USB-8486

FOUNDATION™ Fieldbus Overview

January 2014
370729D-01

NATIONAL INSTRUMENTS™

Worldwide Technical Support and Product Information

`ni.com`

Worldwide Offices

Visit `ni.com/niglobal` to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 683 0100

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on National Instruments documentation, refer to the National Instruments website at `ni.com/info` and enter the Info Code `feedback`.

Legal Information

Warranty

NI devices are warranted against defects in materials and workmanship for a period of one year from the invoice date, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from the invoice date, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

End-User License Agreements and Third-Party Legal Notices

You can find end-user license agreements (EULAs) and third-party legal notices in the following locations:

- Notices are located in the <National Instruments>_Legal Information and <National Instruments> directories.
- EULAs are located in the <National Instruments>\Shared\MDF\Legal\license directory.
- Review <National Instruments>_Legal Information.txt for more information on including legal information in installers built with NI products.

Trademarks

Refer to the *NI Trademarks and Logo Guidelines* at `ni.com/trademarks` for more information on National Instruments trademarks.

ARM, Keil, and µVision are trademarks or registered of ARM Ltd or its subsidiaries.

LEGO, the LEGO logo, WEDO, and MINDSTORMS are trademarks of the LEGO Group.

TETRIX by Pitsco is a trademark of Pitsco, Inc.

FIELDBUS FOUNDATION™ and FOUNDATION™ are trademarks of the Fieldbus Foundation.

EtherCAT® is a registered trademark of and licensed by Beckhoff Automation GmbH.

CANopen® is a registered Community Trademark of CAN in Automation e.V.

DeviceNet™ and EtherNet/IP™ are trademarks of ODVA.

Go!, SensorDAQ, and Vernier are registered trademarks of Vernier Software & Technology. Vernier Software & Technology and `vernier.com` are trademarks or trade dress.

Xilinx is the registered trademark of Xilinx, Inc.

Taptite and Trilobular are registered trademarks of Research Engineering & Manufacturing Inc.

FireWire® is the registered trademark of Apple Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Patents

Export Compliance Information

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

Contents

Chapter 1
What Is Fieldbus?

Chapter 2
Foundation Fieldbus Technology and Layers

Chapter 3
Fieldbus Communication

Chapter 4
Wiring and Connections

Chapter 5
National Instruments Fieldbus Products

Appendix A
Technical Support and Professional Services

Glossary

Index

1

What Is Fieldbus?

This chapter gives a short overview of Foundation Fieldbus and describes the parts of a Foundation Fieldbus network. Refer to the *Glossary* for more explanation of Fieldbus terms and concepts. The generic term *fieldbus* refers to any bus that connects to field devices. This includes Foundation Fieldbus, NI-CAN, DNET, and Profibus. In this manual, the term *Fieldbus* refers specifically to the Foundation Fieldbus.

Related Documentation

General Fieldbus Web sites

- Fieldbus Foundation—Responsible for the definition of the Foundation Fieldbus specification. `www.fieldbus.org`
- Relcom, Inc.—Wiring and test equipment for Foundation Fieldbus. `www.relcominc.com/fieldbus`
- Fieldbus, Inc.—Device developers and Foundation Fieldbus consulting. `www.fieldbusinc.com`

Fieldbus system development documentation

- Fieldbus Foundation resources
  - *Wiring and Installation 31.25 kbit/s, Voltage Mode, Wire Medium*
- Relcom, Inc. Fieldbus resources online
  - *Wiring Design and Installation Guide*
  - Online Tutorial
  - FAQ
  - Sample Fieldbus Topologies
- *Fieldbus Standard for Use in Industrial Control Systems, Part 2, ISA-S50.02.1992*

Fieldbus device development documentation

- *How to Develop Your First Foundation Fieldbus Device*, available from the Fieldbus, Inc. website at `www.fieldbusinc.com`
- *Foundation Specification: HSE Presence* for the Foundation Fieldbus HSE Physical Layer Specifications
- *Foundation Specification: 31.25 kbit/s Physical Layer Profile* for the Foundation Fieldbus Physical Layer Specifications
- Foundation Fieldbus specification *Function Block Application Process, Part 1*

- Foundation Fieldbus specification *Function Block Application Process, Part 2*
- Foundation Fieldbus specification *System Architecture*
- IEC Standard 1158-2 and ISA Standard ISA S50.02

The Fieldbus Foundation

The Fieldbus Foundation is the organization that defines the Foundation Fieldbus specification and certifies products to be compliant with the standard. The Foundation Fieldbus standard defines the way you can bring new devices into the network, set them up, and configure them. Any company with the proper resources can make a Foundation Fieldbus device (if it passes the Conformance Test) that will work with all other Foundation Fieldbus-certified devices and software.

