AB48-38

ICS ELECTRONICS
division of Systems West Inc.

IEEE 488 APPLICATION BULLETIN

An Example C Language Program making VISA Calls

INTRODUCTION

This example shows how to create a link to a GPIB device through the VISA library. Once the link is established, a query is written and a response is read from the device.

GENERAL CONCEPTS

Most GPIB devices will respond to an *IDN query with a unique text string describing the device. Thus a simple communication test consists of writing an *IDN query and then reading back the response. This confirms that bi-directional dataflow is possible and that the correct device is responding.

This example performs the minimal VISA functions required to perform an *IDN query of a user specified VISA resource. It assumes that the user defines the VISA resource name on the command line when invoking the example program. Furthermore it is assumed that the user creates a C project capable of compiling and linking the example.

REQUIREMENTS

The user must supply the VISA.H header file which defines the VISA functions and constants. In addition, the user must supply the VISA OBJ (or LIB) file to be linked to the example. The OBJ (or LIB) contains the stubs required to invoke the VISA DLL library at runtime.

It is also assumed that there is a VISA runtime system installed. On a Win32 system, this is a DLL library and is normally found in the \Windows\System32 folder. The National Instruments VISA library is named VISA-32.DLL.

EXAMPLE CODE

Figure 1 lists an example C language program that calls the National Instruments' VISA library to link to and to query a GPIB instrument.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <visa.h>


main (int argc, char *argv[])

   {
   int          retval, length;
   char         *cp, result[1024];
   ViStatus     status;
   ViSession    rmSession, devSession;
```

Figure 1   C Language Example Program

```
// The VISA resource name must be specified on the command line.
   if (argc != 2)
      {
      printf (“You must specify the VISA resource to use. This may be\n”);
      printf (“either a fully defined VISA resource, or it may be a\n”);
      printf (“VISA alias.\n”);

      exit (1);
      }

   // Before we use any VISA fuctions, we must first open the
   // Resource Manager so it will initialize the VISA layer.
   if (viOpenDefaultRM (&rmSession) != VI_SUCCESS)
      {
      printf (“Unable to open a Resource Manager session\n”);
      exit (1);
      }

   status = viOpen (rmSession, argv[1], VI_NULL, 1000, &devSession);

   if (status != VI_SUCCESS)
      {
      printf (“Unable to open a session with %s\n”, argv[1]);
      retval = 1;
      }
   else
      {
      status = viWrite (devSession, “*IDN?\n”, 6, &length);

      if (status != VI_SUCCESS)
         {
         printf (“Unable to send an *IDN? query to %s\n”, argv[1]);
         retval = 1;
         }
      else
         {
         status = viRead (devSession, result, sizeof(result), &length);

         if (status != VI_SUCCESS)
            {
            printf (“Error reading the *IDN? query response from %s\n”,
                                                            argv[1]);
            retval = 1;
            }
         else
            {
            // Null terminate the received string since he viRead does not
            // auto-magically Null terminate the data. Note that this will
            // not remove the \n if one is included in the result string.
            result [length] = ‘\0’;

            // If the result string contains a \n character, terminate the
            // string at that point - replacing the \n with a Null.
            cp = strchr (result, ‘\n’);

            if (cp)
               *cp = ‘\0’;

            // Display the *IDN response string.
            printf (“*IDN? response:\n”);
            printf (“  [%s]\n”, result);

            // Signal that no errors happened.
            retval = 0;
            }
         }
      }

   // Close the Resource Manager now that we’re done with using VISA.
   viClose (rmSession);

   return (retval);
   }
```

Figure 1 C Language Example continued