

When temperature units are selected, a sensor calibration curve is required. The SIM922 is preprogrammed with a standard curve for Si diode sensors. Each channel also has non-volatile memory to store a separate sensor curve with up to 256 temperature-vs-resistance points. The actual curve to use is selected with the remote interface CURV command.

1.3 Sensor Interface

The sensor interface on the SIM922 consists of two rear-panel DB–9/F connectors, labeled "CHANNELS 1 & 2" and "CHANNELS 3 & 4" (see Figure 1.1). Both connectors follow the same pinout, given in Table 1.1

Pin	Signal
1	I+, Ch 1(3)
2	I−, Ch 1(3)
3	ground
4	I+, Ch 2(4)
5	I−, Ch 2(4)
6	V+, Ch 1(3)
7	V−, Ch 1(3)
8	V+, Ch 2(4)
9	V−, Ch 2(4)

Table 1.1: SIM922 Sensor Interface Connector Pin Assignments, DB–9

1.3.1 Four-wire measurement

To avoid sensitivity to wiring lead resistance, the SIM922 is configured for four-wire measurements. The basic circuit for this wiring scheme is shown in Figure 1.2.

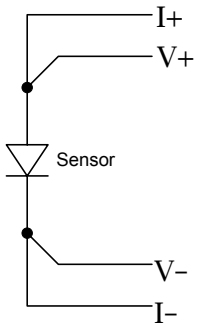


Figure 1.2: Wiring diagram for four-wire readout.

1.3.2 Two-wire measurement

If application-specific constraints limit the number of leads to the sensor, the SIM922 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.

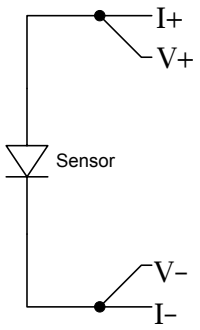


Figure 1.3: Wiring diagram for two-wire readout.

1.4 SIM Interface

The primary connection to the SIM922 Diode Temperature Monitor is the rear-panel DB–15 SIM interface connector. Typically, the SIM922 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 SIM922 directly, without using the SIM900 Mainframe. This section provides details on the interface.

⚠ CAUTION

The SIM922 has no internal protection against reverse polarity, missing supply, or overvoltage on the power supply pins. Misapplication of power may cause circuit damage. SRS recommends using the SIM922 together with the SIM900 Mainframe for most applications.

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 SIM922)
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 SIM922)

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

1.4.2 Direct interfacing

The SIM922 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.

1.4.2.1 Direct interface cabling

If the user intends to directly wire the SIM922 independent of the SIM900 Mainframe, communication is usually possible by directly connecting the appropriate interface lines from the SIM922 DB–15 plug to the RS-232 serial port of a personal computer.[1] Connect RXD from the SIM922 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 SIM922, 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 SIM922		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: SIM922 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 SIM922 supports is 38400. The minimum baud rate is 110. Above 38400, the SIM922 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 SIM922 and the SIM900 Mainframe.

2 Remote Operation

This chapter describes operating the SIM922 over the serial interface.

In This Chapter

2.1 Index of Common Commands

symbol	definition
i,j	Integers
f,g	Floating-point values
c	Channel number (0–4); *c*=0 means "all"
z	Literal token
s	Arbitrary character sequence (no "," or ";")
(*?*)	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

2.2 Alphabetic List of Commands

2.3 Introduction

Remote operation of the SIM922 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 0FF).

Most of the SIM922 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 SIM922 are buffered in a 64-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 SIM922 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 0FF. Note that this *only* resets the communication interface; the basic function of the SIM922 is left unchanged; to reset the instrument, see *RST.

The Device Clear signal will also terminate any streaming outputs from the SIM922 due to a TVAL? 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,j	Integers
f,g	Floating-point values
c	Channel number (0–4); *c*=0 means “all”
z	Literal token
s	Arbitrary character sequence (no “,” or “;”)
(*?*)	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

2.4.3 Readout commands

VOLT? *c* [,*n*]	Voltage Value Query the sensor voltage for channel *c*. If *c*=0, then all four channels are returned in the format: <c1>,<c2>,<c3>,<c4>. If the optional parameter *n* is provided, then *n* sequential conversion results are returned to the host. If *n*=0, the conversion results continue indefinitely. To terminate the stream before *n* results (or when *n*=0), issue the SOUT command. Note that omitting *n* is equivalent to *n*=1.
TVAL? *c* [,*n*]	Temperature Value Query the sensor temperature value for channel *c*. If *c*=0, then all four channels are returned in the format: <c1>,<c2>,<c3>,<c4>. If the optional parameter *n* is provided, then *n* sequential conversion results are returned to the host. If *n*=0, the conversion results continue indefinitely. To terminate the stream before *n* results (or when *n*=0), issue the SOUT command. Note that omitting *n* is equivalent to *n*=1.
SOUT	Stop Streaming Turn off streaming output.

