

Operation and Service Manual

Analog PID Controller

SIM960



Stanford Research Systems

Revision 2.3 • October 10, 2013

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., 2003 – 2013. All rights reserved.

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 9-01558-903



SIM960 Analog PID Controller

Contents

General Information	iii
Safety and Precautions for Use	iii
Symbols	iv
Notation	v
Specifications	vi
1 Getting Started	1 – 1
1.1 General	1–2
1.2 Front Panel Operation	1–2
1.3 Rear Panel Monitoring	1–7
1.4 SIM Interface	1–8
2 Advanced Topics	2 – 1
2.1 PID Tuning Basics	2–2
2.2 Ziegler-Nichols' Tuning	2–5
2.3 Anti-Windup and Conditional Integration	2–7
2.4 Bumpless Transfer	2–8
3 Remote Operation	3 – 1
3.1 Index of Common Commands	3–2
3.2 Alphabetic List of Commands	3–4
3.3 Introduction	3–7
3.4 Commands	3–8
3.5 Status Model	3–25
4 Performance Tests	4 – 1
4.1 Getting Ready	4–2
4.2 Performance Tests	4–2
4.3 Calibration	4–6
5 Circuitry	5 – 1
5.1 Circuit Descriptions	5–2
5.2 Parts Lists	5–4
5.3 Schematic Diagrams	5–7

General Information

Safety and Precautions for Use

Because of the variety of uses for the SIM960, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

 **WARNING**

The SIM960 is not designed, intended, or sold for use in hazardous environments requiring fail-safe operation, including without limitation, operation of nuclear facilities, aircraft or spacecraft control systems, and life support or weapons systems. The user must assure that any failure or misapplication of the SIM960 cannot lead to a consequential failure of any interconnected equipment that could lead to loss of life or limb, or property damage.






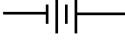



The illustrations, charts, and discussions shown in this manual are intended solely for purposes of example. Since there are many variables and requirements associated with any particular control application, Stanford Research Systems does not assume responsibility or liability for actual use based upon the examples shown in this publication.

Service

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

The SIM960 is a double-wide module designed to be used inside the SIM900 Mainframe. Do not turn on the power to the Mainframe or apply voltage inputs to the module until the module is completely inserted into the mainframe and locked in place. Do not exceed ± 18 V at any input or output connector.

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



WARNING

The following notation will be used throughout this manual.

A warning means that injury or death is possible if the instructions are not obeyed.



CAUTION

A caution means that damage to the instrument or other equipment is possible.

Typesetting conventions used in this manual are:

- 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.

Remote command examples will all be set in monospaced font. In these examples, data sent by the host computer to the SIM960 are set as *straight teletype font*, while responses received by the host computer from the SIM960 are set as *slanted teletype font*.

Specifications

Performance Characteristics

		Min	Typ	Max	Units
Amplifier Settings	Control type	Analog, <i>PID</i> +Offset			
	Input Range	-10		+10	V common mode
		-1		+1	V differential
	Proportional gain	10^{-1}		10^3	V/V
	Integral gain eff. time const.	10^{-2}		5×10^5	1/s
		2×10^{-6}		10^2	s
	Derivative gain	10^{-6}		10	s
	Offset resolution	-10		+10	V
		1		mV	
Amplifier Performance	Bandwidth	100			kHz
	Propagation delay		1		μ s
	Noise ($f > 20$ Hz)		8		nV/ $\sqrt{\text{Hz}}$, RTI
	Output Range	-10		+10	V
Configuration	Parameter control	Digital			
	Parameter accuracy			1	%
	Stability			200	ppm/ $^{\circ}$ C
	Display Resolution		4		digits
Inputs	Measure	BNC, 1 M Ω , ± 10 V range			
	Ext. Setpoint	BNC, 1 M Ω , ± 10 V range			
Setpoint Generator	Setting resolution	-10		+10	V
			1		mV
	Ramp Rate	10^{-3}		10^4	V/s
	Noise ($f > 100$ Hz)			20	nV/ $\sqrt{\text{Hz}}$, RTI
Operating	Temperature [14]	0		40	$^{\circ}$ C
	Power	± 15 , +5			V DC
	Supply current	150 (± 15 V), 80 (+5 V)			mA

General Characteristics

Number of inputs	2
Interface	Serial (RS-232) through SIM interface
Connectors	BNC (3 front, 2 rear); DB-15 (male) SIM interface
Weight	2.1 lbs
Dimensions	3.0" W \times 3.6" H \times 7.0" D

1 Getting Started

This chapter gives you the necessary information to get started quickly with your SIM960 Analog PID Controller.

In This Chapter

1.1	General	1-2
1.2	Front Panel Operation	1-2
1.2.1	Inputs	1-4
1.2.2	Ramping	1-5
1.2.3	Connections	1-5
1.2.4	Bar displays	1-6
1.2.5	Restoring the default configuration	1-6
1.3	Rear Panel Monitoring	1-7
1.3.1	Error Monitor	1-7
1.3.2	Input Setpoint Monitor	1-7
1.4	SIM Interface	1-8
1.4.1	SIM interface connector	1-8
1.4.2	Direct interfacing	1-8

1.1 General

The SIM960 is designed to maintain stability in systems requiring low noise and wide bandwidth. The controller design consists of a front end differential input amplifier, followed by an integrator and a differentiator, arranged in what is known as the “ideal” *PID* topology. The input amplifier (the “error amplifier”) differences the the two single ended inputs, **Setpoint** and **Measure**, and multiplies the resulting error signal (ϵ) by the proportional gain. The amplified error is then passed to three parallel control paths:

1. The proportional path, no change is made to the signal.
2. The integral path with gain *I*.
3. The derivative path gain *D*.

These three signals can be independently selected to combine at a summing amplifier, which is then buffered to the output. A constant offset can also be added, which can be useful in applications that do not use the *I* term. Mathematically, the behavior is

$$\epsilon \equiv \mathbf{Setpoint} - \mathbf{Measure} \quad (1.1)$$

$$\mathbf{Output} = P \times \left\{ \epsilon + I \int \epsilon dt + D \frac{d\epsilon}{dt} \right\} + \mathbf{Offset} \quad (1.2)$$

where the three terms within the braces, and **Offset**, can be independently enabled or zeroed.

For internal stability, the actual differentiator is “rolled off” to limit the derivative gain to +40 dB.

The output circuitry includes a soft limiter that turns on when the output exceeds user specified upper and lower limits and clamps the output to the limit level. The output bar display on the right side of the front panel has red LEDs at each end to indicate when the output is being limited.

1.2 Front Panel Operation

This section discusses the essentials of operating the SIM960 locally, from the front panel. See Chapter 3 for remote operation.

- Press [Select] to choose which configuration parameter to view in the numerical display. The indicator to the left of each descriptor shows which parameter is displayed. When *Shift* is highlighted, pressing [Select] steps the parameter selection backwards.

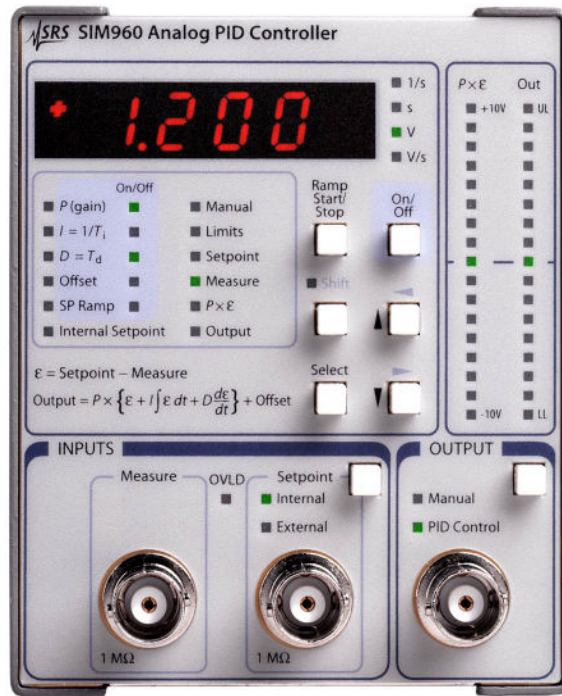


Figure 1.1: The SIM960 front panel.

- The **P**, **I**, **D**, **Offset**, and **SP Ramp** parameters may be enabled/disabled with [On/Off]. Each of these parameters has an additional indicator to the right of the descriptor to indicate the on/off status.
- The **Limits** parameter has two sublevels: upper and lower limit.
- The **Setpoint**, **Measure**, $P \times \epsilon$, and **Output** values are display-only. All the other values can be changed using the [▲▼] buttons; the digit selected for adjustment is indicated by its flashing brightness. Change the digit selection while **Shift** is highlighted ([←→]).
- There are two formats for the numeric display: Exponential, and fixed decimal. The format used for a particular parameter depends on its range. Parameters **P**, **I**, **D** and **SP Ramp** (rate) vary by several orders of magnitude and are therefore displayed in exponential format, while all other parameters range from -10 V to $+10\text{ V}$ and are displayed in fixed decimal format.
- For exponential format, the mantissa may be changed using the up/down arrow buttons. The active digit may be selected using the left/right buttons (= shift, followed by up/down button).

- The right-most digit (after E) is the power of ten exponent. For example, the display $1.2E3 = 1200$.
- Polarity* \implies
- The P parameter has a selectable “ \pm ” indicator before the mantissa. This allows the polarity of the controller to be toggled by the user. All other exponentially displayed parameters are unipolar, so no sign is displayed for these parameters.
 - In fixed decimal format a value between -10 and $+10$ may be selected using [Δ] (and [Shift]).
 - The two outputs, $P \times \epsilon$, and **Output**, are accompanied by bar displays on the right side of the front panel. $P \times \epsilon$ simply ranges from -10 V to $+10$ V. However, since the controller output ranges between the user-programmed upper and lower limits, the output bar display is normalized to that range. For example, if the limits were set to $+5$ V and -1 V, the full range of the bar display would be 6 V, and 0 V would no longer correspond to the center of the bar display, but would be $1/6$ th of the way up from the bottom. The default limits are ± 10 V.
 - Use [Setpoint] in the INPUTS section of the front panel to choose between an external setpoint input, and the internally generated setpoint. An external setpoint can be supplied at the Setpoint BNC input. When the internal setpoint is selected the BNC connector is disconnected from the SIM960 circuitry.
 - The Output BNC connector can be toggled between PID Control mode and Manual mode using [Output] (in the OUTPUT section of the front panel). In manual mode, the SIM960 output is set to the value indicated by the manual parameter.

1.2.1 Inputs

The common mode range of the “Measure” and “Setpoint” inputs extends from -10 V to $+10$ V. If either input is outside this range, the overload LED indicator lights.

The differential input range is ± 1 V. Whenever the difference between Setpoint and Measure exceeds this range, the overload LED indicator turns on. When connected with overall negative feedback and reasonably well tuned, the SIM960 keeps the difference between the setpoint and measure inputs as small as possible, so the differential input range is unlikely to be exceeded. Before the SIM960 has been tuned for a given system, however, this may not be true. It is helpful to keep in mind that exceeding the ± 1 V differential input range will saturate the error amplifier, even if the output signal would otherwise be within the upper and lower **Limits** setting. In such situations, the controller will be effectively limited at some intermediate value.

1.2.2 Ramping

The ramping feature of the SIM960 PID Controller allows the user to linearly slew the internally generated setpoint level from its current value to a new value. The slew rate may be changed using the **SP Ramp** parameter on the front panel.

