

# IEEE 488

## APPLICATION BULLETIN

### FREQUENTLY ASKED QUESTIONS (FAQ) FOR THE ICS's MODEL 8065 ETHERNET-TO-GPIB CONTROLLER

#### INTRODUCTION

This application note lists the Frequently Asked Questions (FAQ) for ICS's Model 8065 Ethernet-to-GPIB Controller. This FAQ list changes often so look here first for the latest answers to your problems.

#### **FAQ #1 What operating system do I need to run ICS's 8065?**

The Model 8065 is operating system independent. The 8065 is a VXI-11 compliant interface and can be connected to any IEEE-802.11 TCP/IP network. Any computer with network access and that is capable of running RPC over a TCP/IP network can be used with the 8065. This includes computers with WIN 32, UNIX, LINUX, Apple's OS X, SunOS, HP-UX, IBM-AIX, to name a few of the compatible operating systems. For JAVA applications, see FAQ #9.

#### **FAQ #2 How do you use Agilent's VISA with the 8065?**

Agilent's current VISA is not fully VXI-11 compliant. The "find devices" and "find servers" both have problems due to incompatibility with VXI-11 protocol. Agilent was too far along with their release process before we could test it with the 8065 and notify them of the problems. Agilent is aware of this problem and says they will fix it in the next release.

You must manually define the VXI-11 devices in the same way you do with the NI VISA. Once you have manually defined the VXI-11 devices (TCP/IP resources), the Agilent VISA works with the 8065. See the detailed directions on using Agilent's VISA in the 8065 Instruction Manual.

Note that the Agilent's VISA implements most but not all of the VXI-11.2 Interface Functions for TCP/IP. However, since Agilent's VISA is a wrapper for their SICL library, you can make SICL calls for the missing VXI-11.2 functions. Refer to ICS's Application Bulletin AB80-2.

#### **FAQ #3 How does ICS's 8065 compare against Agilent's E5810?**

As far as we can tell, Agilent's E5810 does implement most of the VXI-11.2 Interface functions and the VXI-11.3 Instrument functions, like Write, Read, Serial Poll and Clear. The E5810 does have some non-standard VXI-11 commands that increase its interface capability. However the E5810 only supports one client.

ICS's 8065 does implement all of the VXI-11.2 Interface and the VXI-11.3 Instrument functions. ICS's 8065 also supports multiple clients.

#### **FAQ #4 How does ICS's 8065 compare against NI's eNET/100?**

As far as we can tell, NI's eNET/100 does not implement any VXI-11 functions. The eNET/100 uses a proprietary NI protocol to control GPIB instruments and must be used with NI's VISA or NI drivers. The NI eNET/100 only supports one client unless you buy additional software.

ICS's 8065 does implement all of the VXI-11.2 Interface and the VXI-11.3 Instrument functions. The 8065 supports multiple clients as part of its standard firmware.

#### **FAQ #5 How do you use National Instruments' (NI) VISA with the 8065?**

NI's VISA cannot automatically find the GPIB devices connected to the 8065. NI's VISA requires that you run their Measurement and Automation Explorer (MAX) to set the device resource address before running the VISA Interactive Control Panel. LabView VIs and other programs can tell VISA the device resource address so it is not necessary to run MAX when running an application program.

When you run MAX, select the TCP/IP resource and follow the windows to manually enter the 8065's IP address and the instrument's GPIB address. You will need one entry for each instrument. You can now run the VISA Interactive Control Panel and communicate with the instrument. See the detailed directions on using NI's VISA in the 8065 Instruction Manual.

NI's VISA does support VISA calls and TCP/IP resources so it can be used with the 8065. However, NI's VISA only makes VXI-11.3 calls and does not support VXI-11.2 interface calls so it cannot perform the VXI-11.2 GPIB Bus functions.

#### **FAQ #6 How can I program the VXI-11.2 functions?**

There are three good solutions.

If you are a C or Visual Basic language programmer, then you can make VISA library calls from your program. Agilent's VISA library gives you full control of the GPIB bus and communication with GPIB devices through the 8065. You get the VISA library as part of Agilent's IO Library Suite version 14.2 or later. Note the NI VISA library only supports VXI-11.3 Instrument functions.

If you are a C or Visual Basic language programmer and like Agilent's SICL library, then you can make SICL library calls from your program. Agilent's SICL library gives you full control of the GPIB bus and communication with GPIB devices through the 8065. You get the SICL library as part of Agilent's IO Library Suite version 14.1 or later. Refer to ICS's Application Bulletin AB80-2 for a SICL programming example and SICL programming hints.

The third solution is to use RPC commands and call them from your program. The VXI-11 Specification includes the RPCGEN information to create the .h and .obj files for your program. RPC calls are familiar to many UNIX, LINUX, Apple's OS X, SunOS, HP-UX, IBM-AIX and similar operating system programmers. You can use RPC calls with C, Java, Visual Basic or any programming language that can make RPC calls.

#### **FAQ #7 Can I install a 8065 on a Windows system?**

You really do not install the 8065 but rather install the VXI-11 Support Tools on a Windows 2K or later Windows operating system. Run the ICS\_VXI-11\_Install program to install the VXI-11 Configuration and VXI-11 Keyboard programs.

Like all network devices, the 8065 has to have its network setting set to match your network. Use the VXI-11 Configuration utility program to change the 8065's network settings when 'installing' the 8065 on your network.

If you make RPC calls, the 8065 Instruction Manual includes the ICS's RPC Configuration Commands. The desired commands can be run through RPCGEN and then called by a script or console operation to change the 8065's settings.