The goal of the Fieldbus Foundation is to help create products that use a robust industrial network based on existing standards and other proven technologies and to standardize using those sources. Foundation Fieldbus is an open standard, which allows you to use Foundation Fieldbus products from different vendors interchangeably. For more information about the Fieldbus Foundation, refer to their Web site at `www.fieldbus.org`.

Uses of Fieldbus

Foundation Fieldbus is used in process control and monitoring. Process control refers to continuous processes like flow control, temperature control, and tank level control. These types of processes are typically found in places like oil refineries, chemical plants, and paper mills. Foundation Fieldbus can also be used for monitoring over long distances.

Foundation Fieldbus implements distributed control, which means that control is done by the devices instead of by a monitoring computer. Input, output, and process control devices configured on a Fieldbus network can run independently of a computer system.

National Instruments Foundation Fieldbus products can be used to create a Foundation Fieldbus system. For example, National Instruments does not make valves or pressure transmitters, but with National Instruments tools, you can make a valve and pressure transmitter from third parties work together over a Foundation Fieldbus network to implement level control in a tank. You can use National Instruments Lookout with National Instruments Foundation Fieldbus boards to develop the Human Machine Interface (HMI) for the resulting control system. With the NI-FBUS Communications Manger or NI-FBUS Configurator software, you can also use OPC to connect to third-party software packages that support OPC.

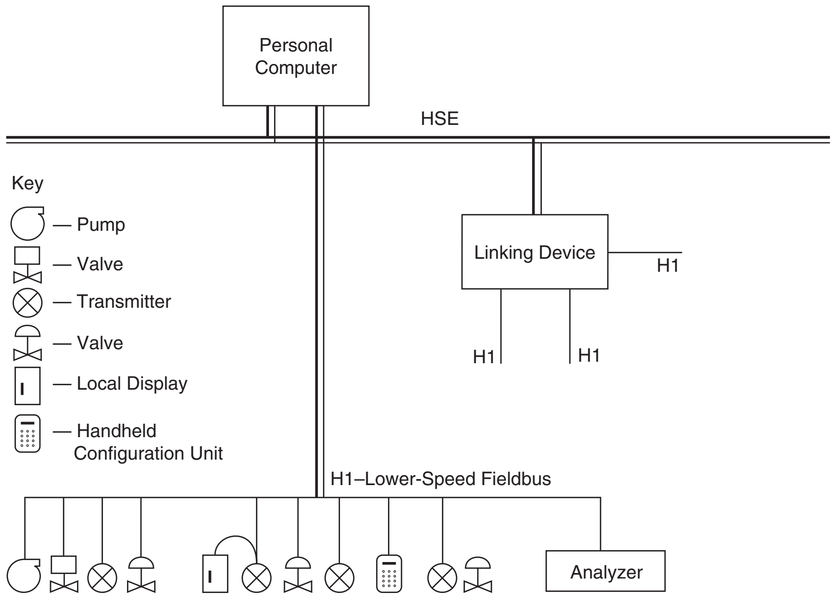
The Fieldbus Network

Foundation Fieldbus is an all-digital, two-way, multi-drop communication system that brings the control algorithms into instrumentation. Foundation Fieldbus is a Local Area Network (LAN) for Foundation Fieldbus devices including process control sensors, actuators, and control devices. Foundation Fieldbus supports digital encoding of data and many types of messages.

Unlike many traditional systems which require a set of wires for each device, multiple Foundation Fieldbus devices can be connected to the same set of wires.

Foundation Fieldbus overcomes some of the disadvantages of proprietary networks by offering a standardized network to connect systems and devices. A simple Fieldbus network setup is shown in Figure 1-1.

