Agilent E5070B/E5071B ENA Series RF Network Analyzers

Controlling Using SICL-LAN Server

First Edition

Agilent Technologies

No. 16000-95022

August 2002

Notices

The information contained in this document is subject to change without notice.

Agilent Technologies Japan, Ltd.

Component Test PGU-Kobe

1-3-2, Murotani, Nishi-ku, Kobe, Hyogo, 651-2241 Japan

Sample Program

The customer shall have the personal, non-transferable rights to use, copy, or modify SAMPLE PROGRAMS in this manual for the customer’s internal operations. The customer shall use the SAMPLE PROGRAMS solely and exclusively for their own purposes and shall not license, lease, market, or distribute the SAMPLE PROGRAMS or modification of any part thereof.

Agilent Technologies shall not be liable for the quality, performance, or behavior of the SAMPLE PROGRAMS. Agilent Technologies especially disclaims any responsibility for the operation of the SAMPLE PROGRAMS to be uninterrupted or error-free. The SAMPLE PROGRAMS are provided AS IS.

AGILENT TECHNOLOGIES DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Agilent Technologies shall not be liable for any infringement of any patent, trademark, copyright, or other proprietary right by the SAMPLE PROGRAMS or their use. Agilent Technologies does not warrant that the SAMPLE PROGRAMS are free from infringements of such rights of third parties. However, Agilent Technologies will not knowingly infringe or deliver software that infringes the patent, trademark, copyright, or other proprietary right of a third party.

Controlling Using SICL-LAN Server

This section describes how to control the E5070B/E5071B using SICL in the Windows environment, with a sample program (a VBA macro for Microsoft Excel) written in Visual Basic. The source file of this program, ctrl_lansicl.xls, is available in the sample program disk.

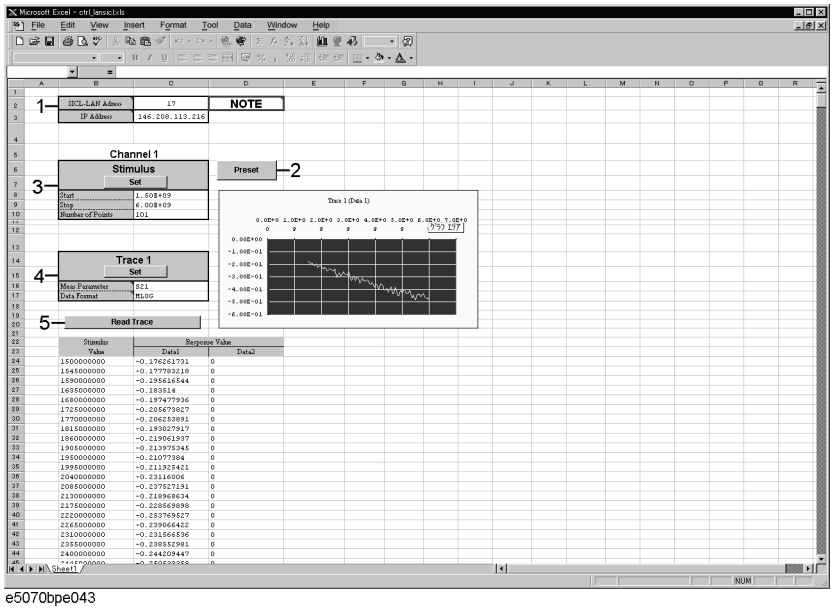
NOTE

To control the E5070B/E5071B using the SICL-LAN server, you need to make preparations described in the chapter2 “Control over SICL-LAN server” in the E5070B/E5071B Programmer’s Guide.

Using VBA macro

Opening ctrl_lansicl.xls in Microsoft Excel display a screen as shown in Figure 1-1.

Figure 1-1 **ctrl_lansicl.xls**



e5070bpe043

For how to use each element in Figure 1-1, refer to the following description.

We begin describing the part 1. In the cell to the right of SICL-LAN Address, enter the address of the E5070B/E5071B for control with the SICL-LAN server. This address is **xx**, which has been set with the command **[System] - Misc Setup - Network Setup - SICL-LAN Address [xx]**. Enter the IP address of the E5070B/E5071B in the cell right side of IP Address. This VBA macro will not work properly without appropriate values in the two cells.

Clicking the button labeled as Preset in the part 2 executes the presetting operation.

In the part 3, sweep range (start and stop points) and number of measurement points for channel 1 are set. Clicking the button labeled as Set executes the setting as shown in the setting table.

The part 4 set the measurement parameters and data format for trace 1 in channel 1.

Clicking the button labeled as Set executes the setting as shown in the setting table.

Clicking the button labeled as Read Trace in the part 5 retrieves the formatted data array of trace 1 in channel 1 and displays it in tabular and graphical formats.