2.4.4 Excitation commands

EXON(?) *c* {,*z*}	Excitation On/Off Set (query) the Channel *c* excitation current {to *z*=(OFF 0, ON 1)}. If *c*=0, then all four channel excitation states are set (queried).

2.4.5 Display & configuration commands

DISX(?) {*z*}

Display Enable/Disable

Set (query) the front panel display {to *z*=(`OFF 0`, **`ON 1`**)}.

The DISX setting is *not* stored in non-volatile memory. At power-on, the SIM922 returns to DISX `ON`.

DTEM(?) {*z*}

Display Temperature

Set (query) the display temperature mode {to *z*=(`OFF 0`, **`ON 1`**)}.

If DTEM OFF, then results are displayed in ohms; otherwise, results are displayed in kelvin.

FPLC(?) {*i*}

Frequency of Power Line Cycle

Set (query) the power line rejection frequency {to *j*=(`50`, `60`)}, in Hz.

2.4.6 Sensor calibration commands

In addition to the built-in curve, each channel has a dedicated 256-point non-volatile memory for storing user calibration curves. Once loaded, these curves are retained by the SIM922 though power cycles.

CINI(?) *c* {,*z*,*s*}

Initialize Sensor Calibration

Initialize sensor calibration curve for channel *c*.

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

z	meaning
`LINEAR 0`	volts, kelvin
`SEMILOGT 1`	volts, $\log_{10}$(kelvin)
`SEMILOGV 2`	$\log_{10}$(volts), kelvin
`LOGLOG 3`	$\log_{10}$(volts), $\log_{10}$(kelvin)

The third 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.

The query form of the command, CINI? *c*, 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 *c,f,g*

Add User Curve Point

Add a new point to the channel *c* user curve. *f* is the raw sensor value (in either volts or $\log_{10}$(volts), depending on curve format), and *g* is the corresponding temperature value (in either kelvin or $\log_{10}$(kelvin), again depending on curve format).

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

CAPT? *c,j*

Query User Curve Point

Query the value of the channel *c* user curve, entry point *j*.

The response is
⟨sensor⟩,⟨temperature⟩,
where ⟨sensor⟩ is the raw sensor value (in either volts or $\log_{10}$(volts), depending on curve format), and ⟨temperature⟩ is the corresponding temperature value (in either kelvin or $\log_{10}$(kelvin), again depending on curve format).

CURV(?) *c* {,*z*}

Select Sensor Curve

Set (query) the sensor curve selection for channel *c* {to *z*=(**STAN 0**, USER 1)}.

If *c*=0, all four channels are set (queried).

The built-in standard curve is selected by *z*=STAN. When CURV USER is selected, the user calibration curve (previously loaded with CINI and CAPT) is used.

2.4.7 Serial communication commands

BAUD(?) {*i*}

Baud Rate

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

At power-on, the baud rate defaults to 9600.

Actual baud rate settings depend on implementation details of the SIM922, based on modulo prescalars of the 10 MHz system clock. 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.

FLOW(?) {*z*}	Flow Control Set (query) flow control {to *z*=(`NONE 0`, `RTS 1`, `XON 2`)}. At power-on, the SIM922 defaults to FLOW `RTS` flow control.
PARI(?) {*z*}	Parity Set (query) parity {to *z* = (`NONE 0`, `ODD 1`, `EVEN 2`, `MARK 3`, `SPACE 4`)}. At power-on, the SIM922 defaults to PARI `NONE`.

2.4.8 Status commands

The Status commands query and configure registers associated with status reporting of the SIM922. See Section 2.5 for more details.