The indicator to the right of **SP Ramp** shows whether ramping is enabled or disabled. Use [On/Off] (with **SP Ramp** selected) to enable/disable ramping. When disabled, changes to the **Internal Setpoint** parameter take effect immediately. When ramping is enabled, however, changes to **Internal Setpoint** do not immediately take effect. Instead, *Internal* (in the Setpoint block of the INPUTS section of the front panel) begins to blink, showing that a new setpoint has been entered and a ramp event is now pending.

To begin the ramp, press [Ramp Start/Stop]. Now, the *Internal* blink rate doubles, indicating that the setpoint is ramping. To pause the ramp, press [Ramp Start/Stop]. When the ramp is paused, the *Internal* blink rate becomes uneven. To continue the ramp, press [Ramp Start/Stop] again. When the setpoint reaches the new programmed value, the ramp automatically terminates, and *Internal* stops blinking.

Note, **SP Ramp** has no sign in the numerical display. This is because the polarity of the ramp rate is unambiguously determined by whether the newly entered setpoint is greater or less than the current setpoint. The range of available ramp rates is from 1 mV/s to 10,000 V/s. For ramp rates less than or equal to 1 V/s, the rate is dynamically trimmed based on real-time measurements from the onboard A-to-D converter.

1.2.3 Connections

Connect the sensor output of the system to be controlled to the “Measure” input of the SIM960. If an external setpoint is to be supplied, connect this to the “Setpoint” input, and use the button in the INPUTS section of the front panel to select “External” input. Before connecting the SIM960 output to the system to control, it may be necessary to set the user programmable output upper and lower **Limits** to guard against damaging the system. Care should be taken to insure that the programmed output range is consistent with the system input range. Once the limits have been programmed, connect the SIM960 output to the system input.

1.2.4 Bar displays

Two LED bar displays have been included on the right side of the SIM960 front panel to provide visual information about the $P \times \epsilon$ and **Output** signals. This reduces the need to frequently return to those fields on the numerical display while trying to adjust other tuning parameters. Some time should be taken to understand what information these bar displays provide.

Each bar has two lighted LEDs; one for the maximum peak of the signal, and one for the minimum peak. The peaks are determined with respect to time variation of the signal, and they decay back to the DC level with a decay time of ~100 ms.

To understand how a signal is represented in the bar display, consider an input sine wave of frequency 1 Hz. Since frequency is low compared to the inverse of the decay time, the maximum and minimum peak values are indistinguishable, and the signal appears as a single LED that tracks the sine wave. As the frequency increases, the maximum peak does not decay quickly enough to track the negative excursions the signal, and the minimum peak also fails to track positive excursions. So there appear to be two lighted LEDs slightly separated, roughly tracking the sine wave. As the frequency is further increased to well above the decay time inverse, the two lighted LEDs no longer decay at all from their peak levels, so there appear to be two lighted LEDs marking the maximum and minimum peaks of the sine wave.

Thus, a slowly varying signal appears as a single lighted LED in the display, tracking the signal changes with time. But a quickly varying signal, however, appears as two lighted LEDs marking the maximum and minimum excursions of the signal in time.

The range of the $P \times \epsilon$ bar display is ± 10 V. The **Output** bar display has a range determined by the user programmed upper and lower limits. For example, if the limits were set to +5 V and -1 V, the full range of the bar display would be 6 V, and 0 V would no longer correspond to the center of the bar display, but would be 1/6th of the way up from the bottom. Also, the **Output** bar display has a red LED on each end to indicate whether the controller output is saturated at its limit.

1.2.5 Restoring the default configuration

The default configuration of the SIM960 can be restored in either of two ways: From the front panel, or via the remote interface.

To restore from the front panel, first turn off the power to the SIM960 by switching its SIM900 Mainframe to “Standby,” then switch the

power on while holding down [Ramp Start/Stop]. Keep the button depressed for about one second after power comes on.

The default configuration can also be restored via the remote interface using the *RST command.

1.3 Rear Panel Monitoring

Two analog monitor signals are available at the rear panel of the SIM960 (see Figure 1.2).



Figure 1.2: The SIM960 rear panel.

1.3.1 Error Monitor

The upper BNC connector carries a buffered copy of the $P \times \epsilon$ output of the error amplifier. This output is always available, even when the P term is disabled from the control law. It is also active when the main SIM960 output is set to Manual mode.

1.3.2 Input Setpoint Monitor

The lower BNC is a copy of the internally generated setpoint voltage. This output is also always available, even when the Setpoint mode is set to External.

1.4 SIM Interface

The primary connection to the SIM960 Analog PID Controller is the rear-panel DB-15 SIM interface connector. Typically, the SIM960 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 SIM960 directly, without using the SIM900 Mainframe. This section provides details on the interface.



CAUTION

The SIM960 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 SIM960 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 1.1

Pin	Signal	Direction Src ⇒ Dest	Description
1	SIGNAL_GND	MF ⇒ SIM	Ground reference for signal
2	-STATUS	SIM ⇒ MF	Status/service request (GND = asserted, +5 V = 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	-5 V	MF ⇒ SIM	Power supply (no connection in SIM960)
7	-15 V	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	+5 V	MF ⇒ SIM	Power supply (digital circuitry)
14	+15 V	MF ⇒ SIM	Power supply (analog circuitry)
15	+24 V	MF ⇒ SIM	Power supply (no connection in SIM960)

Table 1.1: SIM Interface Connector Pin Assignments, DB-15

1.4.2 Direct interfacing

The SIM960 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 1.1. 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 SIM960 independent of the SIM900 Mainframe, communication is usually possible by directly connecting the appropriate interface lines from the SIM960 DB-15 plug to the RS-232 serial port of a personal computer.¹ Connect RXD from the SIM960 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 SIM960, 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 1.2.

DB-15/F to SIM960	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 1.2: SIM960 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 SIM960 supports is 38400. The minimum baud rate is 110. Above 38400, the SIM960 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 SIM960 and the SIM900 Mainframe.

2 Advanced Topics

This chapter discusses a simple “closed-loop” tuning procedure, along with some of the advanced features of the SIM960 Analog PID Controller.

In This Chapter

2.1	PID Tuning Basics	2-2
2.2	Ziegler-Nichols’ Tuning	2-5
2.2.1	Open-loop tuning	2-5
2.2.2	Closed-loop tuning	2-6
2.3	Anti-Windup and Conditional Integration	2-7
2.4	Bumpless Transfer	2-8
2.4.1	Manual-to-PID	2-8
2.4.2	PID-to-Manual	2-8

2.1 PID Tuning Basics

PID control provides a simple way to minimize the effect of disturbances to a system. The system consists of a closed feedback loop between two elements, the SIM960 **controller** and the user **process**. The controller has two inputs, **Setpoint** and **Measure**, and an **Output**. The process consists of a power source that can be directly changed by the controller, in conjunction with a sensor to monitor the process behavior. The sensor signal, after any necessary conditioning, is the process output. This should be connected to the **Measure** input of the SIM960, and the SIM960 **Output** should be connected to the process input, forming a feedback loop.

The difference between the **Setpoint** and **Measure** inputs is the error signal, $\varepsilon \equiv \text{Setpoint} - \text{Measure}$ (Eqn 1.1). In the SIM960 the error signal is amplified by the proportional gain. The controller uses the amplified error, $P \times \varepsilon$, to generate three control signals:

1. Proportional, the P amplified error with no changes.
2. Integral, the time integral of the amplified error signal multiplied by a gain coefficient I .
3. Derivative, the time derivative of the amplified error signal multiplied by a gain coefficient D .

These signals, as well as an **Offset**, are combined at a summing junction to produce the controller output (see Eqn 1.2). Figure 2.1 shows a schematic representation of the SIM960 controller topology. Note the proportional gain coefficient is common to all three terms, so the net integral and derivative gains are $P \times I$ and $P \times D$, respectively, whether or not proportional control is enabled.

The controller monitors the process output and makes small adjustments to the process in order to minimize deviations of **Measure** from **Setpoint** due to external disturbances. To accomplish this, the controller must be properly tuned, meaning that the gains for each of the three control signals—proportional, integral, and derivative—must be chosen appropriately to match the behavior of the process. A well-tuned controller should be able to maintain a stable process output.

The control loop feedback should be negative. However, because the polarity of the process response to the controller output is an arbitrary function of the design of the system, it is vital that the controller polarity be chosen properly. Based on the topology of the SIM960 design, feedback polarity can be changed simply by changing the polarity of the proportional gain parameter P . The user must first determine which polarity will provide negative feedback. If the

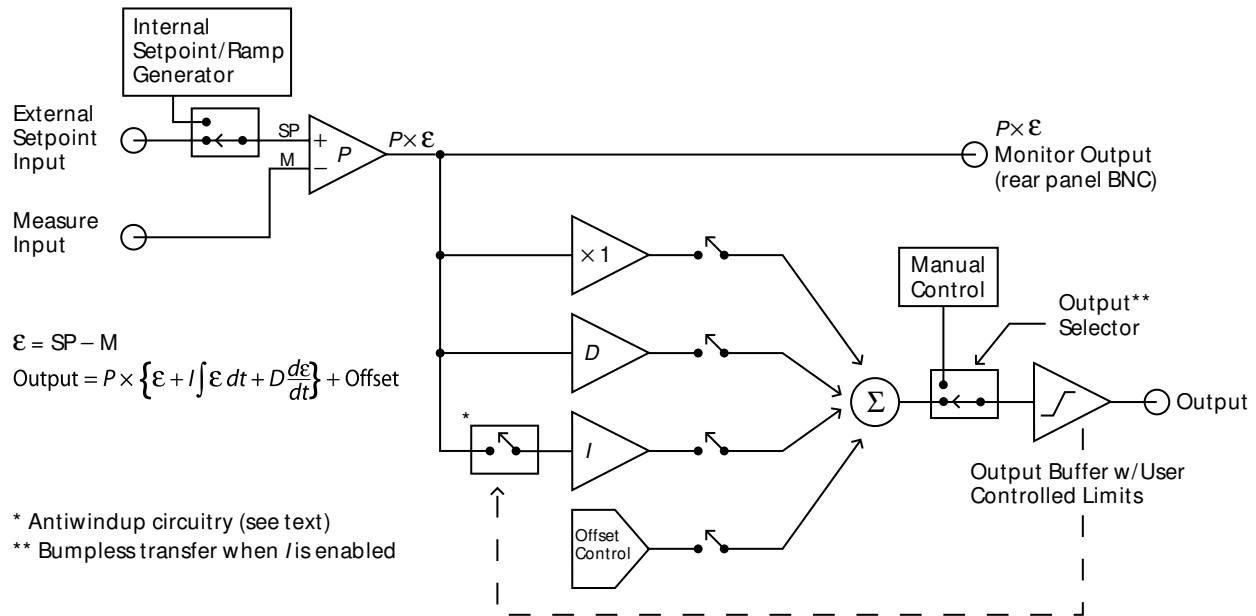


Figure 2.1: The SIM960 block diagram.

process is noninverting, i.e. a small positive change at its input results in a positive change at its output, then using positive P polarity will ensure negative feedback in the loop. To see this, follow the effect of a small positive change at the process output. Since the process output is connected to the **Measure** input of the SIM960, a small positive change would cause a negative change to ϵ . The resulting change at the controller **Output** would also be negative, as would be that of the process output. Thus, the initial small positive change at the process output is “corrected” by a negative change after going around the feedback loop. As a general rule, if the process is noninverting, then the P -polarity should be positive. If the process is inverting, negative P -polarity should be used.

