

# FOUNDATION FIELDBUS INTEROPERABILITY TESTING – THE MAN (OR WOMAN) BEHIND THE CURTAIN

John Yingst  
Fieldbus Systems Product Manager  
Honeywell Automation & Control Solutions  
Phoenix, AZ

## KEYWORDS

Foundation Fieldbus, Testing, Device Interoperability, Reliability, Robustness

## INTRODUCTION



FOUNDATION

The stated goal of FOUNDATION Fieldbus has always been device interoperability, ensured by the use of an open, non-proprietary Fieldbus protocol, standard user layer function blocks and Device Description technology. Device Description (DD) files are one of the keys to allowing devices to coexist and communicate on the same H1 Fieldbus without special drivers or software. To assure interoperability, device suppliers are required to test and register devices with the Foundation and make registered files available to the users. Host Systems are also tested by the Foundation using the Host Interoperability Support Test (HIST) to further assure interoperability.

So, why are all the system suppliers testing devices? Today, nearly every DCS vendor maintains a Fieldbus interoperability test laboratory of some capacity for testing Foundation Fieldbus devices with their respective host control system. And what exactly are host suppliers doing that either the Foundation or the device suppliers simply cannot? And finally, what is the end benefit to the customer?

This paper will explore these questions and take a close look at what all technology providers, whether of host systems or devices, are doing to make FOUNDATION Fieldbus work for everyone.

## SOME HISTORY

From the very beginning, the framers of the FOUNDATION Fieldbus specifications intended for devices – transmitters and actuators alike – to be able to **reside on a powered bus** and **communicate with each other** and **with a host**, regardless of which vendors the devices or the hosts came from. This was to be enabled via standard electrical and communication protocols and with the ability to run control in devices or in the host. The Foundation specified that standard parameters would be present in devices, with the option to include manufacturer-specific parameters as well. These optional parameters would be the way that manufacturers could differentiate themselves. A temperature transmitter, for example, would have a very different set of parameters from a valve positioner.

In most cases, the host being referred to would be a distributed control system (DCS), but it could also be a laptop or handheld bench configuration tool or simply a PC program. The ability of a host to recognize a device and know all about its parameters and capabilities was made possible by the Device Description (DD) files and Capability Files (CF). This set of files that describes a device is commonly referred to in whole as its device files. These device files define the function blocks (FBs) and capabilities of a device, thus enabling a host system to know everything needed about interfacing to a new Foundation Fieldbus (FF) device without ever seeing the device itself. Thus, device files are the key to off-line configuration.

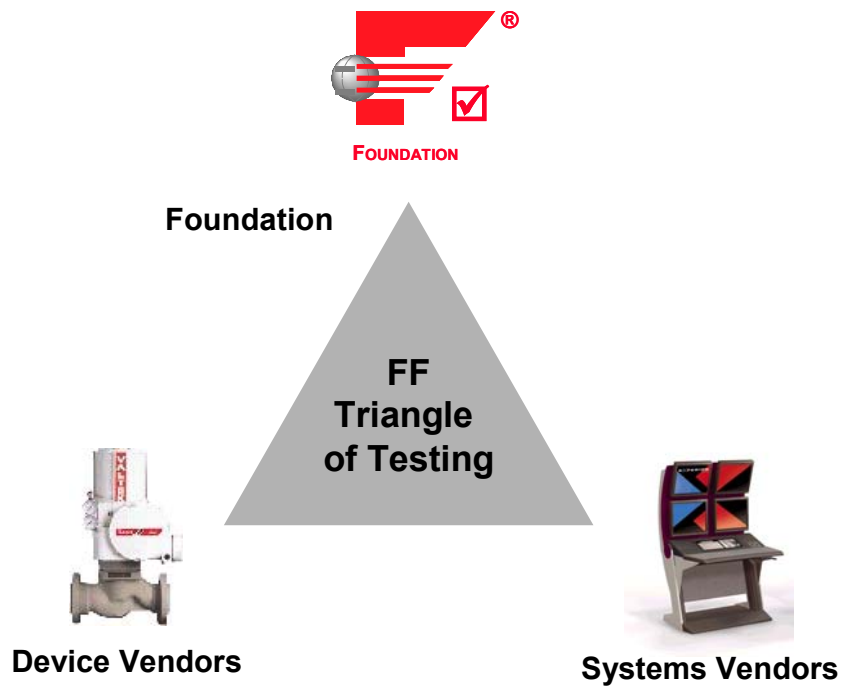
**The need for testing.** In a perfect world with perfect specifications and a complete understanding of these specifications, host and device vendors could go off and develop products that work together perfectly. But in the real world of Fieldbus devices, with complex software developed by real people, it was recognized that testing would be needed – lots of testing. And so the Foundation started out requiring devices to be Foundation tested and registered (more about this testing later), with initially no host system testing required. Most device vendors performed (and still do perform) their own “pre-qualification” before submitting for FF registration. Today, tools provided by the Foundation are used to test the FF communication protocol. National Instruments Fieldbus Configurator is commonly used for device testing. Most device vendors use this package as their “gold standard” against which they test and fix problems beyond the protocol layer.

