SR510 Guide to Programming

The SR510 Lock-in Amplifier is remotely programmable via both RS232 and GPIB interfaces. It may be used with laboratory computers or simply with a terminal. All front panel features (except signal input selection and power) may be controlled and read via the computer interfaces. The SR510 can also read the analog outputs of other laboratory instruments using its four general purpose analog input ports. There are also two programmable analog output ports available to provide general purpose control voltages.

Communicating with the SR510

Before using either the RS232 or GPIB interface, the appropriate configuration switches need to be set. There are two banks of 8 switches, **SW1** and **SW2**, located on the rear panel. **SW1** sets the GPIB address and **SW2** sets the RS232 parameters. The configuration switches are read continuously and any changes will be effective immediately. For details on switch settings, see page 6 at the front of this manual.

Command Syntax

Communications with the SR510 are done using ASCII characters. Commands to the SR510 may be in either UPPER or lower case.

A command to the SR510 consists of a single command letter, arguments or parameters if necessary, and an ASCII carriage return (<cr>) or linefeed (<lf>) or both. The different parts of the command do not need to be separated by spaces. If spaces are included, they will be ignored. If more than one parameter is required by a command, the parameters must be separated by a comma. Examples of commands are:

G 5 <cr>	set the sensitivity to 200 nV
T 1,4 <cr>	set the pre filter to 30 mS
F <cr>	read the reference frequency
P 45.10 <cr>	set phase shift to 45.10°
X 5,-1.23E-1 <cr>	set port X5 to -0.123 V

Multiple commands may be sent on a single line. The commands must be separated by a semicolon (;) character. The commands will not be executed until the terminating carriage return is sent.

An example of a multiple command is:

G 5; T 1,4; P 45.10 <cr>

It is not necessary to wait between commands. The SR510 has a command input buffer of 256 characters and processes the commands in the order received. Likewise, the SR510 has an output buffer (for each interface) of 256 characters.

In general, if a command is sent without parameters, it is interpreted as a request to read the status of the associated function or setting. Values returned by the SR510 are sent as a string of ASCII characters terminated usually by carriage return, line feed. For example, after the above command is sent, the following read commands would generate the responses shown below.

Command	Response from the SR510
G <cr>	5<cr><lf>
T 1 <cr>	4<cr><lf>
P <cr>	45.10<cr><lf>

The choice of terminating characters sent by the SR510 is determined by which interface is being used and whether the 'echo' feature is in use. The terminating sequence for the GPIB interface is always <cr><lf> (with EOI). The default sequence for RS232 is <cr> when the echo mode is off, and <cr><lf> when the echo mode is on. The terminating sequence for the RS232 interface may be changed using the J command.

Note that the terminating characters are sent with each value returned by the SR510. Thus, the response to the command string G;T1;P<cr> while using the RS232 non-echo mode would be 5<cr>4<cr>45.10<cr>.

Front Panel Status LED's

The **ACT** LED flashes whenever the SR510 is sending or receiving information over the computer interfaces.

The **ERR** LED flashes whenever an error has occurred, such as an illegal command has been received or a parameter is out of range, or a communication buffer has exceeded 240 characters. This LED flashes for about two seconds on power-up if the battery voltage is insufficient to retain previous instrument settings.

The **REM** LED is on whenever the SR510 is programmed to be in the remote state.

RS232 Echo and No Echo Operation

In order to allow the SR510 to be operated from a

terminal, an echo feature has been included which causes the unit to echo back commands received over the RS232 port. This feature is enabled by setting switch 6 on **SW2** to the DOWN position. In this mode, the SR510 will send line-feeds in addition to carriage returns with each value returned and will also send the prompts 'OK>' and '?>' to indicate that the previous command line was either processed or contained an error. Operating the SR510 from a terminal is an ideal way to learn the commands and responses before attempting to program a computer to control the SR510. When the unit is controlled by a computer, the echo feature should be turned off to prevent the sending of spurious characters which the computer is not expecting.

Try-Out with an ASCII Terminal

Before attempting any detailed programming with the SR510, it is best to try out the commands using a terminal. Connect a terminal with an RS232 port to the RS232 connector on the rear panel of the SR510. Set the baud rate, parity, and stop bits to match the terminal by setting **SW2** per the switch setting table given on page 6. The echo mode should be enabled (switch 6 DOWN). After setting **SW2** and connecting the terminal, hold down the **REL** key while turning the unit on. This causes the SR510 to assume its default settings so that the following discussion will agree with the actual responses of the SR510. The **ACT** and **ERR** LED's on the front panel will flash for a second and the sign-on message will appear on the terminal. Following the message, the prompt 'OK>' will be displayed. This indicates that the SR510 is ready to accept commands.