Care should be taken in designing the process. The sensor should be situated so that it is responsive to changes to the part of the system under control. Placing the sensor too remotely can result in a time delay which limits the quality of control. Also, the sensor should primarily measure the system’s *response* to external changes, rather than measure the changes directly. The latter can sometimes be used to help the controller anticipate transients, but at the risk of sacrificing accuracy in reaching the target setpoint.

Tuning a PID controller amounts to determining what the relative contributions should be from each of the three types of control. The simplest approach is to start with proportional control and add inte-

gral and derivative one at a time. A simple P -controller generates a control variable that is proportional to the error signal.¹

Increasing the P gain should cause the process output to respond by moving closer to the setpoint. Generally, enough amplification should be used so that the process output is brought reasonably close to the setpoint. Too much gain, however, will cause the system to oscillate. Start with a small P gain, and increase by factors of two until the system begins to oscillate. Then back off in small amounts until stability is recovered.

While it is possible to maintain stability with a simple P controller, in general this will lead to a finite, non-zero ϵ . Increasing P will tend to reduce the resulting ϵ , but too much proportional gain will eventually lead to oscillations.

One way to eliminate this nonzero error problem is to include an offset at the controller output. The SIM960 **Offset** parameter can be turned on and adjusted to hold the process power at a level that maintains a smaller error. However, this is only a coarse improvement, since the necessary power level may change with time.

Integral control provides an “automatic” way to dynamically adjust the effective offset to zero the error; in older controllers, integral action was called “automatic reset” for this reason. Integral control simply integrates the error signal with respect to time. Thus the controller output changes until the error has been reduced to zero, near which point the controller output slows and stops changing. If the error drifts over time, the integrator responds by adjusting the controller output to cancel the error. So it is much like having a dynamic output offset constantly responding to system changes. As with proportional gain, too much integral gain can cause oscillation. Again, start with a small I gain and increase by factors of two until oscillation begins, then back off until stability is recovered.

Though integral control is effective at reducing the error, it is not as effective as proportional control at responding quickly to changes. This is because the integrator needs time to build up a response. To further enhance the response of the process to rapid changes, derivative control is often employed. Derivative control is proportional to the rate of change of the error, so it is relatively unresponsive to slow changes, but rapid changes to the system produce a significant response. Derivative control reduces oscillations that can result from step changes to a system.

¹ During the tuning process, it is important to keep in mind that the differential input range of the SIM960 is ± 1.0 V. It is good practice to occasionally glance at the *OVL*D indicator to ensure the input amplifier is not saturated.

2.2 Ziegler-Nichols' Tuning

For many applications, a good starting point for tuning is one of the two classic Ziegler-Nichols methods². These two methods are briefly described below; for more details, see, for example, Åström & Hägglund, **PID Controllers: Theory, Design, and Tuning**, Instrument Society of America (1995).

2.2.1 Open-loop tuning

The open-loop Ziegler-Nichols method involves introducing a small step change to the process under control, and making a few measurements from the response. The procedure is:

- Switch the SIM960 into **Manual** mode, and then adjust until the process is stable and near the desired operating point.
- Now make a small, sudden step change, Δ , in the control signal. Call this time $t = 0$.
- Record the process response in the **Measure** signal. Define the (dimensionless) process step-response function:

$$h(t) = \frac{\mathbf{Measure}(t) - \mathbf{Measure}(0)}{\Delta}$$

- Observe the point of maximum slope in h . Extend a straight line through this point, tangent to h , downward (see Figure 2.2).
- Let L be the time coordinate where the straight line crosses $h = 0$; and let a be the negative of the h -intercept (i. e., $a > 0$ in Figure 2.2).
- Note that it is *not* necessary to wait for the process to completely settle following the step change Δ —it is sufficient to simply wait until the maximum slope is observed in **Measure**.

From a and L , Ziegler and Nichols suggest tuning for P , PI , and PID control as shown in Table 2.1

Control	P	I	D
P	$1/a$		
PI	$0.9/a$	$1/(3L)$	
PID	$1.2/a$	$1/(2L)$	$L/2$

Table 2.1: Ziegler-Nichols open-loop tuning parameters

² Ziegler, J. G., & Nichols, N. B. 1942, Trans. ASME, 64, 759

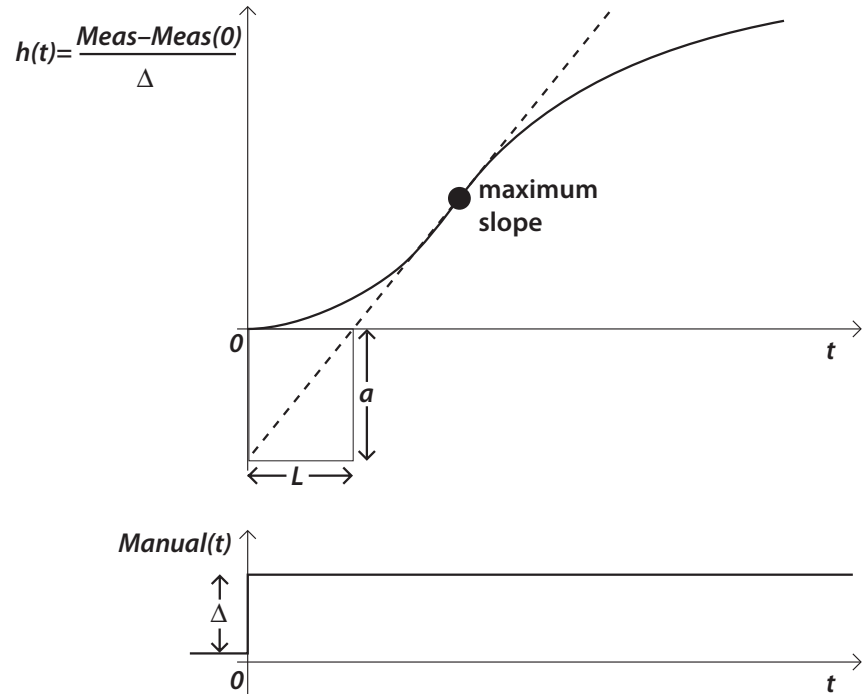


Figure 2.2: The open-loop step response of the process.

2.2.2 Closed-loop tuning

An alternate method, also due to Ziegler and Nichols, is based on measuring the gain at which the process just begins to oscillate. The procedure is:

- Switch the SIM960 into PID mode, with *I* and *D* both *disabled*. Choose a value for **Setpoint** around the desired operating point, and set *P* so some small value.
- Slowly increase *P* until the process starts to oscillate.
- Record this value of *P* as K_u , the “ultimate” gain. Also observe the period of the oscillations, T_u .

From K_u and T_u , Ziegler and Nichols again suggest tuning for *P*, *PI*, and *PID* control as shown in Table 2.2

Control	<i>P</i>	<i>I</i>	<i>D</i>
<i>P</i>	$K_u/2$		
<i>PI</i>	$2K_u/5$	$5/(4T_u)$	
<i>PID</i>	$3K_u/5$	$2/T_u$	$T_u/8$

Table 2.2: Ziegler-Nichols closed-loop tuning parameters

2.3 Anti-Windup and Conditional Integration

For better integral performance, the SIM960 features anti-windup circuitry in the form of conditional integration. The purpose of anti-windup is to improve the controller's ability to recover from output saturation. When the output saturates, the error is likely to be large, since the process is unable to provide power fast enough to recover the process output. However, the integrator contribution may not account for the full amount of the controller output in this case. Subsequently, the integrator continues to integrate the error until the integrator output saturates. This "winding up" aspect of integral control becomes a problem when the process recovers and the error level passes through zero, because the error must move significantly beyond zero for the integrator to "unwind" from saturation. In general, once the controller output is clamped at a limit, nothing is accomplished by driving it harder into that limit by more integration. In fact, it only makes it harder to recover from saturation, since the result is usually large swings back and forth from limit to limit.

There are a variety of anti-windup strategies to mitigate this effect. A simple way to implement anti-windup is to switch off the integrator whenever the output saturates. This is not the same as resetting the integrator (zeroing its output by discharging the feedback capacitance) because the output simply stops moving, but does not go to zero. It is equivalent to momentarily zeroing the integrator *input*, so that there is no signal to integrate while the output is saturated.

An improvement to this scheme comes from recognizing that not all saturation conditions cause unwanted integrator wind-up. For example, suppose the controller/process history were such as to produce the following conditions:

- Error signal negative
- Integrator output finite, not saturated
- Controller output saturated at the positive limit

Then, the integrator output would be moving in the negative direction, since its input, the error, is negative. This would not cause the controller output to be pushed harder into saturation; in fact it may eventually pull it out of saturation. So stopping the integrator would hinder the controller's effort to recover the process variable. The SIM960 uses a technique called "conditional integration:" *Conditional integration only stops the integrator when the polarity of the error is such as to drive the integrator toward the saturated limit.*

2.4 Bumpless Transfer

When switching the output mode between **Manual** and PID Control, transients on the output signal can disturb the system under control. Minimizing these switching transients is known as “bumpless transfer.” The SIM960 supports bumpless transfer under certain conditions, as described below.

2.4.1 Manual-to-PID

When switching from **Manual** output to PID Control output, bumpless transfer is only possible if the integral term is enabled.³ When *I* is turned on and the SIM960 is in **Manual** output mode, the input to the integrator is rerouted to integrate the difference between **Manual** and the (deselected) PID Control output. This effectively allows the PID Control to “track” the **Manual** value, presetting the integrator as necessary. Then, when the output is switched back to PID Control, the controller output is already the same as the **Manual** output level. Were this not the case, the integrator output would likely saturate while in manual mode, and upon switching to PID Control mode, the controller output would suddenly jump. Bumpless transfer insures that the transition from **Manual** to PID Control mode is smooth.

2.4.2 PID-to-Manual

An additional feature of the SIM960 is the ability to preset the manual level to the current PID control output level, so that switching from PID mode to manual mode will also be smooth. With the module in PID mode, select the **Manual** field. Press [On/Off] and hold it down for at least one second. After one second the manual display reading will shift to the current PID output level. The output mode will remain in PID control mode until it is manually switched on the front panel or through the remote interface. But the new manual output level will be equal to the PID control output.

³ This can be understood mathematically, since only the integral term has an “unspecified” initial offset value that can be set to an arbitrary value without violating Eqn 1.2.

3 Remote Operation

This chapter describes operating the module over the serial interface.