Figure 1-1. Foundation Fieldbus-Based Control System



Foundation Fieldbus Communication Protocols

Foundation Fieldbus has two communication protocols: H1 and HSE. The first, H1, transmits at 31.25 Kb/s and is used to connect the field devices. The second protocol, High Speed Ethernet (HSE), uses 10 or 100 Mbps Ethernet as the physical layer and provides a high-speed backbone for the network.

H1 is an all digital, serial, two-way communication system running at 31.25 kbit/s which interconnects field equipment such as sensors, actuator, and controllers. H1 is a Local Area Network (LAN) for instruments used in both process and manufacturing automation with built-in capability to distribute the control application across the network.

HSE is based on 10/100 Mbps standard Ethernet/IP/TCP/UDP protocols and supports the same functions as H1, but at a much higher bandwidth (10/100 Mbps). Its large capacity to move data, along with the inherent Foundation Fieldbus functionality, and publish/subscribe access, fits in with plant-wide integration in the process industries.

Foundation Fieldbus networks may be composed of one or more of these interconnected segments. HSE subnetworks can use a variety of commercially available interconnection devices such as hubs, switches, bridges, routers, and firewalls. H1 links are interconnected physically only by Foundation Fieldbus H1 Data Link bridges. HSE to H1 interconnections are performed by Foundation Fieldbus Linking Devices.

A typical network topology has HSE connections between computers, and runs slower H1 links (31.25 Kbps) between the devices themselves. Devices designed for HSE can be connected to HSE directly. Most devices are designed to use one protocol or the other.

H1 and HSE were specifically designed as complementary networks. H1 is optimized for traditional process control applications, while HSE, which employs low-cost, Commercial Off-the-shelf (COTS) Ethernet equipment, is designed for high-performance control applications and plant information integration. The combined H1/HSE Fieldbus solution allows for full integration of basic and advanced process control, and hybrid/batch/discrete control subsystems, with higher level, supervisory applications. H1/HSE provides the key to optimum enterprise performance by removing unneeded I/O conversion equipment and controllers, sensor networks, and gateways. This flat, integrated architecture provides increased plant uptime (through improved diagnostics and operator information), increased performance (COTS Ethernet), and reduced costs (COTS and less overall equipment).

Foundation Fieldbus Concepts

This section discusses basic concepts of H1 Fieldbus architecture.

H1 Concepts

There are six conceptual parts to a Fieldbus network: links, devices, blocks and parameters, linkages, loops, and schedules.

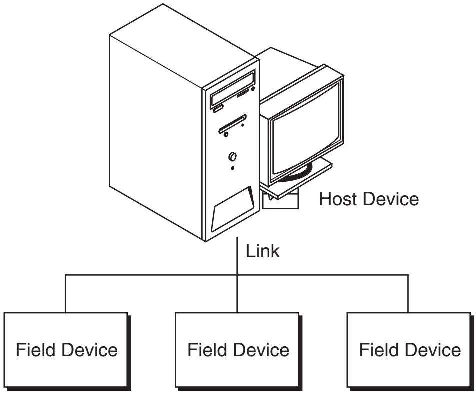
Links

A Foundation Fieldbus network is made up of devices connected by a serial bus. This serial bus is called a *link* (also known as a segment). A Fieldbus network consists of one or more links. Each link is configured with a unique link identifier.

Each link on the Fieldbus network connects physical devices. The devices can be field devices (temperature transmitters, valves, and so on) or host devices (PCs, distributed control systems). Each physical device is configured with a physical device tag, an address, and a device ID. The physical device tag must be unique within a Fieldbus system, and the address must be unique within each link. The device manufacturer assigns a device ID that is unique to the device.

Figure 1-2 shows a link in a Fieldbus H1 network.

Figure 1-2. Fieldbus Network Link



Devices

Devices are uniquely identified on the Fieldbus network by a character string name, or tag. The *device tag* is a configurable attribute of the device that usually describes the type of the device. Device tags are unique to each device on a Fieldbus network.

Another unique identifier of a device is the *device ID*, which includes a serial number unique to the device. The device ID is assigned by the device manufacturer; you cannot configure it.