Type the letter 'P' followed by a carriage return (P<cr>). The SR510 responds by sending to the terminal the characters 0.00 indicating that the phase is set to 0 degrees. In general, a command with no arguments or parameters reads a setting of the unit. To set the phase to 45 degrees, type the command, P45<cr>. To see that the phase did change, use the **SELECT** key on the front panel to display the phase on the **REFERENCE DIGITAL DISPLAY**. Typing the phase read command, P<cr>, will now return the string 45.00 to the terminal.

Now read the gain using the sensitivity read command, G<cr>. The response should be 24 meaning that the sensitivity is at the 24th setting or 500 mV. Change the sensitivity by typing G19<cr>. The sensitivity should now be 10 mV. Check the front panel to make sure this is so.

The output of the lock-in is read by typing the command, Q<cr>. The response is a signed floating point number with up to 5 significant digits plus a signed exponent. Change the gain to 10 uV using the G10 command. The response to the Q command now will be similar to the previous one except that the exponent is different.

Attach a DC voltmeter to the **X6** output on the rear panel. The range should allow for 10V readings. The voltage at the **X6** output can be set using the **X6** command. Type X6,5.0<cr> and the **X6** output will change to 5.0V. To read this back to the terminal, just type **X6**<cr>. When setting the **X6** voltage, the voltage may be sent as an integer (5), real (5.000), or floating point (0.500E1) number. Now connect the **X6** output to the **X1** input (also on the rear panel). **X1** through **X4** are analog input ports. To read the voltage on **X1**, simply type X1<cr>. The response 5.000 should appear on the terminal. The analog ports **X1** through **X6** can be used by your computer to read outputs of other instruments as well as to control other laboratory parameters.

At this point, the user should experiment with a few of the commands. A detailed command list follows.

SR510 Command List

The first letter in each command sequence is the command. The rest of the sequence consists of parameters. Multiple parameters are separated by a comma. Those parameters shown in {} are optional while those without {} are required. Variables m and n represent integer parameters while v represents a real number. Parameters m and n must be expressed in integer format while v may be in integer, real, or floating point format.

A {n}
If n is "1", the A command causes the **auto offset** routine to run. Every time an "A 1" command is received, the auto offset function is executed. If n is "0", then the auto offset is turned off. If n is absent, then the auto offset status is returned. Note that if the manual offset is on, an "A 1" command will turn off the manual offset before executing the auto offset function.

B {n}
If n is "1", the B command sets the **bandpass** filter in. If n is "0", the bandpass filter is taken out. If n is absent, then the bandpass filter status is returned.

C {n}
If n is "1", the C command sets the **reference LCD display** to show the phase setting. If n is "0", the LCD will display the reference frequency. If n is absent, the parameter being displayed (frequency or phase) is returned. Note that the P and F commands are used to read the actual values of the phase and frequency.

D {n}
If n is included, the D command sets the **dynamic reserve**. If n is absent, the dynamic reserve setting is returned.

n	Dyn Res
0	LOW
1	NORM
2	HIGH

Note that not all dynamic reserve settings are allowed at every sensitivity.

E {n}
If n is "1", the E command turns the output **expand** on. If n is "0", the expand is turned off. If n is absent, the expand status is returned.

F
The F command reads the reference **frequency**. For example, if the reference frequency is 100 Hz, the F command returns the string "100.0". If the reference frequency is 100.0 kHz, the string "100.0E+3" is returned. The F command is a read only command.

G {n}
If n is included, the G command sets the **gain** (sensitivity). If n is absent, the gain setting is returned.

n	Sensitivity
1	10 nV
2	20 nV
3	50 nV
4	100 nV
5	200 nV
6	500 nV
7	1 μV
8	2 μV
9	5 μV
10	10 μV
11	20 μV
12	50 μV
13	100 μV
14	200 μV
15	500 μV
16	1 mV
17	2 mV
18	5 mV
19	10 mV
20	20 mV
21	50 mV
22	100 mV
23	200 mV
24	500 mV

Note that sensitivity settings below 100 nV are allowed only when a pre-amplifier is connected.

H
The H command reads the **pre-amplifier status**. If a pre-amplifier is connected, a "1" is returned, otherwise, a "0" is returned. The H command is a read only command.