In the mean time, the DCS world was starting to encounter its own set of challenges. Devices and device files that passed FF testing and might work fine with one system yet have problems with another. Each host supplier was free to implement their own set of Fieldbus functions and features, and several different stacks (protocol implementations) were available and used. In addition, different vendors (both system and device) might have slightly different interpretations of the Fieldbus specification.

So, two things started happening very early on. First, as customer projects required the various DCS systems to work with various available devices, the DCS vendors found they had to implement more formal testing programs. Customers started demanding that devices be tested or “qualified” to provide some assurance that their project implementation would work successfully. And second, as problems were encountered, host system and device suppliers started working together (and with the Foundation) to resolve these problems. Today, most all the major system vendors have test programs in place. Those programs represent the “man (or woman) behind the curtain,” quietly helping to ensure that Fieldbus projects go smoothly and problems are worked out successfully.

### THE “TRIANGLE” OF TESTING

Today, one element contributing to the ongoing success of FOUNDATION Fieldbus is a “**triangle**” of **testing**, consisting of the Foundation, the device suppliers, and the host system suppliers (see Figure 1). Credit must be paid to the many customer test labs and pilot plants, independent consultants and learning centers, but the burden of responsibility is mostly born by these three sectors. Let’s examine what each does and does not do.



*Figure 1. Triangle of Testing*

In addition to providing the technology and organization behind Fieldbus, the **Fieldbus Foundation** is responsible for testing that devices meet the FF specifications. The FF “registers” devices which pass its tests. These tests are extremely important, but there are limits to them. In fact, the FF does not “certify” devices. They do not guarantee that a device will behave in a certain way. Their charter is to make sure that the protocol and standard have been followed. The FF puts devices through rigorous tests to insure that electrical properties of the protocol are followed. The FF Interoperability Test Kit tests devices against the

specification. They test communication protocol functions, checking that a device contains a “registered” stack that passed FF conformance testing.

While FF does run a device interoperability test to check that function blocks in different devices communicate, this does not necessarily ensure correct function block control behavior. FF does not stress test or subject devices to performance related measurements. The FF DD test ensures proper syntax and that it meets certain standards. The CFF test ensures that the capabilities file defines the capabilities present in the device. Again, the primary purpose is to ensure that the FF protocol and specifications have been followed. That is what earns each device its FF checkmark. The Fieldbus Foundation does NOT ensure that devices will behave correctly with systems. Given all of the possible combinations, this would be an impossible task.

**Device vendors** are primarily interested in ensuring that their devices work as intended. They are mostly focused on device functionality (the manufacturer-specific part embedded in the transducer blocks), where their products can be differentiated. Most device vendors rely on third parties, such as Softing, SMAR, or National Instruments, for their stack, and some purchase function blocks from outside sources. Device vendors purchase tools from the Foundation for pre-testing their devices, which helps insure in advance that FF interoperability tests will pass, but these tools do not test things such as algorithm behavior. Many device vendors have limited testing capability, sometimes only using NI Configurator as a user test tool, and few have host systems. NI Configurator does not respond, for example, to alarm behavior. Fortunately, vendors have started acquiring host systems and are now taking a more active role in making sure that all behavioral aspects of their devices are correct. This is all part of the learning process. But their primary purpose is still to make sure their devices work properly and to specification.

