

Communications Examples for SIMs

Introduction

The Small Instrumentation Module family was designed to support simple communications between a user's computer and a collection of instruments. While it is possible to communicate directly with a SIM module, this note will only consider the case of communication through a SIM900 Mainframe.

The SIM900 has two host computer interfaces, RS-232 and GPIB, one of which is active. To switch interfaces, use the rear-panel piano-style DIP switch (position 3): up is for RS-232, down is for GPIB. The 5 right-most switches are interpreted based on the host selection, and determine either default baud rate (RS-232) or instrument address (GPIB). Only one host interface can be active, and the selection is determined at power-on time for the SIM900.

The simplest style of communication through the SIM900 is the "connection" model, where a single bidirectional I/O stream is managed. Upon power-on, this stream is initially directed to the SIM900 itself, so that, for example, an identification query will result in the SIM900 ID string as a response back to the host computer. Using the SIM900 "CONN" command, the user can steer the I/O stream to one of the instrument ports of the Mainframe (1-8 for the internal slots, 9 for the remote SIM port, and A or B for the auxiliary RS-232 ports). After CONNECTing to a particular port, all I/O is directed from the host computer through the Mainframe to the target port, and responses from the target port are passed through the Mainframe back to the host computer. To end the CONNECTION, the host computer sends a preprogrammed "escape" key. The escape key is provided as the second parameter in the CONN command, and should be chosen carefully to ensure that it never inadvertently occurs within the normal stream of I/O from the host computer to the target. In the examples below, the nonsense sequence XYZZY is chosen as an escape key (note the key is case-sensitive; the second example actually uses xyzzy to demonstrate this).

When the Mainframe is relaying data from the host computer through to a CONNECTed port, it continually scans for a possible match with the escape key. If the first character of the key is received, the character is held in a memory buffer of the Mainframe. When the next character is received, it is compared with the second character of the key; if it matches, it too is added to the buffer, otherwise both the buffered previous character and the new character are transmitted to the target port. As an example, consider a connection with escape key XYZZY. If the host attempts to transmit the message XYZ to the port, all three characters will be held back by the Mainframe until another character is received (to make sure it is not the next Z of escape key). There is *no* timeout on this partial-match buffering, so the data will be held back indefinitely until the Mainframe can uniquely disambiguate between the message data and the escape key.

Things get more interesting when multiple SIM900 mainframes are connected together using the auxiliary RS-232 ports. In this case, different escape keys must be used for the two Mainframes, to enable redirection of the "child" SIM900 I/O stream without disconnecting the "parent" stream. See Example 3 below for a demonstration of this.

Example 1: RS-232

Text typed by the user is set “**bold teletype**”, while responses are set as “plain teletype”.

description	session
<i>identify the Mainframe</i>	*IDN? ↵
<i>response from Mainframe</i>	Stanford_Research_Systems,SIM900,s/n000112,ver2.4
<i>connect to port 6</i>	conn 6,'XYZZY' ↵
<i>identify the module @ port 6</i>	*idn? ↵
<i>response from port 6 (SIM970)</i>	Stanford_Research_Systems,SIM970,s/n000001,ver2.0
<i>query channel 1 voltage</i>	volt?1 ↵
<i>response (13.3 μV)</i>	0.0000133
<i>query channel 2 voltage</i>	volt?2 ↵
<i>response (-18.2 μV)</i>	-0.0000182
<i>disconnect back to Mainframe</i>	XYZZY
<i>check identity string</i>	*idn? ↵
<i>response from Mainframe</i>	Stanford_Research_Systems,SIM900,s/n000112,ver2.4

Example 2: GPIB

The GPIB (IEEE-488) example is written as pseudo-code, since there is no analogous console program for GPIB. Actual data sent to the SIM system is again set as “**bold teletype**”, while responses are set as “plain teletype”.

description	session
<i>connect to SIM900 via GPIB</i>	GPIB_Open(address=2)
<i>reset SIM900 interface</i>	GPIB_Device_Clear()
<i>identify the Mainframe</i>	GPIB_Write(“*IDN?”)
<i>response from Mainframe</i>	GPIB_Read(); response: Stanford_Research_Systems,SIM900,s/n000112,ver2.4
<i>convert LF to EOI</i>	GPIB_Write(“CEOI ON”)
<i>convert EOI to LF</i>	GPIB_Write(“EOIX ON”)
<i>connect to port 6</i>	GPIB_Write(“conn 6,'xyZZy'”)
<i>set module to LF termination</i>	GPIB_Write(“TERM LF”)
<i>identify the module @ port 6</i>	GPIB_Write(“*IDN?”)
<i>response from port 6 (SIM970)</i>	GPIB_Read(); response: Stanford_Research_Systems,SIM970,s/n000001,ver2.0
<i>query channel 2 voltage</i>	GPIB_Write(“VOLT?2”)
<i>response (-18.2 μV)</i>	GPIB_Read(); response: -0.0000182
<i>disconnect back to Mainframe</i>	GPIB_Write(“xyZZy”)
<i>check identity string</i>	GPIB_Write(“*IDN?”)
<i>response from Mainframe</i>	GPIB_Read(); response: Stanford_Research_Systems,SIM900,s/n000112,ver2.4

Example 3: Multiple Mainframes

In the following (RS-232) example, the host computer is connected to SIM900 serial number 000112. The internal slots of that mainframe are filled with SIM970 Voltmeters, and Auxiliary port A is connected to a second SIM900 Mainframe (serial number 000321). This second Mainframe is filled with SIM925 Multiplexers. Actual data sent to the SIM system is again set as “**bold teletype**”, while responses are set as “plain teletype”.

description	session
<i>identify the Mainframe</i>	*IDN? ↵
<i>response from Mainframe</i>	Stanford_Research_Systems,SIM900,s/n000112,ver2.4
<i>connect to port 4</i>	conn 4,'!1XYZ' ↵
<i>identify the module @ port 4</i>	*idn? ↵
<i>response from port4 (SIM970)</i>	Stanford_Research_Systems,SIM970,s/n000001,ver2.0
<i>query channel 1 voltage</i>	volt?1 ↵
<i>response (13.3 μV)</i>	0.0000133
<i>disconnect back to Mainframe</i>	!1XYZ
<i>check identity string</i>	*idn? ↵
<i>response from Mainframe</i>	Stanford_Research_Systems,SIM900,s/n000112,ver2.4
<i>connect to port A</i>	conn A,'!1XYZ'
<i>identify the instrument @ port A</i>	*idn? ↵
<i>response from port A</i>	Stanford_Research_Systems,SIM900,s/n000321,ver2.4
<i>connect to port 3 of 2nd Mainframe</i>	conn 3,'!2XYZ' ↵
<i>identify the module @ port 3</i>	*idn? ↵
<i>resp. from port 3 (2nd Mainframe)</i>	Stanford_Research_Systems,SIM925,s/n003456,ver1.3
<i>disconnect back to 2nd Mainframe</i>	!2XYZ
<i>disconnect back 1st Mainframe</i>	!1XYZ
<i>check identity string</i>	*idn? ↵
<i>response from Mainframe</i>	Stanford_Research_Systems,SIM900,s/n000112,ver2.4