I {n}
If n is included, the I command sets the **remote-local** status. If n is absent, the remote-local status is returned.

n	Status
0	Local: all front panel keys are operative
1	Remote: front panel keys are not operative. The display up key returns the status to local.
2	Lock-out: front panel keys are not operative. No key returns the status to local. Another I command is needed to return to local.

When using the GPIB interface, the REN, LLO, and GTL commands are not implemented. The I command is used by both interfaces to set the remote-local status.

J {n1,n2,n3,n4}
The J command sets the RS232 **end-of-record** characters sent by the SR510 to those specified by the ASCII codes n1-n4. If no argument is included, the end-of-record sequence returns to the default (a carriage return), otherwise, up to four characters may be specified. The end-of-record required by the SR510 when receiving commands is not affected.

K n
The K command simulates a front panel **key press**. The effect is exactly the same as pressing the selected key once. The parameter n is required.

n	Key
1	Post Time Constant Up
2	Post Time Constant Down
3	Pre Time Constant Up
4	Pre Time Constant Down
5	Offset Up
6	Offset Down
7	Zero Phase (Simultaneous 90° Up and Down)
8	Line Notch Filter
9	Bandpass Filter
10	Line X 2 Notch Filter
11	Relative (Auto Offset)
12	Offset (On/Off)
13	Expand
14	Local (Display Up when REMOTE)
15	Reference Trigger Mode
16	Reference Mode (f/2f)
17	Degrees Up
18	Degrees Down
19	Quad Up
20	Quad Down
21	Select Display (f/phase)
22	Sensitivity Up
23	Sensitivity Down
24	Dyn Res Up
25	Dyn Res Down
26	Display Up
27	Display Down

L m {,n}
The L command sets and reads the status of the line **notch filters**. If m is "1", then the 1X line notch is selected, if m is "2", the 2X line notch is selected. The parameter m is required. If n is "1", the L command sets the selected filter in. If n is "0", the selected filter is taken out. If n is absent, the status of the selected filter is returned.

M {n}
If n is "1", the M command sets the **reference mode** to 2f. If n is "0", the reference mode is set to f. If n is absent, the reference mode is returned.

N {m}
If m is "1", the N command sets the **ENBW** to 10 Hz. If m is "0", the ENBW is set to 1 Hz. If m is absent, the ENBW setting is returned.

O {n} {,v}
If n is "1", the O command turns the **offset** on. If n is "0", the offset is turned off. If n is absent, the offset status (on or off) is returned. (The value of the offset is read using the S and Q commands.) If n is included, then v may also be sent. v is the offset value up to plus or minus full scale in units of volts. For example, to offset half of full scale on the 100 µV sensitivity, v should be "50.0E-6" or an equivalent value. However, if the sensitivity is then changed to 200 µV, the offset is now half of the new full scale or 100 µV. When the sensitivity is changed, the offset is retained as a constant fraction of full scale rather than as a voltage referred to the input. The expand function will, on the other hand, preserve the value of the offset as an input referred voltage. Once a value of v is sent, the offset may be turned off and on without losing the offset value by using the O command without the v parameter. Note that if the auto offset is on, an "O 1" command will turn the auto offset off and turn the manual offset on without changing the actual offset value.

P {v}
If v is absent, the P command returns the **phase** setting from -180 to +180 degrees. When v is included, the phase is set to the value of v up to ±999 degrees.

Q
The Q command returns the **output** reading in units of volts. For an input signal of 50 µV on a full scale sensitivity of 100 µV, the Q command will return the string "50.00E-6". The parameter read is the same as that being shown on the output display and can be changed with the S command.

R {n}
If n is included, the R command sets the **reference input trigger** mode. If n is absent, the trigger mode is returned.

n	Mode
0	Positive
1	Symmetric
2	Negative

S {n}
If n is included, the S command selects the parameter shown on the analog meter and output digital **display** as well as the output BNC. If n is absent, the parameter being displayed is returned.

n	Display
0	X
1	Offset
2	Noise

T m {,n}
The T command sets and reads the status of the **time constants**. If m is "1", the pre time constant is selected, if m is "2", the post time constant is selected. The parameter m is required. If n is included, the T command sets the selected time constant. If n is absent, the setting of the selected time constant is returned.

n	Pre Time Constant (m=1)
1	1 mS
2	3 mS
3	10 mS
4	30 mS
5	100 mS
6	300 mS
7	1 S
8	3 S
9	10 S
10	30 S
11	100 S

n	Post Time Constant (m=2)
0	none
1	0.1 S
2	1 S

U m {,n}
The U command sets and reads the unit's ROM **calibration** bytes. m is the address offset of the byte, 0-255. If n is absent, the value of the addressed calibration byte is returned. If n is included, the addressed calibration byte is set to the value of n, 0-255. The new value will be in effect until the power is turned off or a reset command is issued. Use of this command is not recommended.