**Systems vendors** have responsibility not only for interfacing to and integrating with Fieldbus devices but for the overall system control behavior, which includes function blocks in the devices. In the end, the customer expects the whole system to work correctly. For that reason, as well as a number of others to be explored here, system vendors have taken on a strong role in device testing. Today, formal Fieldbus interoperability testing programs are in place by Honeywell (PlantScape and Experion PKS), Emerson (DeltaV and Ovation), Yokogawa (Centum), ABB (Industrial IT), Rockwell Automation and SMAR (System 302). (The author apologizes if any others were missed.) Although individual practices and policies may vary, the end purpose is that same – to ensure that available devices work correctly and dependably with the control system.

There are many reasons why testing by the systems vendors makes good sense. These include:

- behaviors that are best observed with host systems, such as alarming and event reporting;
- device capabilities evolving at a different pace from the systems capabilities (e.g., block instantiation);
- new capabilities being added to the specifications all the time;
- new twists in the specification being tried (i.e., aspects of the spec that were not excluded but no one had yet tried); and
- possible uses of features not intended to be supported, but not explicitly prohibited.

Vendors also test because (a) customer expect system assurance, (b) problems are sometimes found with DD/CFF files, (c) problems are sometimes found with device behaviors (even if FF certified), and (d) host vendors want to debug and improve their systems. That last reason is very important, as all of this testing has helped make the control systems stronger as well. Without Fieldbus Interoperability Testing by the system vendors, it doesn't all work!

## ROLE OF HIST TESTING

Before proceeding further, it would be appropriate to comment on the **Foundation's Host System Interoperability Test**, also known as HIST, which is intended for qualification of FF hosts. An excellent definition of this procedure can be found on the Foundation website, so it will not be repeated here. In a nut shell, HIST is a test to show that a system can perform certain basic categories of Fieldbus functions. Although it serves as a means to establish which functions a host has implemented, it does not guarantee that those functions will work correctly for all registered devices and applications. This is not meant to be a criticism of the HIST procedure, but just a statement of reality. It is not intended to be a comprehensive test of functionality. While it is an extremely valuable tool, the HIST test is not a substitute for host system interoperability testing by vendors. This is not likely to change in the near future. Also it should be pointed out that HIST was meant to be applied over a broad range of products, from full-blown control systems to lab and bench configuration tools. There are many more capabilities required of a control system than of a bench calibration tool, and HIST must be able to be applied at both ends of the spectrum.

## GENERAL TESTING POLICIES

This section will describe **interoperability testing policies** as presently practiced at Honeywell. In many cases, some of these same general policies are used by other vendors as well, but the rules may vary from vendor to vendor, and no attempt has been made to compare or contrast them.

In general, system vendors have adopted policies of testing devices at no charge to the customer or to the device vendor. In return, the system vendors only ask to be provided with the devices to test. To date, this seems to be a well-established and accepted practice. Credit should be given here to an FF standard which, as presently implemented in most systems, makes integrating a new device into a system relatively easy. This helps keep the relative testing effort required quite reasonable.

Practices for informing customers and device vendors which devices have been tested vary among system vendors. At Honeywell, we post the device files for devices which have been successfully tested on our public web site, <http://www.acs.honeywell.com>. This is our list. Since our system makes direct use of device files, it is also an important mechanism to support customer projects. Most other companies seem to be making some use of web tools for that purpose. It is important to point out that we specifically post the device files that work with our system, as these files may or may not have had to be modified or corrected, or there may be multiple versions of files available. We also include any information specific to integration of that device with our system. The latter is probably the most compelling and legitimate reason for each system vendor posting device files. Even if the device files as provided work perfectly, there may be other system nuances that must be pointed out.

So, what happens when testing reveals some kind of problem or issue? Historically, vendors try to work together to iron these out, and that is always the first course of action. Patience is important here, as fixes and changes can take time, and this process sometimes involves retesting, but sometimes customers needs dictate the pace. If there is a problem or behavior of concern which is not resolved, especially concerning control, we try to provide some information that is mutually acceptable to the devices vendor with the device files. Note that we do not publish detailed test reports; we consider the device file posting sufficient notice.