There are three types of devices on a Fieldbus H1 network: *link masters*, *basic devices*, and *H1 bridges*.

- Link master—A link master device is capable of controlling the communications traffic on a link by scheduling the communication on the network. Every Fieldbus network needs at least one link master-capable device. A link master can be an interface board in a PC, a distributed control system, or any other device, such as a valve or a pressure transducer. Link masters need not be separate devices; they can have I/O functionality (for example, you could buy temperature transmitters both with and without link master capability). The National Instruments Foundation Fieldbus interface boards are link master devices.

  Fieldbus can operate independently of a computer system because of link masters on the bus. Link masters have processing capability and are capable of controlling the bus. After you download a configuration to your device(s), your control loop can continue to operate—even if the monitoring computer is disconnected.

  All of the link masters receive the same information at the time of download, but only one link master will actively control the bus at a given time. The link master that is currently controlling the bus is called the Link Active Scheduler (LAS). If the current Link Active Scheduler fails, the next link master will take over transparently and begin controlling bus communications where the previous Link Active Scheduler left off. Thus, no special configuration is required to implement redundancy.

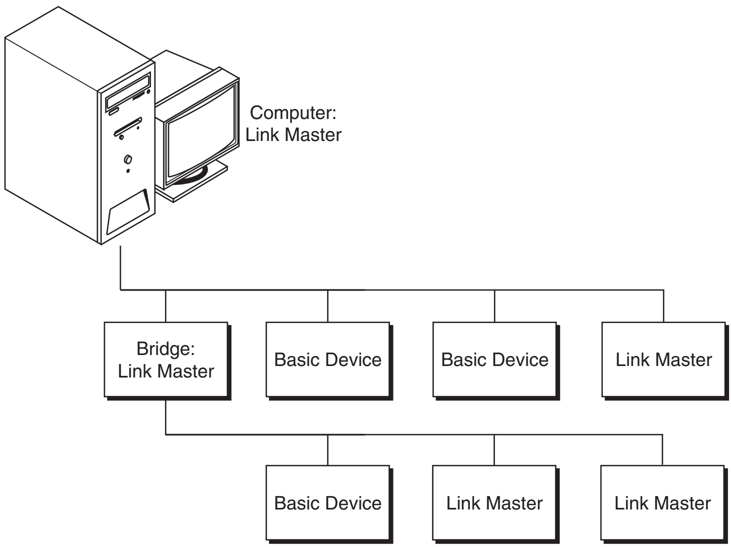
The Link Active Scheduler device follows the schedule downloaded to it and the other link masters during the configuration process. At the appropriate times, it sends commands to other devices, telling them when to broadcast data. The Link Active Scheduler also publishes time information and grants permission to devices to allow them to broadcast unscheduled (acyclic) messages, such as alarms and events, maintenance and diagnostic information, program invocation, permissives and interlocks, display and trend information, and configuration.

- Basic device—A basic device is a device which is not capable of scheduling communication. Basic devices cannot become the Link Active Scheduler.
- H1 bridge—Bridge devices connect links together into a spanning tree. They are always link master devices and they must be the Link Active Scheduler. An H1 bridge is a device connected to multiple H1 links whose data link layer performs forwarding and republishing between and among the links.

Note Be aware of the difference between a bridge and a gateway. While a bridge connects networks of different speeds and/or physical layers, a gateway connects networks that use different communications protocols.

Figure 1-3 shows these three types of devices.

Figure 1-3. Fieldbus Network Devices



Blocks and Parameters

Blocks and parameters are described in the *Blocks* section of Chapter 2, *Foundation Fieldbus Technology and Layers*.

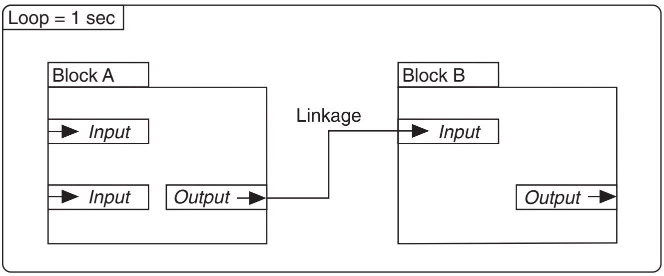
Linkages

The function blocks configured to control a process are linked, or connected by configuration objects inside the devices. These *linkages* allow you to send data from one block to another. A linkage is different from a link, in that a link is a physical wire pair that connects devices on a Fieldbus network, and a linkage is a logical connection that connects two function blocks. For more information on linkages, refer to the *Objects* section of Chapter 2, *Foundation Fieldbus Technology and Layers*.