V {n}
If n is included, the V command sets the GPIB **SRQ** (service request) mask to the value n. If n is absent, the value of the SRQ mask is returned.

W {n}
The W command sets and reads the RS232 character **wait** interval. If n is included, the SR510 will wait n*4 mS between characters sent over the RS232 interface. This allows slow computer interfaces to keep up. n can range from 0 to 255. If n is absent, the wait value is returned. The wait interval is set to 6 on power-up.

X n {,v}
n designates one of the 6 general purpose **analog ports** located on the rear panel. If n is 1,2,3, or 4, the X command will return the voltage on the designated analog input port (X1-X4) in volts. If n is 5 or 6, then v may also be sent. If v is included, the designated analog output port (X5 or X6) will be set to v volts where v has the range -10.24V to +10.24V. If v is absent, the output value of the selected port is returned. On power-up, port X5 is the ratio output. An "X 5" command will read the ratio output. An "X 5" command with the parameter v will set port X5 to v volts, overriding the ratio output. Port X5 will return to the ratio output on power-up or reset.

Y {n}
The Y command reads the **status** byte. (See below for a definition of the Status Byte.) n designates one bit, 0-7, of the status byte. If n is included, the designated bit of the status byte is returned. The bit which is read is then reset. If n is absent, the value of the entire byte is returned and all status bits are then reset. This status byte may also be read over the GPIB using the serial poll command.

Z
The Z command causes an internal **reset**. All settings return to their default values. The **ERR** LED will be on for about 2 seconds to indicate that the stored instrument settings are being ignored. If the RS232 echo mode is on, the sign-on message is sent over the RS232 interface.

Status Byte

The SR510 maintains an 8-bit status register which the user may read to obtain information on the unit's status. The status byte may be read in two ways: by sending the Y command, which returns the value of the byte in ASCII coded decimal, or, when using the GPIB, by performing a serial poll. The returned status byte reflects all of the status conditions which have occurred since the last time the byte was read. After the status byte has been read, it is cleared. Thus, the status byte should be read initially to clear all previous conditions (especially after a power up or after settings have been changed).

The definitions for each bit of the status byte are given below:

Bit 0
Busy. When this bit is set, it indicates the SR510 has unprocessed commands pending on its command queue. For RS232 communications, this bit is always high since the Y command itself will be an unprocessed command. This bit is not reset when read but only when there are no pending commands. Since the SR510 buffers incoming commands, it is not necessary to read this bit before sending each command. Commands received while the SR510 is executing a previous command are stored until all previously received commands have been executed.

Bit 1
Command Parameter Out of Range. This bit is set if a parameter associated with a command is not in the allowed range.

Bit 2
No Reference. This bit is set when no reference input is detected, either because the amplitude is too low or the frequency is out of range.

Bit 3
Unlock. This bit is set when the reference oscillator is not locked to the reference input. If there is no reference input, bit 2 (no reference) will be set but bit 3 (unlock) may not be.

Bit 4
Overload. This bit is set if there is a signal overload. This can happen when the sensitivity is too high, the dynamic reserve is too low, the offset is on, or the expand is on. Overloads on the general purpose A/D inputs or the ratio output are not detected.

Bit 5
Auto Offset Out of Range. This bit is set if the auto offset function cannot zero the output because the output exceeded 1.024X full scale.

Bit 6
SRQ. This bit is high if the SR510 has generated an SRQ on the GPIB interface. This bit is reset after the SR510 has been serial polled. This bit is set only for status reads via a serial poll, ie., Bit 6 always zero for the RS232.

Bit 7
Command Error. This bit is set when an illegal command string is received.

Errors

Whenever a 'parameter out of range' or an 'unrecognized command' error occurs, the appropriate status bits are set and the **ERR** LED flashes. In addition, any commands remaining on the current command line (up to the next <cr>) are lost. The **ERR** LED will also light if any of the internal communication buffers overflows. This occurs when 240 characters are pending on the command queue or output queue. The **ERR** LED will go off as soon as all buffers drop below 200 characters again.