*CLS	Clear Status *CLS immediately clears the ESR, CESR, and the OVSR.
*STB? [*i*]	Status Byte Reads the Status Byte register [bit *i*]. Execution of the *STB? query (without the optional bit *i*) always causes the −STATUS signal to be deasserted. Note that *STB? *i* will *not* clear −STATUS, even if bit *i* is the only bit presently causing the −STATUS signal.
*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*}.
CESR? [*i*]	Communication Error Status Query Communication Error Status Register [for bit *i*]. Upon executing a CESR? query, the returned bit(s) of the CESR register are cleared.

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

2.4.9 Interface commands

*RST	Reset Reset the SIM922 to default configuration. The following commands are internally executed upon *RST: • EXON 0,ON • CURV 0,STAN • DTEM ON • SOUT • DISX ON
CONS(?) {*z*}	Console Mode Set (query) the Console mode {to *z*=(0FF 0, ON 1)}. CONS causes each character received at the Input Buffer to be copied to the Output Queue. At power-on and Device-Clear, CONS is set to OFF.

*IDN?

Identify

Read the device identification string.

The identification string is formatted as:
`Stanford_Research_Systems,SIM922,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	Curve point past end

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

LDDE? Device Error

Query the last device error code. The only valid code is:
1 — Curve Erased

LBTN? Button

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

Value	Definition
0	no button pressed since last query
1	[Units]
2	[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 SIM922 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 SIM922 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 two categories of registers in the SIM922 status model:

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.

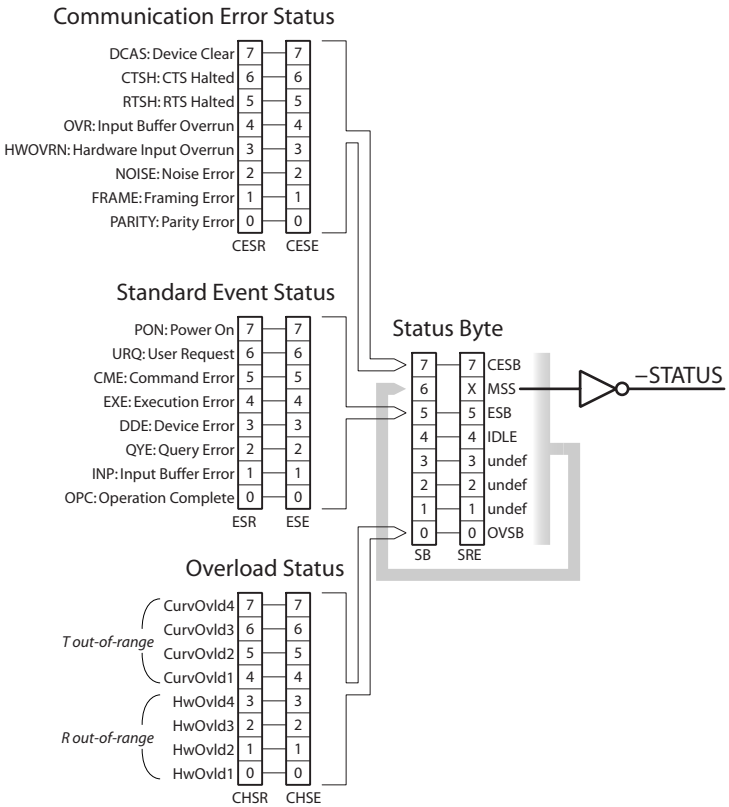


Figure 2.1: Status Register Model for the SIM922 Diode Temperature Monitor.

2.5.1 Status Byte (SB)

The Status Byte is the top-level summary of the SIM922 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 SIM922 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. Indicates a SIM922 had a delayed execution error, due to an illegal mode state. The error code can be queried with LDDE?.

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 : Undefined for the SIM922. Command Error. Indicates a parser-detected error.

CTSH : Undefined for the SIM922.

DCAS : Device Clear. Indicates the SIM922 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 (OVSR)

The Overload 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 OVSR? *i* query) clears only bit *i*.

Weight	Bit	Flag
1	0	HwOvld1
2	1	HwOvld2
4	2	HwOvld3
8	3	HwOvld4
16	4	CurvOvld1
32	5	CurvOvld2
64	6	CurvOvld3
128	7	CurvOvld4

HwOvld*n* : Hardware Overload. Indicates channel *n* resistance measured in excess of ~1500 Ω.

CurvOvld*n* : Curve Out-of-range. Indicates channel *n* resistance measurement falls outside the bounds of the selected calibration curve.

2.5.8 Channel 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.