Description of operation in VBA macro

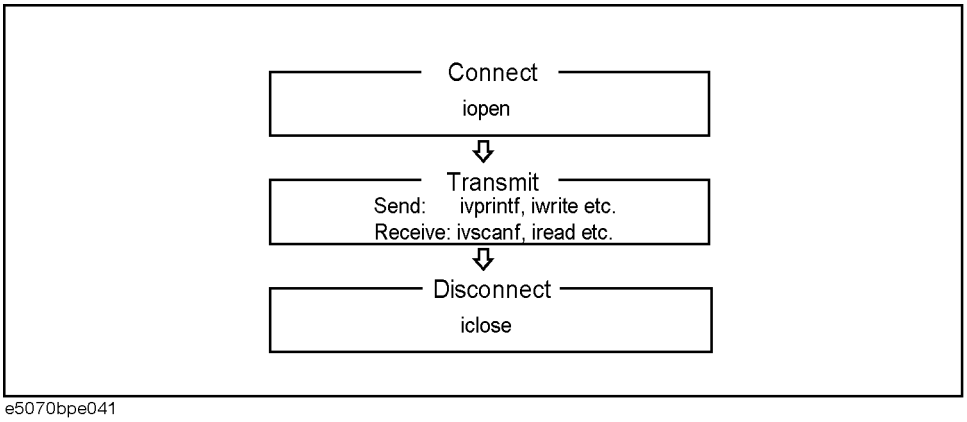
Here described is operation of the VBA macro, focusing on the part related to controlling with SICL.

In order to use SICL in your VBA macro, you must declare functions and define variables with a definition file of SICL (for VB). In the VBA macro, ctrl_sicllan.xls, the standard module whose object name is "SICL" is the definition file.

Basic control flow with SICL is shown in Figure 1-2.

NOTE In this sample program, the **ivprintf** function, the **ivscanf** function, and the **iread** function are used in its communication part; you can use other SICL functions as well. For detail, refer to sicl.hlp (the online help of SICL).

Figure 1-2 Flow of control using SICL



e5070bpe041

NOTE For more information on how to use each function of SICL, refer to the manual of SICL.

Procedures in each step in Figure 1-2 are described below.

Connection

The procedure corresponding to connection is OpenSession (Example 1-1). OpenSession establishes connection to the E5070B/E5071B with the **iopen** function of SICL, using SICL-LAN Address and IP Address entered in the part 1 in Figure 1-1. The **iopen** function takes the address information of the E5070B/E5071B you specify as its parameters.

Syntax

addr = iopen(*dev*)

Variable

	addr
Description	Session information (output)
Data type	Integer type

	dev
Description	Address information of the instrument you specify (input)
Data type	Character string type
Grammar	*sicl-name**1[*ip-address**2]:*interface**3,*sicl-lan-address**4

*1. The SICL interface name you have set with the Agilent I/O Libraries in external control.
*2. The IP address of the E5070B/E5071B.
*3. For the E5070B/E5071B, specify hpib9.
*4. The address of the E5070B/E5071B for control with the SICL-LAN server.

For example, if the parameter (*dev*) is "lan[128.10.0.3]:hpib9,17," connection is made to the address of **17** of the interface of **hpib9** with the E5070B/E5071B whose IP address is **128.10.0.3** using the external controller whose SICL interface name is **lan**.

Example 1-1 **OpenSession**

```
Function OpenSession() As Integer

  Dim ServAddr As String
  Dim IpAddr As String

  On Error GoTo ErrHandler

  '''Get Sicl-Lan Address
  Sheets("Sheet1").Select
  Range("C2").Select
  ServAddr = ActiveCell.FormulaR1C1

  '''Get Ip Address
  Sheets("Sheet1").Select
  Range("C3").Select
  IpAddr = ActiveCell.FormulaR1C1

  OpenSession = iopen("lan[" & IpAddr & "]:hpib9," & ServAddr)
  Call itimeout(OpenSession, 10000)
  Exit Function
```

```
ErrHandler:
  MsgBox "*** Error : " & Error$
  Call siclcleanup
  End

End Function
```

Sending

The procedure corresponding to sending in communication is OutputSiclLan (Example 1-2). OutputSiclLan uses the **ivprintf** function of SICL to send messages (SCPI commands). The **ivprintf** function takes the session information outputted from the **iopen** function and a program message as its parameters.

Syntax

Status = ivprintf(*addr,mes*)

Variable

Table 1-1 Variable (*Status*)

	Status
Description	Return value of the function (output)
Data type	Integer type

Table 1-2 Variable (*addr*)

	addr
Description	Session information (input)
Data type	Integer type

	mes
Description	Program message (input)*1
Data type	Character string type

*1. When sending a program message of a SCPI command, add the message terminator at the end of the message (in Example 1-2, Chr$(10)).