In This Chapter

3.1	Index of Common Commands	3-2
3.2	Alphabetic List of Commands	3-4
3.3	Introduction	3-7
3.3.1	Power-on configuration	3-7
3.3.2	Buffers	3-7
3.3.3	Device Clear	3-7
3.4	Commands	3-8
3.4.1	Command syntax	3-8
3.4.2	Examples	3-9
3.4.3	Controller settings commands	3-10
3.4.4	Controller configuration commands	3-12
3.4.5	Monitor commands	3-14
3.4.6	Display commands	3-16
3.4.7	Serial communication commands	3-17
3.4.8	Status commands	3-17
3.4.9	Interface commands	3-20
3.5	Status Model	3-25
3.5.1	Status Byte (SB)	3-26
3.5.2	Service Request Enable (SRE)	3-26
3.5.3	Standard Event Status (ESR)	3-27
3.5.4	Standard Event Status Enable (ESE)	3-27
3.5.5	Communication Error Status (CESR)	3-27
3.5.6	Communication Error Status Enable (CESE)	3-28
3.5.7	Instrument Status (INCR)	3-28
3.5.8	Instrument Status (INSR)	3-29
3.5.9	Analog to Digital Status Enable (INSE)	3-29
3.5.10	Analog to Digital Status (ADSR)	3-29
3.5.11	Analog to Digital Status Enable (ADSE)	3-30

3.1 Index of Common Commands

symbol	definition
<i>i,j</i>	Integers
<i>f,g</i>	Floating-point values
<i>z</i>	Literal token
(?)	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

Controller Settings

PCTL(?) <i>z</i>	3 – 10	Proportional action ON/OFF
ICTL(?) <i>z</i>	3 – 10	Integral action ON/OFF
DCTL(?) <i>z</i>	3 – 10	Derivative action ON/OFF
OCTL(?) <i>z</i>	3 – 10	Offset ON/OFF
GAIN(?) { <i>f</i> }	3 – 10	Proportional Gain
APOL(?) <i>z</i>	3 – 11	Controller Polarity
INTG(?) { <i>f</i> }	3 – 11	Integral Gain
DERV(?) { <i>f</i> }	3 – 11	Derivative Gain
OFST(?) { <i>f</i> }	3 – 11	Output Offset

Controller Configuration

AMAN(?) <i>z</i>	3 – 12	Output (Manual Output/PID Control)
INPT(?) <i>z</i>	3 – 12	Input (Internal/External Setpoint)
SETP(?) { <i>f</i> }	3 – 12	New setpoint
RAMP(?) <i>z</i>	3 – 12	Internal setpoint ramping ON/OFF
RATE(?) { <i>f</i> }	3 – 12	Setpoint ramping Rate
RMPS?	3 – 13	Setpoint ramping status
STRT <i>z</i>	3 – 13	Pause or continue ramping
MOUT(?) { <i>f</i> }	3 – 13	Manual Output
ULIM(?) { <i>f</i> }	3 – 13	Upper Output Limit
LLIM(?) { <i>f</i> }	3 – 14	Lower Output Limit

Monitor

SMON? [<i>i</i>]	3 – 14	Setpoint Input Monitor
MMON? [<i>i</i>]	3 – 14	Measure Input Monitor
EMON? [<i>i</i>]	3 – 15	Amplified Error Monitor
OMON? [<i>i</i>]	3 – 15	Output Monitor
RFMT(?) { <i>z</i> }	3 – 15	Output Streaming Records Format
SOUT [<i>z</i>]	3 – 16	Stop Streaming
FPLC(?) { <i>i</i> }	3 – 16	Frequency of Power Line Cycle

Display

DISP(?) {z}	3-16 Select Field
SHFT(?) {z}	3-16 Shift Status
DISX(?) {z}	3-17 Front Panel Display Enable

Serial Communications

BAUD(?) {i}	3-17 Baud Rate
FLOW(?) {z}	3-17 Flow Control
PARI(?) {z}	3-17 Parity

Status

*CLS	3-17 Clear Status
*STB? [i]	3-18 Status Byte
*SRE(?) [i, j]	3-18 Service Request Enable
*ESR? [i]	3-18 Standard Event Status
*ESE(?) [i, j]	3-18 Standard Event Status Enable
CESR? [i]	3-18 Comm Error Status
CESE(?) [i, j]	3-19 Comm Error Status Enable
INCR? [i]	3-19 Instrument condition register
INSR? [i]	3-19 Instrument status register
INSE(?) [i, j]	3-19 Instrument status enable register
ADSR? [i]	3-19 A-to-D status register
ADSE(?) [i, j]	3-19 A-to-D status enable register
PSTA(?) {z}	3-20 Pulse -STATUS Mode

Interface

*RST	3-20 Reset
CONS(?) {z}	3-21 Console Mode
*IDN?	3-21 Identify
*TST?	3-21 Self Test
*OPC(?)	3-22 Operation Complete
WAIT i	3-22 Wait
LEXE?	3-22 Execution Error
LCME?	3-23 Command Error
LBTN?	3-23 Last Button
TOKN(?) {z}	3-24 Token Mode
TERM(?) {z}	3-24 Response Termination

3.2 Alphabetic List of Commands

★

*CLS	3 – 17	Clear Status
*ESE(?) [i,] {j}	3 – 18	Standard Event Status Enable
*ESR? [i]	3 – 18	Standard Event Status
*IDN?	3 – 21	Identify
*OPC(?)	3 – 22	Operation Complete
*RST	3 – 20	Reset
*SRE(?) [i,] {j}	3 – 18	Service Request Enable
*STB? [i]	3 – 18	Status Byte
*TST?	3 – 21	Self Test

A

ADSE(?) [i,] {j}	3 – 19	A-to-D status enable register
ADSR? [i]	3 – 19	A-to-D status register
AMAN(?) z	3 – 12	Output (Manual Output/PID Control)
APOL(?) z	3 – 11	Controller Polarity

B

BAUD(?) {i}	3 – 17	Baud Rate
-------------	--------	-----------

C

CESE(?) [i,]{j}	3 – 19	Comm Error Status Enable
CESR? [i]	3 – 18	Comm Error Status
CONS(?) {z}	3 – 21	Console Mode

D

DCTL(?) z	3 – 10	Derivative action ON/OFF
DERV(?) {f}	3 – 11	Derivative Gain
DISP(?) {z}	3 – 16	Select Field
DISX(?) {z}	3 – 17	Front Panel Display Enable

E

EMON? [i]	3 – 15	Amplified Error Monitor
-----------	--------	-------------------------

F

FLOW(?) {z}	3 – 17	Flow Control
FPLC(?) {i}	3 – 16	Frequency of Power Line Cycle

G

GAIN(?) {f}	3 – 10	Proportional Gain
-------------	--------	-------------------

I	
ICTL(?) z	3 – 10 Integral action ON/OFF
INCR? [i]	3 – 19 Instrument condition register
INPT(?) z	3 – 12 Input (Internal/External Setpoint)
INSE(?) [i], {j}	3 – 19 Instrument status enable register
INSR? [i]	3 – 19 Instrument status register
INTG(?) {f}	3 – 11 Integral Gain

L	
LBTN?	3 – 23 Last Button
LCME?	3 – 23 Command Error
LEXE?	3 – 22 Execution Error
LLIM(?) {f}	3 – 14 Lower Output Limit

M	
MMON? [i]	3 – 14 Measure Input Monitor
MOUT(?) {f}	3 – 13 Manual Output

O	
OCTL(?) z	3 – 10 Offset ON/OFF
OFST(?) {f}	3 – 11 Output Offset
OMON? [i]	3 – 15 Output Monitor

P	
PARI(?) {z}	3 – 17 Parity
PCTL(?) z	3 – 10 Proportional action ON/OFF
PSTA(?) {z}	3 – 20 Pulse –STATUS Mode

R	
RAMP(?) z	3 – 12 Internal setpoint ramping ON/OFF
RATE(?) {f}	3 – 12 Setpoint ramping Rate
RFMT(?) {z}	3 – 15 Output Streaming Records Format
RMPS?	3 – 13 Setpoint ramping status

S	
SETP(?) {f}	3 – 12 New setpoint
SHFT(?) {z}	3 – 16 Shift Status
SMON? [i]	3 – 14 Setpoint Input Monitor
SOUT [z]	3 – 16 Stop Streaming
STRT z	3 – 13 Pause or continue ramping

T	
TERM(?) {z}	3 – 24 Response Termination

TOKN(?) {z} 3 – 24 Token Mode

U

ULIM(?) {f} 3 – 13 Upper Output Limit

W

WAIT *i* 3 – 22 Wait

3.3 Introduction

Remote operation of the SIM960 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 SIM mainframe or directly via RS-232 (see section 1.4.2.1).

See Table 1.1 for the specification of the DB-15 SIM Interface Connector.

3.3.1 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 SIM960 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**.

3.3.2 Buffers

The SIM960 stores incoming bytes from the host interface 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 SIM960 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.

3.3.3 Device Clear

The SIM960 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 SIM960 is left unchanged; to reset the meter, see *RST.

The Device Clear signal will also terminate any streaming outputs from the SIM960 due to an SMON?, MMON?, EMON? and/or OMON? query of multiple conversions.

3.4 Commands

This section provides syntax and operational descriptions for remote commands.

3.4.1 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 should be 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 the command 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.

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
(?)	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

3.4.2 Examples

Each command is provided with a simple example illustrating its usage. In these examples, all data sent by the host computer to the SIM960 are set as straight teletype font, while responses received the host computer from the SIM960 are set as *slanted teletype font*.

The usage examples vary with respect to set/query, optional parameters, and token formats. These examples are not exhaustive, but are intended to provide a convenient starting point for user programming.

3.4.3 Controller settings commands

PCTL(?) z	<p>Proportional action ON/OFF</p> <p>Set (query) the proportional control {to z=(OFF 0, ON 1)}.</p> <p>When ON, the PID Control path includes the proportional control term.</p> <p><i>Example:</i> PCTL 1</p>
<hr/>	
ICTL(?) z	<p>Integral action ON/OFF</p> <p>Set (query) the integral control {to z=(OFF 0, ON 1)}.</p> <p>When ON, the PID Control path includes the integral control term.</p> <p><i>Example:</i> ICTL? 0</p>
<hr/>	
DCTL(?) z	<p>Derivative action ON/OFF</p> <p>Set (query) the derivative control {to z=(OFF 0, ON 1)}.</p> <p>When ON, the PID Control path includes the derivative control term.</p> <p><i>Example:</i> DCTL OFF</p>
<hr/>	
OCTL(?) z	<p>Offset ON/OFF</p> <p>Set (query) the offset control {to z=(OFF 0, ON 1)}.</p> <p>When ON, the PID Control path includes the constant output offset.</p> <p><i>Example:</i> OCTL? ON</p>
<hr/>	
GAIN(?) {f}	<p>Proportional Gain</p> <p>Set (query) the proportional gain (P) {to f}, in V/V.</p> <p>Values may be entered in decimal or exponential format, and are signed.</p> <p>GAIN may be set with 2 digits of resolution for $1.0 \leq P \leq 10^3$, and with single-digit resolution for $0.1 \leq P \leq 0.9$. Note that setting GAIN does <i>not</i> modify whether the proportional term is enabled or disabled. For on/off control, see PCTL.</p> <p>Setting GAIN overrides the previous setting of APOL.</p> <p>The allowed range for GAIN is $10^{-1} \leq P \leq 10^3$.</p> <p><i>Example:</i> GAIN +2.5E+2</p>