Reset

The Z command resets the unit to its default state. The default front panel settings are listed in the **DEFAULTS** section of the **Guide to Operations**. In addition, the interface status returns to LOCAL, the SRQ mask is cleared, the RS232 character WAIT interval is set to 6, and the terminating sequence is reset to the proper defaults.

The command and output buffers are cleared by the Z command. Therefore, it is bad practice to use the Z command before all previous commands have been processed and all responses have been received.

Trouble-Shooting Interface Problems

If you are having difficulty getting your computer to communicate with the SR510 look to the sections on the RS232 and GPIB interfaces for some tips specific to your particular interface.

An ASCII terminal is a valuable aid for debugging interface problems. You can use it to:

1) become familiar with the SR510's command structure,

2) see GPIB bus transactions by using the GPIB echo mode,

3) eavesdrop on transactions when using the RS232 interface,

4) substitute a human for the SR510 by using a null modem cable (to make the DTE a DCE) and attaching the terminal to the port to which you would normally have connected the SR510. This allows you to test your program's responses to inputs which you provide from the terminal.

Common Hardware Problems include:

1) The RS232 or GPIB cables are not properly attached.

2) The configuration switches for the RS232 characteristics or GPIB address are not set correctly (Make sure the RS232 echo is off when using the RS232 interface with a computer. The GPIB with RS232 echo mode should be off when not debugging the GPIB interface.)

3) Your computer requires an RS232 control line to be asserted, but your cable does not pass it between the SR510 and the computer, or, your computer is not asserting the DTR line on the RS232.

Common Software Problems include:

1) You have sent the wrong command to ask for data from the SR510. Your program will wait forever for a response which is not going to come. This may not be your fault; we have seen Microsoft's Interpreted Basic on the IBM PC occasionally send a curly bracket (ASCII 253) when it was supposed to have sent a carriage return (ASCII 13).

2) Your computer's baud rate has been changed and no longer matches the SR510's baud rate.

3) The initial command sent to the SR510 was invalid due to a garbage character left in the command queue from power-up, or, the first character in you computer's UART is garbage, also due to power-up. It is good practice to send a few carriage returns to the SR510 when your program begins, and have your program clear-out its UART at the start of your program.

4) The SR510 is not sending the correct 'end-of-record' marker for your computer. For example, it appears that Microsoft's Rev 3.2 FORTRAN on the IBM PC under DOS 2.1 requires two carriage returns for an end-of-record marker. The J command can be used to set the SR510 end-of-record marker to 2 carriage returns. [The end-of-record marker is that sequence which indicates that the response is complete. From the keyboard, a single carriage return is the end-of-record marker.]

5) Answers are coming back from the SR510 too fast, overwriting the end-of-record markers, and causing the computer to hang waiting for a complete response. In this case, the W command can be used to slow down the response time of the SR510 preventing overwriting.

6) Answers are coming back from tthe SR510 too slowly due to the W6 default setting for the character interval time. Use the W command to speed up the transmission from the SR510. This can cause problems for the GPIB interface if the echo mode is on (switch 6 of SW21).

The SR510 with the RS232 Interface

The RS232 is a popular serial interface standard for bit serial communication. Despite the existence of the standard there are many permutations of control lines, baud rates, and data formats. If you do not have a lot of experience interfacing RS232 equipment you should read Appendix B for a description of the RS232 and interfacing tips.

Data Communications Equipment (DCE)

The SR510 is configured as DCE so that it may be connected directly to a terminal. If the SR510 is to be interfaced with another DCE device, a special cable (sometimes referred to as a 'modem' cable) is required. To use the RS232 interface you must set the switches in **SW2** to match your computer's baud rate, parity, and number of stop bits. Refer to Page 6 for details.

Wait Command

The SR510 normally waits until the RS232 'Clear to Send' control line (CTS) is asserted before sending characters. However, some computers do not set and reset the CTS line, possibly causing the SR510 to send data when the computer is not ready to read it. The SR510 may be 'slowed down' using the W command. Sending

'Wn' causes the unit to wait nX4 mS before sending each character over the RS232 bus. The command W0 sets the wait interval to zero and results in the fastest transmission. The wait interval is set to 6 (24 mS) on power-up.