Example 1-2 OutputSiclLan

```
Sub OutputSiclLan(addr As Integer, message As String)

  Dim Status As Integer
  Dim actualcnt As Long
  Dim length As Long

  On Error GoTo ErrHandler

  length = Len(message)

  Status = ivprintf(addr, message & Chr$(10))
  Exit Sub
```

```
ErrHandler:
  MsgBox "*** Error : " & Error$
  Call siclcleanup
  End

End Sub
```

Receiving

The procedure corresponding to receiving ASCII format messages in communication is EnterSiclLan (Example 1-3). EnterSiclLan uses the **ivscanf** function of SICL to receive a message in ASCII format and store it into the output variable. The **ivscanf** function takes the session information outputted from the **iopen** function, the format for output, and data to be outputted as its parameters.

Syntax

Status = ivscanf(*addr,fmt,ap*)

Variable

	fmt
Description	Format for output
Data type	Character string type

	ap
Description	Data to be outputted (output)
Data type	Character string type

For information on the variable (*Status*) and the variable (*addr*), refer to Table 1-1, “Variable (Status)” in the section “Sending” and Table 1-2, “Variable (addr)” in the section “Sending,” respectively.

NOTE

In Visual Basic, variables must be declared as a fixed length string when receiving string data using the **ivscanf** function.

Example 1-3 **EnterSiclLan**

```
Sub EnterSiclLan(addr As Integer, Query As String)


  Dim Status As Integer
  Dim actualcnt As Long
  Dim res As String * 256

  On Error GoTo ErrHandler

  Status = ivscanf(addr, "%t", res)
  Query = Trim(res)
  Exit Sub

ErrHandler:
  MsgBox "*** Error : " & Error$
```

```
  Call siclcleanup
  End

End Sub
```

The procedure corresponding to receiving array data in communication is EnterSiclLanArrayReal64 (Example 1-4). EnterSiclLanArrayReal64 uses the **iread** function of SICL to receive array data in the IEEE 64 bit floating point binary transfer format and store it into the output variable. The **iread** function takes the session information outputted from the **iopen** function, data to be outputted, the number of data bytes, the condition to finish reading data, and the number of data bytes actually read out as its parameters.

Syntax

Status = iread(*addr*,*buf*,*bufsize*,*reason*,*actual*)

Variable

	buf
Description	Data to be outputted (output)
Data type	Character string type

	bufsize
Description	The number of data bytes (input)
Data type	Long integer type

	reason
Description	The condition to finish reading out data (input)
Data type	Integer type

	actual
Description	The number of data bytes actually read out (output)
Data type	Long integer type

For information on the variable (*Status*) and the variable (*addr*), refer to Table 1-1, “Variable (Status)” in the section “Sending” and Table 1-2, “Variable (addr)” in the section “Sending,” respectively.

Each functional part of EnterSiclLanArrayReal64 is described below.

(1) retrieves the data header.

(2) stores the number of data bytes into the size variable in the header part.

(3) retrieves the formatted data array for trace 1 in channel 1, and stores it into the databuf variable.

(4) retrieves the message terminator at the end of the data.

Example 1-4 **EnterSiclLanArrayReal64**

```
Function EnterSiclLanArrayReal64(addr As Integer, databuf() As Double)
As Long

  Dim Status As Integer
  Dim actualcnt As Long
  Dim buf As String * 8
  Dim size As Long

  On Error GoTo ErrHandler

  '''Read header info of "#6NNNNNN"
  Status = iread(addr, buf, 8, I_TERM_MAXCNT, actualcnt)
'.................(1)

  size = Val(Mid$(buf, 3, 6))
'.................(2)

  '''Read data
  Status = iread(addr, databuf, size, I_TERM_MAXCNT, actualcnt)
'.................(3)

  '''Read ending LF
  Status = iread(addr, buf, 1, I_TERM_MAXCNT, actualcnt)
'.................(4)

  EnterSiclLanArrayReal64 = size / 8
  Exit Function

ErrHandler:
  MsgBox "*** Error : " & Error$
  Call siclcleanup
  End

End Function
```

Disconnection

The **iclose** function of SICL is used to disconnect communication. The **iclose** function takes the session information outputted from the **iopen** function as its parameter.

Syntax

Status = iclose(*addr*)

For information on the variable (*Status*) and the variable (*addr*), refer to Table 1-1, “Variable (Status)” in the section “Sending” and Table 1-2, “Variable (addr)” in the section “Sending,” respectively.

Sample control

The E5070B/E5071B can be controlled by executing the above procedures in order, following the control flow in Figure 1-2. This is demonstrated by the procedure Preset (a procedure which is executed when the Preset button is clicked) as described in Example 1-5.

Example 1-5 **Preset**

```
Sub Preset()

  ''' Open Session
  E507x = OpenSession

  '''Presetting the analyzer
  Call OutputSiclLan(E507x, ":SYST:PRES")

  '''Close Session
  Call iclose(E507x)

End Sub
```