APOL(?) z	<p>Controller Polarity</p> <p>Set (query) the proportional gain polarity {to z=(POS 1, NEG 0)}. Setting APOL will override the sign of a previously-commanded GAIN.</p> <p><i>Example:</i> APOL? POS</p>
<hr/>	
INTG(?) {f}	<p>Integral Gain</p> <p>Set (query) the integral gain (I) {to f}, in V/(V·s).</p> <p>INTG may be set with 2 digits of resolution for $10^{-1} \leq I \leq 5 \times 10^5$, and with single-digit resolution for $10^{-2} \leq I \leq 9 \times 10^{-2}$. Integral gains are unsigned (positive values only). Note that setting INTG does <i>not</i> modify whether the integrator is enabled or disabled. For on/off control, see ICTL.</p> <p>The allowed range for INTG is $10^{-2} \leq I \leq 5 \times 10^5$.</p> <p><i>Example:</i> INTG? +1.5E+3</p>
<hr/>	
DERV(?) {f}	<p>Derivative Gain</p> <p>Set (query) the derivative gain {to f}, in V/(V/s).</p> <p>DERV may be set with 2 digits of resolution for $10^{-5} \leq D \leq 10$, and with single-digit resolution for $10^{-6} \leq D \leq 9 \times 10^{-6}$. Derivative gains are unsigned (positive values only). Note that setting DERV does <i>not</i> modify whether the derivative is enabled or disabled. For on/off control, see DCTL.</p> <p>The allowed range for DERV is $10^{-6} \leq D \leq 10$.</p> <p><i>Example:</i> DERV 0.000015 DERV? +1.5E-5</p>
<hr/>	
OFST(?) {f}	<p>Output Offset</p> <p>Set (query) the output offset {to f}, in volts.</p> <p>The offset voltage can be set with millivolt resolution. Note that setting OFST does <i>not</i> modify whether the offset is enabled or disabled. For on/off control, see OCTL.</p> <p>The allowed range is $-10.000 \leq \text{OFST} \leq +10.000$.</p> <p><i>Example:</i> OFST -12.3E-2; OFST? -0.123</p>

3.4.4 Controller configuration commands

AMAN(?) z	<p>Output (Manual Output/PID Control)</p> <p>Set (query) controller output state {to z=(MAN 0, PID 1)}.</p> <p><i>Example:</i> AMAN? 1</p>
-----------	--

INPT(?) z	<p>Input (Internal/External Setpoint)</p> <p>Set (query) setpoint input state {to z=(INT 0, EXT 1)}.</p> <p><i>Example:</i> INPT INT</p>
-----------	--

SETP(?) {f}	<p>New setpoint</p> <p>Set (query) the setpoint value {to f}, in volts.</p> <p>The setpoint can be set with millivolt resolution. If ramping is enabled (see RAMP), SETP will initiate a ramp to f. Otherwise, the setpoint value changes immediately to the new value.</p> <p>The allowed range is $-10.000 \leq \text{SETP} \leq +10.000$.</p> <p><i>Example:</i> SETP -1.234</p>
-------------	--

RAMP(?) z	<p>Internal setpoint ramping ON/OFF</p> <p>Set (query) internal setpoint ramping {to z=(OFF 0, ON 1)}.</p> <p>When ON, the changes to the internal setpoint are made with constant slew-rate ramping enabled.</p> <p><i>Example:</i> RAMP 1</p>
-----------	---

RATE(?) {f}	<p>Setpoint ramping Rate</p> <p>Set (query) the setpoint rate {to f}, in V/s.</p> <p>RATE may be set with 2 digits of resolution for values above 10^{-2}, and with single-digit resolution below that. Note that setting RATE does <i>not</i> modify whether setpoint changes are made with constant slew-rate ramping or not. For on/off control of linear ramping, see RAMP.</p> <p>The allowed range is $10^{-3} \leq \text{RATE} \leq 10^4$.</p> <p><i>Example:</i> RATE 2.2E-3 RATE? +0.2E-2</p>
-------------	--

RMPS?	<p>Setpoint ramping status</p> <p>Query the ramp status.</p> <p>For slow ramps of the internal setpoint, the RMPS? query will monitor the real-time status of a setpoint transition.</p> <p>The response is one of the following token values: IDLE 0, PENDING 1, RAMPING 2, PAUSED 3.</p> <p><i>Example:</i> RMPS? RAMPING</p>
<hr/>	
STRT z	<p>Pause or continue ramping</p> <p>Cause a setpoint ramping event in progress to pause (STOP) or continue (START). z=(STOP 0, START 1).</p> <p>Note that STRT <i>cannot</i> be used to initiate a new setpoint transition from the RMPS PENDING state—this can only be accomplished by pressing [Ramp Start/Stop] on the front panel.</p> <p><i>Example:</i> STRT START</p>
<hr/>	
MOUT(?) {f}	<p>Manual Output</p> <p>Set (query) the manual output value {to f}, in volts.</p> <p>The manual output can be set with millivolt resolution. Note that setting MOUT does <i>not</i> modify whether the controller is in manual or <i>PID</i> control mode. For on/off control of manual output, see AMAN.</p> <p>The allowed range for MOUT is $-10.000 \leq \text{MOUT} \leq +10.000$.</p> <p><i>Example:</i> MOUT? +8.000</p>
<hr/>	
ULIM(?) {f}	<p>Upper Output Limit</p> <p>Set (query) the upper output limit {to f}, in volts.</p> <p>The upper limit can be set with 10 mV resolution. Note that, regardless of the operating mode of the SIM960 (see AMAN), the output voltage will always be clamped to remain less positive than the ULIM limit. Combined with the LLIM limit, this results in the output obeying:</p> $-10.00 \leq \text{LLIM} \leq \text{Output} \leq \text{ULIM} \leq +10.00$ <p>The allowed range is $\text{LLIM} \leq \text{ULIM} \leq +10.00$.</p>

LLIM(?) { <i>f</i> }	<p>Lower Output Limit</p> <p>Set (query) the lower output limit {to <i>f</i>}, in volts.</p> <p>The lower limit can be set with 10 mV resolution. The output voltage of the SIM960 will always be clamped to remain less negative than the LLIM limit. See ULIM for more details.</p> <p>The allowed range is $-10.00 \leq \text{LLIM} \leq \text{ULIM}$.</p>
----------------------	--

3.4.5 Monitor commands

SMON? [<i>i</i>]	<p>Setpoint Input Monitor</p> <p>Query the Setpoint input voltage to the error amplifier, in volts.</p> <p>If INPT INT is set, then SMON? monitors the value of the internally-generated setpoint. If INPT EXT, then SMON? monitors the voltage applied at the front-panel Setpoint BNC input.</p> <p><i>i</i> is an optional parameter that causes streaming of Setpoint data. If <i>i</i> is specified, then <i>i</i> measurements will be output at a rate of approximately two measurements per second. If <i>i</i> is specified as 0, then measurements will be output indefinitely. The SOUT command can be used to stop streaming.</p>
--------------------	---

Example: SETP 1.2; SMON? 5
+01.004496
+01.066567
+01.128909
+01.191273
+01.200073

MMON? [<i>i</i>]	<p>Measure Input Monitor</p> <p>Query the Measure input voltage to the error amplifier, in volts.</p> <p>MMON? always reports the voltage applied at the front-panel Measure BNC input.</p> <p><i>i</i> is an optional parameter that causes streaming of Measure data. If <i>i</i> is specified, then <i>i</i> measurements will be output at a rate of approximately half a second per measurement. If <i>i</i> is specified as 0, then measurements will be output indefinitely. The SOUT command can be used to stop streaming.</p>
--------------------	---

Example: MMON?
-00.005900

EMON? [<i>i</i>]	<p>Amplified Error Monitor</p> <p>Query the $P \times \epsilon$ voltage, in volts.</p> <p><i>i</i> is an optional parameter that causes streaming of ($P \times \epsilon$) data. If <i>i</i> is specified, then <i>i</i> measurements will be output at a rate of approximately half a second per measurement. If <i>i</i> is specified as 0, then measurements will be output indefinitely. The SOUT command can be used to stop streaming.</p> <p><i>Example:</i> EMON? +00.105537</p>
<hr/>	
OMON? [<i>i</i>]	<p>Output Monitor</p> <p>Query the Output voltage, in volts.</p> <p>OMON? always reports the voltage generated at the front-panel OUTPUT BNC connector (regardless of the state of AMAN).</p> <p><i>i</i> is an optional parameter that causes streaming of Output data. If <i>i</i> is specified, then <i>i</i> measurements will be output at a rate of approximately half a second per measurement. If <i>i</i> is specified as 0, then measurements will be output indefinitely. The SOUT command can be used to stop streaming.</p> <p><i>Example:</i> OMON? +01.106139</p>
<hr/>	
RFMT(?) { <i>z</i> }	<p>Output Streaming Records Format</p> <p>Set (query) the output streaming record format {to $z=(\text{OFF } 0, \text{ON } 1)$}.</p> <p>When ON, data are output on a single line with three comma delimiters. Since there are four monitor channels that can be streamed to output, and any combination of the four may be streamed, the comma delimiters allow unambiguous identification of channel data.</p> <p>The record format is <SMON>,<MMON>,<EMON>,<OMON></p> <p><i>Example:</i> RFMT ON SMON? 3; MMON? 3; EMON? 3; OMON? 3 +00.099909,-00.006053,+00.105601,+01.106135 +00.099909,-00.006031,+00.105615,+01.106123 +00.099915,-00.006001,+00.105636,+01.106151</p>

SOUT [z] Stop Streaming

Turn off streaming (of channel z= (SMN 0, MMN 1, EMN 2, OMN 3)).

If the optional parameter z is not specified, then all streaming outputs are turned off.

Example: SOUT

FPLC(?) {i} Frequency of Power Line Cycle

Set (query) the power line cycle frequency {to i=(50, 60)} Hz.

FPLC is used to program the power-line rejection frequency for the precision voltage monitors (SMON?, MMON?, EMON?, OMON?).

Example: FPLC?
60

3.4.6 Display commands

DISP(?) {z} Select Field

Set (query) the field level to be displayed {to z}. Allowed values of z are

PRP 0	Proportional gain
IGL 1	Integral gain
DER 2	Derivative gain
OFS 3	Output offset
RTE 4	Setpoint rate
STP 5	Setpoint value
MNL 6	Manual output value
ULM 7	Upper limit of output
LLM 8	Lower limit of output
SMN 9	ADC measurement of Setpoint input
MMN 10	ADC measurement of Measure input
EMN 11	ADC measurement of P-Amplified error
OMN 12	ADC measurement of PID/Manual Output

Example: DISP 1

SHFT(?) {z} Shift Status

Set (query) the current shift status {to i=(**OFF 0**, **ON 1**)}.

Example: SHFT?
OFF

DISX(?) {z} Front Panel Display Enable

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

When the display is turned off (DISX OFF), all front panel indicators *and* buttons are disabled.

Example: DISX OFF

3.4.7 Serial communication commands

BAUD(?) {i} Baud Rate

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

At power-on, the baud rate defaults to 9600. Changing baud rate must be carefully orchestrated to ensure proper connectivity throughout the transaction (see the SIM900 manual discussion of the BAUD command for more examples).

Example: BAUD 38800

FLOW(?) {z} Flow Control

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

At power-on, the SIM960 defaults to FLOW RTS flow control.

Example: FLOW 0

PARI(?) {z} Parity

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

At power-on, the SIM960 defaults to PARI NONE.

Example: PARI?
NONE

3.4.8 Status commands

The Status commands query and configure registers associated with status reporting of the SIM960.

***CLS** Clear Status

*CLS immediately clears the ESR, CESR, and the SIM960 status registers.