Loops

A *loop* (or control loop) is a group of function blocks connected by linkages executing at a configured rate. Each block executes at the configured rate and data moves across the linkages between the blocks at the configured rate. Figure 1-4 shows an example of a control loop.

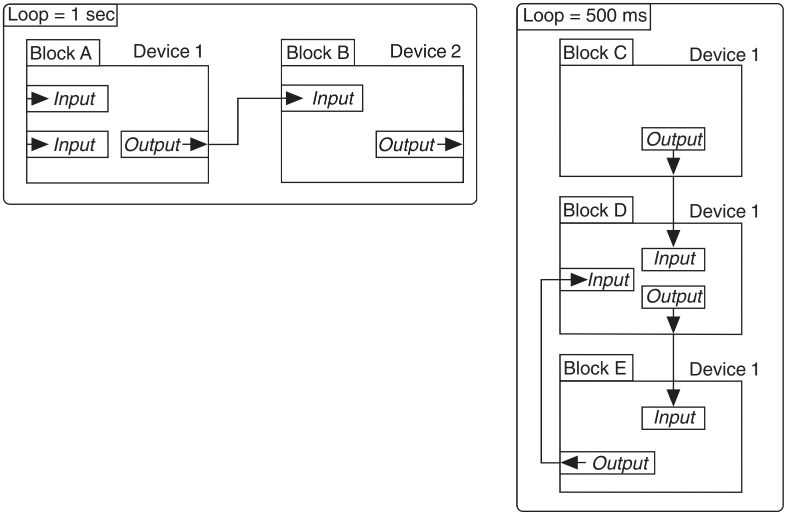
Figure 1-4. Control Loop



Multiple Loops

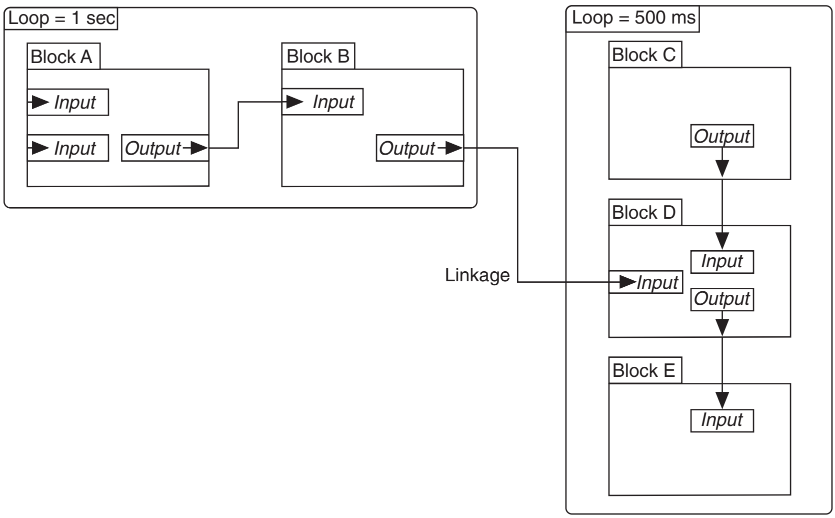
It is possible to have multiple loops running at different rates on a link. Figure 1-5 shows an example of multiple loops.

Figure 1-5. Multiple Loops Running At Different Rates



Even if loops are running at different rates, they can send each other data through linkages. Figure 1-6 shows an example of a linkage between two loops. All loops on a link run within one macrocycle. A *macrocycle* is the least common multiple of all the loop times on a given link. For example, the macrocycle in Figure 1-6 is 1 second.

Figure 1-6. Linkage Between Two Loops



HSE Concepts

This section discusses the concepts of HSE architecture. There are four conceptual parts to an HSE network: HSE Device, HSE Field Device, Linking Device, and I/O Gateway Device.