#### **FAQ #8 What kind of network security does the 8065 have?**

The short answer is none. The VXI-11 Specification does not describe a security or password protection scheme. However, only client applications that can generate VXI-11 commands can communicate with the 8065. Browsers and other programs cannot affect the 8065's operation.

Network security is different for each company and it is up to the IT manager to define his needs. Routers and firewalls can be used to create secure subnets or to protect a remote 8065 from a hacker.

#### **FAQ #9 Is there a Java program example for the 8065?**

Yes. A 8065 user has posted a Java example and drivers on SourceForge.net. The example name is jGpibEnet and the target GPIB Controller is ICS's 8065. The link to the Java example is: <http://sourceforge.net/projects/jgpibenet>.

#### **FAQ #10 How do I transfer large blocks of data with the 8065?**

The 8065 has a *maxRecvSize* of 1024 bytes that limits the amount of the data that can be transferred to the 8065 in one block. *maxRecvSize* is the last parameter returned when the link is created to the 8065.

The user can transfer larger amounts of data to a GPIB device by dividing the desired amount of data by the *maxRecvSize* and then sending *maxRecvSize* blocks of data until all of the data has been transferred. The 8065 does not terminate a write operation until it receives a packet with the end condition set.

Reading large amounts of data from a GPIB device works the same way. The 8065 does not terminate a read operation until the end condition is met. There is no readdressing of the GPIB device between packets when multiple packets are used to transfer large amounts of data. The client can request a read of more than the *maxRecvSize* number of bytes but the 8065 will only send a packet with 1024 bytes and with no reason bits set if there is more data to be sent. The client continues reading packets until it receives a packet with one or more reason bits set. See VXI-11 Rule B.6.23.

The recommendation is to read *maxRecvSize* when the link to the 8065 is created and to use this value to calculate the size of the data packets. That way, future changes in the 8065 will not affect your program.

#### **FAQ #11 How can I assign a host name to an 80xx Device?**

You can assign host names to ICS's 80xx devices in Win32 Systems by adding them to the host file. Open the host file at C:\WINNT\system32\drivers\etc\host with Notepad and add a line for each device you want to name with the device's IP address (tab) hostname. i.e

```
192.168.0.99      tempchamber
```

Save the file and test it with the VXI11\_kybd program by entering your new hostname in the Found Servers window and clicking the Link to Server button. Copy the host file to each PC.

## FAQ #12 Can I use MATLAB to control the 8065?

Yes. You need to create a device object or interface object in the MATLAB tmtool. See the following example. File esg1.m - was generated by creating a device object in the Matlab tmtool. File esg2.m - was generated by creating an interface object in the Matlab tmtool with the instrfind function and a gpib Boardindex of 7.

The device object example is:

### function esgConfig (Device object)

```
%ESGCONFIG M-Code for communicating with an instrument.
%
% This is the machine generated representation of an instrument control
% session using a device object. The instrument control session comprises
% all the steps you are likely to take when communicating with your
% instrument. These steps are:
%
% 1. Create a device object
% 2. Connect to the instrument
% 3. Configure properties
% 4. Invoke functions
% 5. Disconnect from the instrument
%
% To run the instrument control session, type the name of the M-file,
% esgConfig, at the MATLAB command prompt.
%
% The M-file, ESGCONFIG.M must be on your MATLAB PATH. For additional
% information
% on setting your MATLAB PATH, type 'help addpath' at the MATLAB
% command
% prompt.
%
% Example:
%     esgConfig;
%
% See also ICDEVICE.
%
% Creation time: 07-Aug-2006 02:27:55
```

```
instrhwinfo visa
instrhwinfo('vxipnp')
path
% Create a device object.
deviceObj = icdevice('hpESG.mdd', 'TCPIP::192.168.3.100::gpib0,19::
INSTR');

% Connect device object to hardware.
connect(deviceObj);

% Execute device object function(s).
deviserest(deviceObj);
%%Disconnect
% Disconnect device object from hardware.
disconnect(deviceObj);
% Delete object.
delete(deviceObj);
```

The Interface object example uses the instrfind function and a GPIB boardindex of 7. The example is:

### function esg2 (Interface object)

```
%ESG2 M-Code for communicating with an instrument.
%
% This is the machine generated representation of an instrument control
% session. The instrument control session comprises all the steps you
% are
% likely to take when communicating with your instrument. These steps
% are:
%
% 1. Create an instrument object
% 2. Connect to the instrument
% 3. Configure properties
% 4. Write and read data
% 5. Disconnect from the instrument
%
% To run the instrument control session, type the name of the M-file,
% esg2, at the MATLAB command prompt.
%
% The M-file, ESG2.M must be on your MATLAB PATH. For additional
% information
% on setting your MATLAB PATH, type 'help addpath' at the MATLAB
% command
% prompt.
%
% Example:
%     esg2;
%
% See also SERIAL, GPIB, TCPIP, UDP, VISA.
%
% Creation time: 07-Aug-2006 06:46:06

% Create a GPIB object.
obj1 = instrfind('Type', 'gpib', 'BoardIndex', 7, 'PrimaryAddress', 19,
'Tag', '');

% Create the GPIB object if it does not exist
% otherwise use the object that was found.
if isempty(obj1)
    obj1 = gpib('AGILENT', 7, 19);
else
    fclose(obj1);
    obj1 = obj1(1)
end

% Connect to instrument object, obj1.
fopen(obj1);

% Communicating with instrument object, obj1.
data1 = query(obj1, '*IDN?')

% Disconnect from instrument object, obj1.
fclose(obj1);

% Clean up all objects.
delete(obj1);
```

**Disclaimer** - While we believe the examples in this Application Note or referenced herein are correct, the user assumes all responsibility for their operation in his system. Neither ICS Electronics or the program authors assume any responsibility for their operation.