Example: *CLS

*STB? [i]	<p>Status Byte</p> <p>Reads the Status Byte register [bit i].</p> <p>Execution of the *STB? query (without the optional bit i) always causes the –STATUS signal to be deasserted. Note that *STB? i will <i>not</i> clear –STATUS, even if bit i is the only bit presently causing the –STATUS signal.</p> <p><i>Example:</i> *STB? 16</p>
<hr/>	
*SRE(?) [i,] {j}	<p>Service Request Enable</p> <p>Set (query) the Service Request Enable register [bit i] {to j}.</p> <p><i>Example:</i> *SRE 32; *SRE? 5 1</p>
<hr/>	
*ESR? [i]	<p>Standard Event Status</p> <p>Reads the Standard Event Status Register [bit i].</p> <p>Upon executing *ESR?, the returned bit(s) of the ESR register are cleared.</p> <p><i>Example:</i> GAIN 0 ESR? 16</p> <p>The binary value (16) corresponds to an Execution Error, since GAIN 0 is an illegal value (minimum gain is 0.1).</p>
<hr/>	
*ESE(?) [i,] {j}	<p>Standard Event Status Enable</p> <p>Set (query) the Standard Event Status Enable Register [bit i] {to j}.</p> <p><i>Example:</i> *ESE 16</p>
<hr/>	
CESR? [i]	<p>Comm Error Status</p> <p>Query Comm Error Status Register [for bit i].</p> <p>Upon executing a CESR? query, the returned bit(s) of the CESR register are cleared.</p> <p><i>Example:</i> CESR? 0</p>

CESE(?) [i,]{j}	<p>Comm Error Status Enable</p> <p>Set (query) Comm Error Status Enable Register [bit <i>i</i>] {to <i>j</i>}.</p> <p><i>Example:</i> CESR 0</p>
<hr/>	
INCR? [i]	<p>Instrument condition register</p> <p>Query the instrument condition register [bit <i>i</i>].</p> <p>The values of the bits in the instrument condition register are determined by the current (real-time) condition of the events defined in the instrument status register (see Section 3.5.8).</p> <p>Reading the instrument condition register does not affect the register.</p> <p><i>Example:</i> INCR? 16</p>
<hr/>	
INSR? [i]	<p>Instrument status register</p> <p>Query the instrument status register [bit <i>i</i>].</p> <p><i>Example:</i> INSR? 16</p>
<hr/>	
INSE(?) [i], {j}	<p>Instrument status enable register</p> <p>Set (query) the instrument status enable register [bit <i>i</i>] {to <i>j</i>}.</p> <p><i>Example:</i> INSE 16</p>
<hr/>	
ADSR? [i]	<p>A-to-D status register</p> <p>Query the analog to digital status register [bit <i>i</i>].</p> <p>When new data become available from the A-to-D converter, the A-to-D status register bit corresponding to the channel of the new data is set (see Section 3.5.10).</p> <p><i>Example:</i> ADSR? 15</p>
<hr/>	
ADSE(?) [i], {j}	<p>A-to-D status enable register</p> <p>Set (query) the A-toD status enable register [bit <i>i</i>] {to <i>j</i>}.</p> <p><i>Example:</i> ADSE 2</p>

PSTA(?) {z}	<p>Pulse –STATUS Mode</p> <p>Set (query) the Pulse –STATUS Mode (to z=(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> <p><i>Example:</i> PSTA? OFF</p>
-------------	--

3.4.9 Interface commands

The Interface commands provide control over the interface between the SIM960 and the host computer.

*RST	<p>Reset</p> <p>Reset the SIM960 to its default configuration. The effect of this command is equivalent to the following sequence of commands:</p> <ul style="list-style-type: none"> • DISX ON • DISP PRP • SHFT OFF • GAIN 1.0 • APOL POS • INTG 1.0 • DERV 1.0E-6 • OFST 0.0 • RATE 1.0 • PCTL ON • ICTL OFF • DCTL OFF • OCTL OFF • RAMP OFF • SETP 0.0 (must not precede RAMP OFF) • MOUT 0.0 • ULIM +10.0
------	--

- LLIM -10.0
- INPT EXT
- AMAN PID
- TOKN OFF
- SOUT

The baud rate of the SIM960 is unaffected by *RST. The entire status model is also unaffected by *RST.

Example: *RST

CONS(?) {z}

Console Mode

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

CONS causes each character received at the Input Buffer to be copied to the Output Queue.

At power-on, CONS is set to OFF.

Example: CONS ON

*IDN?

Identify

Read the device identification string.

The identification string is formatted as:

Stanford_Research_Systems,SIM960,s/n*****,ver#.#

where SIM960 is the model number, ***** is the 6-digit serial number, and #.# is the firmware revision level.

Example: *IDN?

Stanford_Research_Systems,SIM960,s/n003173,ver2.15

*TST?

Self Test

There is no internal self-test in the SIM960, so this query always returns 0.

Example: *TST?

0

***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.

Example: *OPC?
 1

WAIT *i* Wait
 Wait *i* milliseconds before processing more commands from the host.
 When using the WAIT command, be careful to not overflow the input buffer of the SIM960 (see section 3.3.2).

Example: SETP 0
 RATE 0.1
 SETP 1.0; WAIT 5000; SMON?
 +00.483159

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	Invalid parameter
17	Missing parameter
18	No change
20	Ramp in progress
21	Limits conflict

Example: *STB? 12; LEXE?; LEXE?
 3
 0

The error (3, “Invalid bit,”) is because *STB? only allows bit-specific queries of 0–7. The second read of LEXE? returns 0.

LCME?

Command Error

Query the last command error code. Valid codes are:

Value	Definition
0	No execution 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

Example: *IDN
LCME?
4

The error (4, "Illegal set") is due to the missing "?".

LBTN?

Last Button

Query the last button that was pressed. The values returned are:

Value	Button
0	no button pressed since last LBTN?
1	[Setpoint]
2	[Output]
3	[Ramp Start/Stop]
4	[Shift]
5	[Select]
6	[On/Off]
7	[▲]/ [←]
8	[▼]/ [→]

Example: LBTN?
5

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 SIM960 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 is set to OFF.

Example: TOKN OFF

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 10 (line feed). The token mnemonic gives the sequence of characters.

At power-on, the default is TERM CRLF.

Example: TERM?
3

3.5 Status Model

The SIM960 status registers follow the hierarchical IEEE–488.2 format. A block diagram of the status register array is given in Figure 3.1.

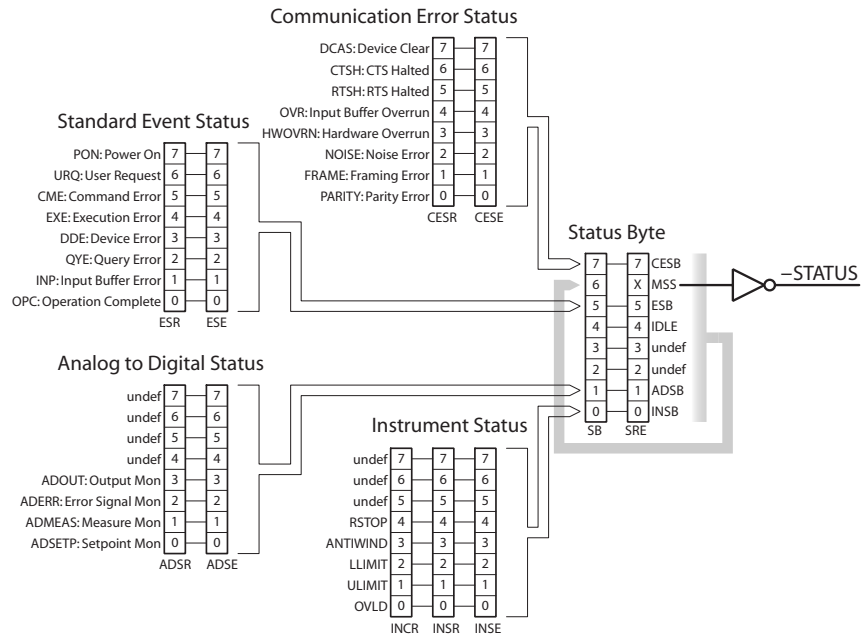


Figure 3.1: Status Register Model for the SIM960 Analog PID Controller.

There are three categories of registers in the SIM960 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. If 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.

3.5.1 Status Byte (SB)

The Status Byte is the top-level summary of the SIM960 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.

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

INSB : Instrument Status Summary Bit. Indicates whether one or more of the enabled flags in the Instrument Status Register has become true.

ADSB : Analog to Digital Status Bit. Indicates whether one or more of the enabled flags in the Analog to Digital 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 SIM960 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.

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

3.5.2 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 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.

This register is cleared at power-on.

3.5.3 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. Undefined for SIM960.

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

3.5.4 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.

3.5.5 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 SIM960. Command Error. Indicates a parser-detected error.

CTSH : Undefined for the SIM960.

DCAS : Device Clear. Indicates the SIM960 received the Device Clear signal (an RS-232 <break>). Clears the Input Buffer and Output Queue, and resets the command parser.

3.5.6 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.

3.5.7 Instrument Status (INCR)

The Instrument Condition Register consists of 5 single-bit monitors of condition events within the SIM960. Bits in the INCR reflect the real-time values of their corresponding signals. Reading the entire register, or individual bits within it, does not affect the value of INCR.

Weight	Bit	Flag
1	0	OVLD
2	1	ULIMIT
4	2	LLIMIT
8	3	ANTIWIND
16	4	RSTOP
32	5	undef (0)
64	6	undef (0)
128	7	undef (0)

OVLD : Amplifier Overload. Set to indicate an overload (either differential *or* common-mode) is presently occurring in the front-end amplifier.

ULIMIT : Upper Limit Reached. Set to indicate the output signal is presently saturated into the programmable upper-limit voltage.

LLIMIT : Lower Limit Reached. Set to indicate the output signal is presently saturated into the programmable lower-limit voltage.

ANTIWIND : Anti-windup Active. Set to indicate the anti-windup circuit is actively inhibiting integration of the error signal.

RSTOP : Ramp Stopped. Set to indicate that no internal setpoint ramp is in progress; cleared to indicate ramping is presently underway.

3.5.8 Instrument Status (INSR)

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

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

3.5.9 Analog to Digital Status Enable (INSE)

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

This register is cleared at power-on.

3.5.10 Analog to Digital Status (ADSR)

The Analog to Digital Status Register consists of 4 event flags; each of which is set by a corresponding conversion completion for one of

the 4 monitored analog signals. Bits in the ADSR are cleared only by reading or with the *CLS command. Reading a single bit (with the ADSR? *i* query) clears only bit *i*.

Weight	Bit	Flag
1	0	ADSETP
2	1	ADMEAS
4	2	ADERR
8	3	ADOUT
16	4	undef (0)
32	5	undef (0)
64	6	undef (0)
128	7	undef (0)

ADSETP : Setpoint Monitor Conversion Complete. Indicates a new conversion result is available for SMON?.

ADMEAS : Measure Monitor Conversion Complete. Indicates a new conversion result is available for MMON?.

ADERR : Amplified Error Monitor Conversion Complete. Indicates a new conversion result is available for EMON?.

ADOUT : Output Monitor Conversion Complete. Indicates a new conversion result is available for OMON?.

While reading this register (with the ADSR? query) will clear any Trip n bit(s) that are set, it will *not* reset the overvoltage protection circuit. To do that, the user must issue the TRIP command. As long as a channel remains tripped off, the Trip n bit will continuously be reasserted.

3.5.11 Analog to Digital Status Enable (ADSE)

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

This register is cleared at power-on.

4 Performance Tests

This chapter describes the tests necessary to verify the SIM960 is operating correctly and within specified calibration.