HSE Device

An HSE Device is any Foundation Fieldbus device-type connected directly to HSE Media. All HSE devices contain an FDA Agent, an HSE SMK (System Management Kernel), and an HSE NMA (Network Management Agent) virtual field device (VFD). Examples include Linking Devices, I/O Gateway Devices, and HSE Field Devices.

HSE Field Device

An HSE Field Device is an HSE device that also contains at least one Function Block Application Process (FBAP).

Linking Device

Linking devices are HSE devices used to attach H1 links to the HSE network. They provide access between HSE devices and H1 devices and access between H1 devices interconnected by a HSE network. A linking device may also contain an H1 bridge that provides for H1 to H1 communications between bridged H1 links.

I/O Gateway Device

An I/O gateway device is an HSE device used to provide HSE access to non-Foundation Fieldbus devices via function blocks.

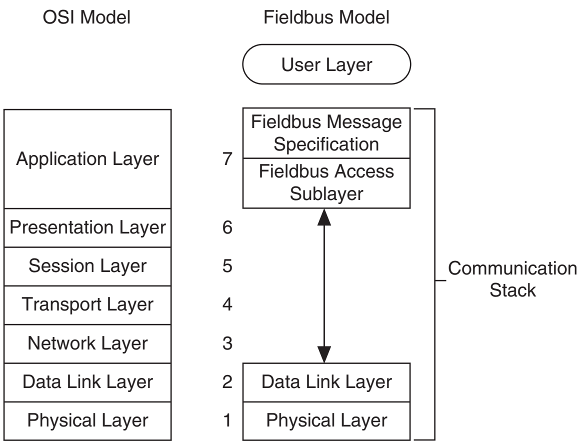
2

Foundation Fieldbus Technology and Layers

Note The first few sections of this chapter apply only to Foundation Fieldbus device development. If you are not developing a Foundation Fieldbus device, proceed to the *User Layer* section.

Foundation Fieldbus communication layers consists of the physical layer, the communication stack, and the user layer. Figure 2-1 shows a diagram of the Fieldbus layers compared to the Open Systems Interconnect (OSI) layered communication model. Notice that the OSI model does not define a user layer.

Figure 2-1. The Fieldbus Model and the OSI Seven-Layer Communications Model



Foundation Fieldbus does not implement layers three, four, five, and six of the OSI model because the services of these layers are not required in a process control application. A very important part of Foundation Fieldbus is the defined user layer, often referred to as layer eight.

Refer to the Foundation Fieldbus specifications for complete information about the layers of the Foundation Fieldbus network.

Physical Layer

The physical layer converts digital Fieldbus messages from the communication stack to physical signals on the Fieldbus transmission medium and vice versa. Refer to the *Foundation Specification: 31.25 kbit/s Physical Layer Profile* for the Foundation Fieldbus H1 physical layer specifications. The physical layer implements IEC Standard 1158-2 and ISA Standard ISA S50.02. Refer to the *Foundation Specification: HSE Presence* for the Foundation Fieldbus HSE physical layer specifications.

Communication Stack

The communication stack performs the services required to interface the user layer to the physical layer. The communication stack consists of three layers: the Fieldbus Message Specification, the Fieldbus Access Sublayer, and the Data Link Layer. The communication stack encodes and decodes user layer messages and ensures efficient and accurate message transfer.

The Data Link Layer manages access to the Fieldbus through the Link Active Scheduler by splitting data into frames to send on the physical layer, receiving acknowledgment frames, and re-transmitting frames if they are not received correctly. It also performs error checking to maintain a sound virtual channel to the next layer.

The Fieldbus Access Sublayer provides an interface between the Data Link Layer and the Fieldbus Message Specification layer. The Fieldbus Access Sublayer provides communication services such as client/server, publisher/subscriber, and event distribution.

The Fieldbus Messaging Specification layer defines a model for applications to interact over the Fieldbus. The object dictionary and the virtual field device are important in this model. The object dictionary is a structure in a Fieldbus device that describes data that can be communicated on the Fieldbus. You can think of the object dictionary as a lookup table that gives information about a value (such as data type) that can be read from or written to a device. The virtual field device is a model for remotely viewing data described in the object dictionary. The services provided by the Fieldbus Messaging Specification allow you to read and write information about the object dictionary, read and write the data variables described in the object dictionary, and perform other activities such as uploading/downloading data and invoking programs inside a device.