Termination Sequences

The default RS232 termination characters are sufficient to interface with most computers, however, it will occasionally be necessary to send special terminating sequences to fit the requirements of some computers. This can be done with the J command. The format for the command is:

J {n1,n2,n3,n4}

where n1, n2, n3, and n4 are decimal values between 0 and 255 corresponding to the decimal ASCII codes of the desired termination characters. For instance, if the desired termination sequence is an asterisk, (ASCII 42), two carriage returns, (ASCII 13), and a line feed, (ASCII 10), the appropriate command is:

J 42,13,13,10

If a G command is sent requiring an answer of 24 (sensitivity = 500 mV), the SR510 would respond with the string

24*<cr><cr><lf>

Up to four terminating characters may be specified by the J command. If no arguments are sent with the J command, the terminating sequence returns to the default (echo on: <cr><lf>; echo off: <cr>).

The J command does not affect the terminating character (<cr>) required at the end of commands received by the SR510. It also does not affect the terminating sequence sent with data over the GPIB interface.

The SR510 with the GPIB Interface

For a brief introduction to the GPIB standard, please read Appendix C at the back of this manual. Before using the GPIB interface you must set the switches in **SW1** per the instructions on page 6.

GPIB Capabilities

The GPIB capabilities of the SR510 consistent with IEEE standard 488 (1978) are shown in the table below. Also shown are the responses of the SR510 to some standard commands.

Code	Function
SH1	Source handshake capability
AH1	Acceptor handshake capability
T5	Basic Talker, Serial Poll, Unaddressed to talk if addressed to listen
L4	Basic Listener, Unaddressed to listen if addressed to talk
SR1	Service request capability
PP0	No parallel poll capability
DC1	Device Clear capability
RL0	REN,LLO, GTL not implemented. 'I' command sets Remote-Local.

SR510 Response to GPIB Commands

Mnemonic	Command	Response
DCL	Device Clear	Same as Z command
SDC	Selected Device Clear	Same as Z command
SPE	Serial Poll Enable	Send Status Byte, & clear status byte

Because the SR510 can be controlled by an RS232 interface as well as the GPIB, the remote-local functions are not standard. There is no local with lock out state. When in the local state, remote commands are processed, even without the REN command being issued. This is because the RS232 interface has no provision for bus commands and remote commands over the RS232 interface would never be enabled.

Serial Polls and Service Requests

The status byte sent by the SR510 when it is serial polled is the same status byte which is read using the Y command (except for bit 6, SRQ). Of course, when the SR510 is serial polled, it does not encode the status byte as a decimal number. The SR510 can be programmed to generate a service request (SRQ) to the GPIB controller every time a given status condition occurs. This is done using the V{n} command. The mask byte, n (0-255), is the SRQ mask byte. The mask byte is always logically anded with the status byte. If the result is non-zero, the SR510 generates an SRQ and leaves the status byte unchanged until the controller performs a serial poll to determine the cause of the service request. When the unit has been serial polled, the status byte is reset to reflect all of the status conditions which have occurred since the SRQ was generated.

For example, if we want to generate an SRQ whenever there is an overload or unlock condition, we need an SRQ mask byte equal to 00011000 binary, or 24 decimal ("V24" command). The byte 00011000 binary corresponds to the status byte

with the 'no reference' and 'unlock' status bits set. If an overload occurs, then an SRQ will be generated. The serial poll will return a status byte showing SRQ and overload. If an unlock condition occurs before the serial poll is concluded, another SRQ will be generated as soon as the serial poll is finished. A second serial poll will reflect the unlock condition.

Any SRQ generated by the 'no reference, 'unlock', 'overload', and 'auto over-range' conditions will also reset the corresponding bit in the SRQ mask byte. This is to prevent a constant error condition (such as no reference applied to the input) from continually interrupting the controller. When such an SRQ occurs, the controller should change some parameter so as to solve the problem, and then re-enable the SRQ mask bit again using the V command.

GPIB with RS232 Echo Mode

It is sometimes useful when debugging a GPIB system to have some way of monitoring exactly what is going back and forth over the bus. The SR510 has the capability to echo all characters sent and received over the GPIB to its RS232 port. This mode of operation is enabled by setting switch 6 of **SW1** to the DOWN position. The baud rate, stop bits, and parity of the RS232 port are still set by **SW2**. Of course, the RS232 port operates at much lower speeds than the GPIB and will slow down the GPIB data rate in this mode. (Use the W0 command to allow the RS232 interface to run at full speed, otherwise, the GPIB transactions may take so long that the controller can hang.) During actual use, this mode should be disabled.

The SR510 with BOTH Interfaces

If both interfaces are connected, commands may be received from either interface. Responses are always sent to the source of the request (except in GPIB echo mode). It is unwise to send commands from the two interfaces at the same time since the characters from different sources can become interleaved on the command queue and result in 'unrecognized command' errors.