In This Chapter

4.1	Getting Ready	4-2
4.2	Performance Tests	4-2
4.2.1	Input Amplifier Offset	4-2
4.2.2	A to D Converter test	4-2
4.2.3	Proportional Gain Accuracy	4-3
4.2.4	Derivative Gain Accuracy	4-4
4.2.5	Integral Gain Accuracy	4-5
4.2.6	Ramp Rate Accuracy	4-5
4.2.7	Offset Control Accuracy	4-6
4.2.8	Manual Output Accuracy	4-6
4.3	Calibration	4-6

4.1 Getting Ready

Recommended instruments include:

1. SRS SIM900 Mainframe
2. Agilent 3458A 8 1/2 digit multimeter
3. SRS SR620 Time interval counter
4. SRS SR785 Dynamic signal analyzer
5. PC with remote interface to SIM900 Mainframe

Also needed is a 96:1 resistive divider formed by soldering two resistors, a 210.0 Ω and a 20.00 k Ω resistor (both $\pm 0.1\%$) in series. The resulting three conductors of this network will be labeled for future reference as follows: The conductor running between the resistors will be called the “center”, the conductor on the 210 Ω end will be the “bottom”, and the conductor on the 20 k Ω end will be the “top”. This divider will be used externally to the SIM960 to test the integral gain accuracy.

The SIM960 should be given at least one hour to warm up after power is turned on, and care should be taken not to constrict the ventilation holes in the SIM900 mainframe. It should be located in a room with stable temperature, preferably from 65 to 75 degrees F.

4.2 Performance Tests

The various subsystems of the SIM960 can be tested with the following procedures. In all cases, if the measurement is outside the tolerance or range indicated, then the SIM960 is out of calibration.

4.2.1 Input Amplifier Offset

Ground the two inputs, ‘Setpoint’ and ‘Measure’ of the SIM960 using BNC grounding caps or 50 Ω terminators. Using the remote interface, reset the SIM960 using the *RST command. Select ‘External’ by pressing [Setpoint] on the front panel. Adjust the *P* (gain) parameter to 1000 (maximum gain). Use the Multimeter to measure the Error output at the rear panel BNC of the SIM960. Switch the polarity of the *P* (gain) parameter and observe the change at the rear panel Error BNC. The readings for both polarities should be within ± 10 mV of zero.

4.2.2 A to D Converter test

- Reset the SIM960 using the *RST command.

- Set the P (gain) parameter to +8 using GAIN 8.0.
- Turn off P control using PCTL OFF.
- Set the I parameter to 10^5 using INTG 1.0E5.
- Turn on I control using ICTL ON.
- Use a short BNC cable to connect the SIM960 Output to the Measure input.
- Select “Internal” by pressing [Setpoint] on the front panel.
- Select “Setpoint” and observe the displayed value. It should be within ± 10 mV of 0.000 V.
- Select “Measure”. It should also be within ± 10 mV of 0.000 V.
- Select “Output”. It should also be within ± 10 mV of 0.000 V.
- Set the internal setpoint to +8.000 V by sending SETP +8.0, and repeat the previous three steps, each time observing that the displayed value is within ± 10 mV of +8.000 V.
- Set the internal setpoint to –8.000 V by sending SETP –8.0, and repeating the same three measurements, each time observing that the displayed value is within ± 10 mV of –8.000 V.
- Now disconnect the Output from the Measure input, and ground the Measure input using a BNC grounding cap or 50 Ohm terminator.
- Turn off I control using ICTL OFF, and turn on P control using PCTL ON. The P (gain) parameter should still be 8.0.
- Change the internal setpoint to 0.000 V using SETP 0, and select the $P \times \epsilon$ display level. The value displayed should be within ± 50 mV of 0.000 V.
- Now change the internal setpoint to +1.000 V using SETP +1. The $P \times \epsilon$ value should be within ± 50 mV of +8.000 V.
- Change the internal setpoint to –1.000 V using SETP -1. The $P \times \epsilon$ value should be within ± 50 mV of -8.000 V.

4.2.3 Proportional Gain Accuracy

Reset the SIM960 via the remote interface using *RST. Ground the Measure input. Connect the Source output of the SR785 to the SR785 Channel 1A input, and to the SIM960 Setpoint input. Connect the SIM960 output to the SR785 Channel 2A input. With the SR785 in swept sine mode measure the frequency response at 1 kHz, adjusting

the source output amplitude for each P (gain) setting from the table below.

P (gain)	Source amplitude (volts)
8	0.5
8.1	0.5
16	0.3
16.1	0.3
32	0.15
33	0.15
64	0.08
65	0.08
128	0.04
129	0.04
250	0.02
260	0.02
510	0.01
520	0.01
1000	0.005

In each case the frequency response should be within $\pm 1\%$ of the programmed gain value. The gain should not vary by more than $\pm 1\%$ over the full 100 kHz bandwidth at the P (gain) = 8 setting.

4.2.4 Derivative Gain Accuracy

Use the same connections as for the proportional gain accuracy test. Reset the SIM960 via the remote interface using *RST. Turn off the P (gain) control using PCTL OFF. Turn on the D control using DCTL ON. With the SR785 in swept sine mode and the source output amplitude at 0.5 V, measure the frequency response at the frequencies below for each D setting.

D (sec)	frequency	expected response
1.00×10^{-5}	1.600 kHz	0.10053
1.01×10^{-5}	1.600 kHz	0.10154
1.00×10^{-4}	1.600 kHz	1.0053
1.01×10^{-4}	1.600 kHz	1.0154
1.00×10^{-3}	160 Hz	1.0053
1.01×10^{-3}	160 Hz	1.0154
1.00×10^{-2}	16 Hz	1.0053
1.01×10^{-2}	16 Hz	1.0154
1.00×10^{-1}	1.6 Hz	1.0053
1.01×10^{-1}	1.6 Hz	1.0154

In each case the frequency response should be within $\pm 2\%$ of the programmed gain value.

4.2.5 Integral Gain Accuracy

Use the same connections as for the proportional and derivative gain accuracy tests, but now add the divider network to form a closed-loop configuration.

- Connect the top of the divider to the Output of the SIM960, connect the center of the divider to the Measure input of the SIM960, and connect the bottom of the divider to ground. This ground should be accessed at one of the BNC shields.
- Reset the SIM960 via the remote interface using *RST.
- Set the *P* (gain) parameter to 8.0 using GAIN 8.0.
- Turn off the *P* (gain) control using PCTL OFF.
- Turn on the *I* control using ICTL ON.
- With the SR785 in swept sine mode and the source output amplitude at 0.5 V, measure the frequency response at the frequencies below for each *I* setting.

<i>I</i> (1/sec)	frequency	expected response
5	10 Hz	0.6366
100	150 Hz	0.8488
2×10^3	3.0 kHz	0.8488
5×10^4	100 kHz	0.6366
5×10^5	100 kHz	6.366

In each case the frequency response should be within $\pm 2\%$ of the programmed gain value.

4.2.6 Ramp Rate Accuracy

To test the ramp rate, wire the SIM960 as a 'follower' by connecting the output to the Measure input. Reset the SIM960 using *RST. Set the *P* (gain) to 8.0 using GAIN 8.0, and turn off the *P* control using PCTL OFF. Turn on *I* control using ICTL ON, and set the *I* parameter to 10^5 using INTG 1.0E5.

The Setpoint input should be set to Internal using INPT INT, and ramping should be enabled using RAMP ON. The Multimeter can be used to measure the SIM960 output during a ramp. Set up the multimeter to take 10-20 samples at a known time interval during a ramp, based on the ramp rate being tested, and the range of the ramp. Then extract the ramp rate by calculating the average slope of the ramp data using a least squares fit routine. For each ramp rate being tested, measure both a positive going ramp and a negative one, and calculate the average ramp rate magnitude.

Rate (V/sec)
0.01
0.1
0.101
2.0
2.1
35
36
600
610
10000

In each case the ramp rate magnitude should be within $\pm 2\%$ of the programmed value.

4.2.7 Offset Control Accuracy

- Ground both the Measure and Setpoint inputs of the SIM960 using either BNC grounding caps or $50\ \Omega$ terminators.
- Reset the SIM960 using *RST.
- Turn off *P* control using PCTL OFF.
- Turn on *Offset* control using OCTL ON.
- Change the *Offset* value to 0.000 V, +8.000 V, and –8.000 V, using the OFST command, and observe the 'Output' display at each level.

In each case, the output display value should be within $\pm 5\ \text{mV}$ of the programmed offset.

4.2.8 Manual Output Accuracy

- Reset the SIM960 using *RST.
- Select Manual output using AMAN MAN.
- Use the MOUT command to change the manual output level to 0.000 V, +8.000 V, and –8.000 V, and observe the 'Output' display value for each level.

In each case, the output display value should be within $\pm 5\ \text{mV}$ of the programmed manual output value.

4.3 Calibration

If any of the preceding tests fail, the SIM960 should be returned to the factory for recalibration. Contact Stanford Research Systems or an authorized representative before returning the SIM960.

5 Parts Lists and Schematics

This chapter presents a brief description of the SIM960 circuit design. A complete parts list and circuit schematics are included.

In This Chapter

5.1	Circuit Descriptions	5-2
5.1.1	Microcontroller	5-2
5.1.2	Front Panel Display	5-2
5.1.3	Input Amplifier	5-2
5.1.4	Proportional-Integral-Derivative	5-3
5.1.5	Output Circuitry	5-3
5.2	Parts Lists	5-4
5.2.1	Digital Board & Front Panel	5-5
5.2.2	Analog Board	5-6
5.3	Schematic Diagrams	5-7

5.1 Circuit Descriptions

The SIM960 consists of three separate printed circuit boards: the digital board, the front-panel board, and the analog board. The digital board is directly beneath the left-hand cover (as viewed from the front of the module).

Pages 1–3 of the schematics correspond to the digital board. Page 4 is the front-panel board, and pages 5–9 are the analog board.

5.1.1 Microcontroller

The SIM960 is controlled by microcontroller U103. It is clocked at 10 MHz by the oscillator built around U102, which will track the reference +REF.10MHZ signal (if present) on JP103.

5.1.2 Front Panel Display

The front panel display is illuminated by successively strobing eight digital lines from U301. Each strobe line consists of an NPN emitter-follower (Q301 through Q308) that energizes one seven segment display chip and a set of eight LEDs in parallel. The eight cathodes from the segment display are held high or low at U302 based on a pattern from the controller. Similarly U303 controls the eight lines from the LED cathodes, using NPN open-collectors (Q309 through Q316) as output current buffers.

5.1.3 Input Amplifier

The *Measure* and *Setpoint* single ended inputs of the SIM960 are differenced to form the error signal, ϵ , with a standard three op-amp instrumentation amplifier (U512, U513), operating at $9\times$ gain. R544, R549, R553, & R558 are high-stability resistors (0.1 %, 5 ppm/°C), used to reduce offset drift and to improve common mode rejection (CMRR). R550 is a 10 Ω trimpot for trimming the CMRR.

Polarity control is implemented at U514, which switches between gain $+1\times$ and $-1\times$ with U501A.

Next, U505B is a 12-bit multiplying D-to-A converter (DAC), which together with U504 is used as a vernier attenuator in the error amplifier. Finally, there are three inverting amplifiers in series, each of which may be switched between gain $-1\times$ and some larger gain. The three amplifier gains ($-16\times$, $-4\times$, and $-2\times$) allow the total amplifier gain to be switched by factors of two up to approximately $128\times$. The order of the amplifiers, largest first, is intended to optimize noise referred to input.

5.1.4 Proportional–Integral–Derivative

The resulting amplified error signal $P \times \varepsilon$ (inverted at this point) is then distributed to the three paths, P , I , and D (see schematic page 6). The P path is unchanged, while the other two paths, I and D , each use 12-bit DACs (U606A and U606B) for vernier attenuation of the I and D gains.