Our interoperability test plan and a policy are not posted publicly but are available to customers and device vendors upon request. This seems to vary considerable among vendors.

## TESTING STEPS

The primary objective of interoperability testing is to ensure system support for selected FOUNDATION Fieldbus devices as required by customers and to qualify registered devices as interoperable with the host system. This objective is fairly consistently echoed around the industry as other systems vendors support their customers. Documenting interoperability and functional issues discovered during tests and working to resolve issues with vendors are also important objectives.

Below is a **basic outline of steps followed** in testing a device. These apply to our program, but are generic enough to apply to that of any vendor.

- A representative device and device files are obtained from the vendor or the customer.

- A system representation of the device (what we call a Library Template) is built from the device files. This is the first test of the device files. (This step varies from vendor to vendor.)
- If a problem is encountered with the files and we are able to fix it, we do so and notify the vendor. In rare cases where we cannot fix a problem, we work with the vendor to resolve it. Sometimes, we find that we must make a change or modification to our system.
- The next step is to load the device, which (in our system) also loads the Resource and all Transducer blocks. We want to make sure (a) the device is stable using default network protocol settings and (b) all parameters appear to work properly. This is done in a system with a mix of devices from other vendors present, as we are looking for any potential interoperability issues. Preferably, the H1 network is heavily loaded.
- Once basic operation of the device is verified, various available function blocks are used and tested in control strategies. This step ranges in scope from just testing one AI or DI block to a large combination of input, output and function blocks. (More detail is provided in the next section.)
- Function blocks are tested in both “control on the wire” and “control in the host” configurations. These are both very important test scenarios. Also, different (and multiple) macrocycle times are used.
- Other behaviors and characteristics tested include confirmation that:
  - Read/write parameters can be accessed and written within vendor specified limits;
  - Read-only parameters can be read and contain expected values as specified by the vendor;
  - Device address assignments and tag names can be changed;
  - Optionally, device calibration can be performed if documentation is available;
  - For devices capable of acting as Link Active Scheduler (LAS), link mastership, LAS fail-over and recovery all work correctly;
  - System behaviors, such as alarming, bus power failure recovery, connection/disconnection from a live bus, and device save/restore work as expected; and
  - Options presented in the Resource block work correctly, especially event reporting.
- Note that some devices have extra Transducer blocks which are vendor-specific and should be tested. For example, most Rosemount transmitters have a TRANSDUCER\_LCD block that controls behavior of the display.

## CONTROL BEHAVIOR TESTING

The use of function blocks is unique to FOUNDATION Fieldbus, and so special attention is paid to testing function block control behaviors. This is a testing area of particular interest to control system suppliers because, in one sense, FOUNDATION Fieldbus devices are “another part” of a bigger picture – the control system. Also, most systems vendors are grounded in the fundamentals of control and well understand the behaviors that must work correctly.

Below are examples of typical **test cases** run to test function block behaviors in Fieldbus devices. Figure 2 shows a typical example of an interoperability control test application for PID. Keep in mind that input and output blocks are tied to transducer blocks, and so these are tested together. Control function blocks (PID, etc.) have no such links.

- Test simple input device (AI and DI) connections to host and to other device function blocks. Verify that VALUE and STATUS are propagated correctly by causing alarms and changing MODE of AI or DI block. Verify alarming behavior, including inhibit and priority. Verify that publications are being updated every macrocycle. Verify that range works correctly with transducer block. In cases where there are multiple channels, verify that all channels work correctly.
- Test PID control behavior, using IN, CAS\_IN, RCAS\_IN, ROUT\_IN, and FF\_VAL as inputs. Verify that various control and PV options work correctly. Verify that mode shedding works correctly. Verify alarming behavior. Verify various range options work.
- Test output behaviors to AO and DO blocks. As with PID, use CAS\_IN, RCAS\_IN, and ROUT\_IN as inputs. Test that mode shedding works correctly. Test that any readback or feedback functions work correctly. For multi-channel devices (e.g., multiple DO blocks), verify that channel assignments work correctly and do not conflict.
- Test all other FUNCTION BLOCKS available in the device. Refer to device-specific instructions for various control options available to test.
- Test device blocks – input, output and control – with the host function block applications as well.
- Since there are two types of block connections, publish/subscribe and asynchronous connections, both should be tested. For example, RCAS\_IN or RCAS\_OUT are remote client-