Integral action is achieved using a multiplexed feedback capacitor design. U602 is addressed to choose one of four capacitors, all of which are parallel to the fixed feedback capacitor C643, whose small capacitance sets the maximum integral gain value. A similar switching method is employed in the derivative path to choose the input capacitance of the differentiating amplifier U609B. Note R617, which limits the maximum derivative gain to just over +40 dB.

Each of the three signal paths, P , I and D , passes through an analog switch (U601A, U601B, and U601C, respectively) before being combined at the summing junction, the inverting input of op-amp U605B. These switches allow individual signals (P , I or D) to be switched on or off at the summing junction. Manual offset control also enters the summing junction, though no switch is used since zeroing the offset circuit (U610 with U611) is equivalent to switching out the offset signal. Note that R643 is placed in series with the feedback resistor R615 in order to roughly match the switch resistance of each of the switches in U601.

5.1.5 Output Circuitry

U807 is used to switch the output of the SIM960 between Manual and PID modes. The signal then passes through two cascaded diode limiter circuits.

D801 and D802 clamp the output signal with respect to the upper limit voltage generated by U508A, together with U509B and U509C. The effect of cascading two diode limiter circuits is to narrow the clamping range by roughly a factor of two, down to about 100 mV. Comparator U805 switches high when the output signal (going into the limiter circuit at R823) exceeds the upper limit voltage. The inverting input of U805 is referenced to the upper limit voltage through the divider combination R844 and R845, effectively shifting the saturation turn-on location with respect to the clamping “knee”. The lower limit clamp is similarly implemented by D803 and D804.

The output of the limiter circuit is buffered by a composite amplifier consisting of U821 and U822. This arrangement provides the driving capability of the LT1010 without suffering its large input offset voltage, since the output of U822 is servoed to the noninverting input to

U821 via the feedback resistor R819.

U823 is a photo-MOS switch that remains off during power-up until the ± 15 V rails reach about ± 13 V. By then, the output of the Manual DAC, which is driving the SIM960 output during power-up, will have settled to near ground level and may be passed on to the output connector BNC without large start-up transients. Until switch U823 closes, the SIM960 output is referenced to ground via R863 (100k).

U820A provides a buffered analog output of the $P \times \varepsilon$ signal at the back panel of the SIM960. Also, the output of U731A is passed to the back panel to provide an analog output of the internally generated setpoint signal.

5.2 Parts Lists

The parts list for the analog board is separate from the digital & front-panel boards.

5.2.1 Digital Board & Front Panel

Part Reference	SRS P/N	Value	Part Reference	SRS P/N	Value
C102	5-00366-100	18P	R220,R221,R222,R223,R224,	4-01136-110	1.58K
C104	5-00376-100	120P	R225,R226,R227		
C105	5-00368-100	27P	R228	4-01244-110	21.0K
C107,C108,C110	5-00102-030	4.7U	R232	4-01117-110	1.00K
C116,C117,C118,C216,C220	5-00387-100	1000P	R234,R236,R238,R240,R275	4-01146-110	2.00K
C119	5-00345-090	4.0-34P	R243,R248,R259	4-01184-110	4.99K
C201,C202,C203,C204	5-00466-120	.1U	R249,R250,R254,R255	4-01243-110	20.5K
C301,C221	5-00318-110	2.2U/T35	R251,R260,R270,R271,R272,	4-01242-110	20.0K
C222	5-00522-110	47U/T	R273		
C227,C228	5-00375-100	100P	R257	4-01211-110	9.53K
C241,C233	5-00367-100	22P	R262,R263	4-00925-110	10
C240	5-00454-120	.01U	R274	4-01670-121	20K 1% 2PPM
D101,D102,D464,D465,D466,	3-00945-143	BAT54S	R280,R277	4-01280-110	49.9K
D467,D468,D469,D470,D471			R283,R285	4-01288-110	60.4K
D201,D202,D203,D204	3-01430-143	BAS40-05	R284,R287	4-01088-110	499
D205	3-01409-145	BAV99DW	R288,R289	4-01271-110	40.2K
D401-D420,	3-00424-060	GREEN	R301,R302,R303,R304,R305,	4-01459-100	150
D421,D422,D424,D427,D428,			R306,R307,R308		
D429-D443,			R309,R310,R311,R312,R313,	4-01462-100	200
D444,D445,D446,D448,D449,			R314,R315,R316		
D450,D451,D452,D453,D454,			R319,R321,R323,R325,R327,	4-01496-100	5.1K
D455,D456,D457,D458,D459,			R329,R331,R333		
D460,D461,D462			S401,S402,S403,S404,S405,	2-00053-000	B3F-1052
D426,D447,D463	3-00425-060	RED	S406,S407,S408		
JP101	1-00302-010	6 PIN DIF CES	U101	3-00903-124	MAX6348
JP103	1-00367-040	15 PIN D	U102	3-01378-103	74HCU04
JP104	1-00086-002	3 PIN SI	U103	3-01379-114	68HC912B32
J202	1-01014	30 PIN 3x10 F	U104	3-00662-103	74HC14
J301	1-00593-009	HEADER_SIL26	U201,U202	3-01380-120	LF444CM
J401	1-00594-019	HEADER_SIF26	U203,U212	3-00731-120	5534
L102,L103,L105	6-00174-051	BEAD	U204	3-01386-122	DG408
L301	6-00236-130	BEAD	U205	3-01425-170	LTC2415CGN
Q201	3-00927-150	MMBT2907ALT1	U206	3-01383-123	REF02
Q301-Q316	3-01421-150	MMBT2222	U207	3-00663-103	74HC08
R101	4-01495-100	4.7K	U208	3-00116-030	78L05
R102,R106	4-01479-100	1.0K	U209	3-00952-120	OPA2277
R103	4-01431-100	10	U210	3-00727-121	LM339
R105	4-01511-100	22K	U211	3-00724-120	LF353
R107	4-01057-110	237	U213	3-00728-121	LM393
R109	4-01405-110	1.00M	U301	3-01433-103	74HC259
R110,R116,R120,R121,R318,	4-01455-100	100	U302,U303	3-00751-103	74HC574
R320,R322,R324,R326,R328,			U401	3-01424-061	HDSP-A107
R330,R332			U402,U403,U404,U405,U406	3-00290-061	HDSP-A101
R111,R114,R118,R123,R151,	4-01503-100	10K			
R152,R153,R154,R155,R276,			X101-X314 (41 total)	5-00299-100	.1U
R334			Y101	6-00571-020	10.000MHZ
R112,R113,R122	4-01527-100	100K	Y104,Y108,Y202,Y203,Y204,	4-01213-110	10.0K
R115,R117,R119	4-01465-100	270	Y206,Y207,Y208,Y210,Y211,		
R128,R242	4-01309-110	100K	Y229,Y252,Y253,Y256,Y258,		
R201,R205,R209,R212,R233,	4-01405-110	1.00M	Y261,Y282,Y283,Y284,Y285		
R235,R237,R239,R282,R286			PCB, SIM960 Digital Board	7-01258	

5.2.2 Analog Board

Part Reference	SRS P/N	Value	Part Reference	SRS P/N	Value
C502,C504,C506,C507,C536, C537,C538,C541,C608,C637, C638	5-00365-100	15P	R702	4-01050-110	200
C511,C521,C618,C708,C816	5-00369-100	33P	R704	4-01195-110	6.49K
C602,C702,C832	5-00026-100	22P	R708	4-01287-110	59.0K
C603,C616,C712	5-00072-050	10U	R711,R712,R719,R720,R761, R803,R809,R811	4-01088-110	499
C610	5-00025-100	100P	R723,R724,R725,R726,R740, R744,R866	4-00218-000	10.00K
C611	5-00442-120	.001U	R746	4-00011-053	10K
C612	5-00454-120	.01U	R748	4-00014-053	5K
C613	5-00466-120	.1U	R748	4-00014-053	5K
C614	5-00538-050	1.0U	R760	4-01128-110	1.30K
C640	5-00350	.56U	R763,R762	4-01021-110	100
C641	5-00582	.033U	R764	4-01169-110	3.48K
C642	5-00048-050	.0015U	R765	4-01455-100	100
C643	5-00583	100P	R804,R812	4-01551-100	1.0M
C740	5-00059-051	.47U	R808,R810,R814,R815	4-01486-100	2.0K
C741,C742	5-00098-030	10U	R817,R826	4-01111-110	866
C811,C821,C903,C904,C906	5-00102-030	4.7U	R821	4-01038-110	150
C834	5-00377-100	150P	R837,R840,R842	4-00913-000	49.9 FP
D802,D801	3-01430	BAS40-05	R844,R847	4-01142-110	1.82K
D804,D803	3-00901-145	BAS40-06	R863	4-01309-110	100K
D808,D807	3-01487	12V zener	R865,R864	4-00219-000	20.00K
J505,J506,J701,J804,J805	1-00073	Insulated BNC	U501	3-01358-122	DG444
J901	1-01015	30 PIN 3x10 M	U504,U506,U513,U605	3-01361-120	OPA2228
K502	3-00308-203	DS2E-ML2-DC5V	U505,U508,U606	3-01363-171	LTC1590
Q504,Q505,Q802	3-00927-150	MMBT2907A	U509	3-01364-120	OPA4277
R801	3-01421-150	MMBT2222	U514,U512	3-01360-120	OPA228
R502,R507,R542,R547	4-01175-110	4.02K	U601	3-01365-122	DG411DY
R510,R533,R534,R819	4-01146-110	2.00K	U702,U602	3-01366-122	DG333ADW
R527	4-01158-110	2.67K	U603,U604,U807	3-01367-122	DG419DY
R528	4-01149-110	2.15K	U701,U607	3-01398	OPA2131
R530	4-01204-110	8.06K	U608	3-01386-122	DG408DY
R531	4-01262-110	32.4K	U609	3-01361-120	OPA2228
R535,R536,R538,R539,R641, R642,R816,R829	4-01242-110	20.0K	U610	3-01431	LTC1595
R541,R566	4-00012-053	20K	U611,U708,U720,U809	3-00952-120	OPA2277
R544,R549,R553,R558	4-01649-000	1.000K	U703	3-01369-122	DG409/SO
R545,R556,R710,R839,R860	4-01405-110	1.00M	U808,U707	3-01372-171	LTC1596-1
R548	4-01418-100	3	U709	3-01374-103	74HC132A
R550	4-01614-053	10	U710	3-00742-103	74HC74
R551,R617,R709,R713,R721, R862	4-01117-110	1.00K	U711	3-01375-103	74HC86
R557,R554	4-01067-110	301	U712,U802,U803,U805	3-00813-121	LM311M
R559,R563,R564,R565	4-00925-110	10	U823,U730	3-01488	AQY221R2S
R560,R561,R562,R718,R747, R806,R813,R823,R824	4-01184-110	4.99K	U731	3-01387-120	LT1097S8
R602,R614,R615,R619,R628, R845,R846	4-01163-110	3.01K	U732	3-00133	OPA131
R616	4-01317-110	121K	U804	3-00744-103	74HC151
R618,R620,R621,R622,R623, R624,R717,R805	4-01561-100	2.7M	U820	3-01370-120	OPA277UA
R644,R643	4-00997-110	56.2	U821	3-00998-120	OPA227UA
R645,R646,R647,R648	4-01503-100	10K	U822	3-00279-340	LT1010CN8
			U901	3-01432	OPA4131
			U903,U902	3-00787-103	74HC595
			X509-X910 (90 total)	5-00299-100	.1U
			Y537-Y827 (16 total)	4-01213-110	10.0K
			PCB, SIM960 Analog Board	7-01259	

5.3 Schematic Diagrams

Schematic diagrams follow this page.