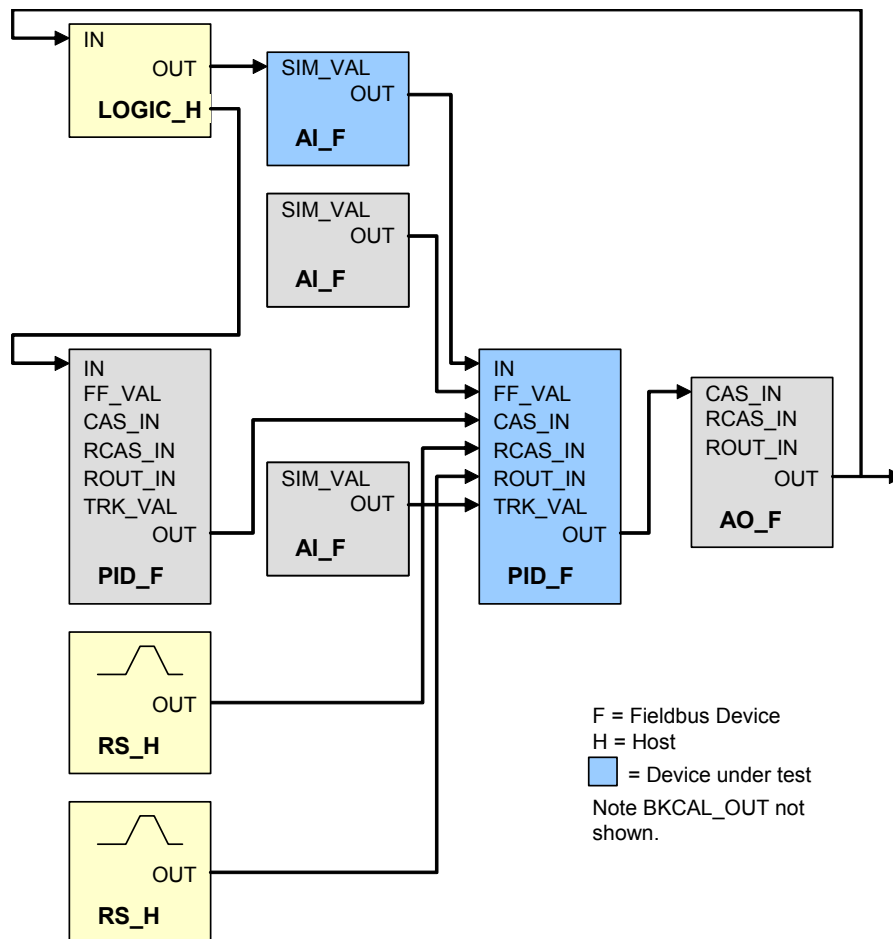


Figure 2. Typical Fieldbus Interoperability Control Application

server connections, intended for DDC control, while CAS\_IN is a publish/subscribe connection. Initialization behavior should be verified where applied. These tests should also be done with a heavily loaded link.

- Confirm fault state operation of functions blocks as well as of the device itself. Note in many cases there are fault states which cannot be tested because they may be destructive, difficult to simulate, or because they are unknown to the system vendor.

## TROUBLESHOOTING AND PROBLEM RESOLUTION

The subject of **problem resolution** has already been discussed in some length. When testing reveals problems, a number of techniques and tools are used to troubleshoot and provide details about what is going on. Some include:

- Use of NI Configurator or other bench calibration tool like the Rosemount 375 to investigate whether a problem appears to be in the device or with the system. For example, NI Configurator might be used to confirm that a certain parameter cannot be written to even though it is specified to be read/write.
- Use of NI Monitor to capture events on the bus and verify that a device's behavior is correct or in error. This requires a highly detailed understanding of fundamental FF protocol behavior and is not something the typical customer would have knowledge of.
- Use of DD Viewer to examine the details of the DD files provided. For example, we can see more details about intended use of a parameter. Again, this requires highly detailed understanding of the FF specifications.
- Use of other devices (which have been successfully tested) to compare and contrast behaviors such as alarming or control functions.

The question often comes up as to whether a problem is with the device or the system. Sometimes the FF technical specifications must be referred to in order to confirm that there has been some kind of specification violation, but occasionally “gray areas” are encountered which must be resolved with the involvement of the Foundation. When this happens, one or more parties can file an AR (Action Request) to essentially get a “ruling” on a technical issue. ARs further clarify ambiguities in the specification and act as a way to help improve the FF technology overall. A common misconception is that since a registered device is tested, a given interoperability issues **MUST** be the fault of the host.

An important note about differences between devices should be made here. While FF device vendors are required to use certain standard parameters in their devices and function blocks, and they are permitted to add their own vendor-specific parameters, there is no requirement that all internal functional behaviors of the standard blocks be completely identical. Some behaviors are standardized, while others are not. For example, PID blocks in two different vendors' devices might look the same but not act exactly the same. The internal equations, etc., remain proprietary to the vendors. This is allowable and provides for differentiation. In practice, most differences observed seem to be minor, and customers certainly are well served by as much consistency as possible.

### HONEYWELL'S TEST FACILITIES

Honeywell maintains two separate systems specifically devoted to testing Fieldbus devices with Experion PKS, each having a different focus. Our primary facility for detailed device testing is the Fieldbus Interoperability Test Lab in Bangalore, India, facility run by Honeywell Technology Solutions Lab (HTSL). One of our primary test engineers is shown in Figure 3 with the system in the lab. The lab is also shown in Figures 4 and 5. Previous discussions about test methodology apply to this facility. Testing of an average Fieldbus device can require as much as a week of effort, but this can vary considerably from device to device depending on number of function blocks and complexity of the device. As one would expect, this also varies depending on issues encountered.



***Figure 3. Honeywell Fieldbus Test Engineer in Bangalore, India***

We also maintain a large system Experion PKS test bed at our development facility in Ft. Washington, PA. The focus of this system is on large system performance with full H1 loading and heavy alarm and display loading. As part of the lab, shown in Figures 6 and 7, we incorporate devices from several vendors.

In addition, a number of smaller systems are maintained for various development, troubleshooting and demonstration purposes. Included among them are systems run by Honeywell's Technical Assistance Center (TAC) in Phoenix, Arizona. Sometimes these systems are involved in troubleshooting and preliminary device testing, but primary responsibility for the testing belongs to the Bangalore facility.



**Figure 4 & 5. Honeywell Fieldbus Interoperability Test Lab in Bangalore, India**

#### WHAT DOES THIS ALL MEAN?

The title of this paper refers to the “man (or woman) behind the curtain.” This phrase is meant to infer that the interoperability testing effort by systems vendors is an important ongoing element of FF reliability that is not all that visible to the end users. But without it we lack the important system verification so essential to this technology. System vendors play an important role in establishing and maintaining the robustness and reliability of Foundation Fieldbus devices as well as systems.

In 2000, it was predicted that the improved Capabilities Files standard from the Foundation would provide some “light at the end of the tunnel” with respect to the amount of testing required. Now the same is being said of the latest move to support totally unmodified device files from the Foundation, and it is believed that this will represent a major improvement. CFF 1.7 is a direct result of modifications to the specification based on HIST team comments. But these changes take time, and, with the dynamic and evolving nature of the technology, from this vantage point it appears as though testing may be with us for some time to come.

It is important to recognize that testing in all three “corners” is essential to insure uncomplicated commissioning using a complex underlying technology. Fieldbus Foundation, device vendor, and host system supplier testing all play an important role. The different testing combinations, along with hundreds of thousands of devices in operation, demonstrate the reliability and viability of this technology.



*Figure 6 & 7. Honeywell Large System Test Lab in Ft. Washington, PA*

### FINAL RECOMMENDATIONS

If you have an upcoming project, it is recommended that you contact your system vendor early to make sure that devices you want to use are being or have been integration tested. Also, it is critical to establish which revisions will be used. Testing the wrong revision might not provide any validity, as vendors can and have made major changes to devices as part of a revision change.