Within the Fieldbus Messaging Specification layer are two management layers called System Management and Network Management. System Management assigns addresses and physical device tags, maintains the function block schedule for the function blocks in that device, and distributes application time. You can also locate a device or a function block tag through System Management.

Network Management contains objects that other layers of the communication stack use, such as data link, Fieldbus Access Sublayer, and Fieldbus Messaging Specification. You can read and write System Management and Network Management objects over the Fieldbus using the FMS Read and FMS Write services.

Virtual Field Devices

The virtual field device (VFD) is a model for remotely viewing data described in the object dictionary. The services provided by the Fieldbus Messaging Specification allow you to read and write information about the object dictionary, read and write the data variables described in the object dictionary, and perform other activities such as uploading/downloading data and invoking programs inside a device.

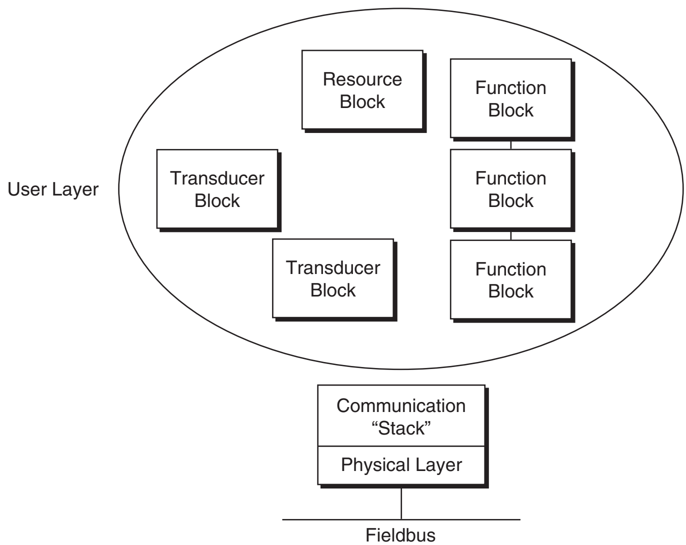
Each physical device on the Fieldbus can have one or more virtual field devices. A network configuration application can assign each virtual field device a tag that is unique within the device. Most devices have only one virtual field device. Each virtual field device has one resource block and one or more function blocks and transducer blocks. Each block should be assigned a tag that is unique within the Fieldbus system.

Note This manual assumes each device contains only one virtual field device.

User Layer

The user layer provides the interface for user interaction with the system. The user layer uses the device description to tell the host system about device capabilities. The user layer defines blocks and objects that represent the functions and data available in a device. Rather than interfacing to a device through a set of commands, like most communication protocols, Foundation Fieldbus lets you interact with devices through a set of blocks and objects that define device capabilities in a standardized way. The user layer for one device consists of the resource block, and one or more transducer blocks and function blocks, as illustrated in Figure 2-2.

Figure 2-2. The User Layer



Device Descriptions

A key objective for Foundation Fieldbus is interoperability—the ability to build systems comprised of devices from a variety of manufacturers to take full advantage of both the standard and unique capabilities of every device.

Instead of requiring that device manufacturers use only a given set of functions in a device to ensure that a system can always communicate with a new device, Foundation Fieldbus uses device descriptions, which describe all the functions in a device. They allow manufacturers to add features beyond the standard Foundation Fieldbus interface without fearing loss of interoperability. The device vendor supplies device description files, which describe the parameters of the function and transducer blocks contained in a device. The device description also defines attributes of parameters and blocks (such as names and help strings), ranges of values for parameters, functional information (such as menus and methods that you can use with the device), and so on. At the device manufacturer's discretion, names and help strings can even be provided in multiple languages. The language can be set in the configuration software. Using the device description, the host in a control system can obtain the information needed to create an interface that configures parameters, calibrates, performs diagnostics, and accomplishes other functions on the device.

The names of device description files are numbers, and they have `.ffo` and `.sym` file extensions.

Blocks

Blocks can be thought of as processing units. They can have inputs, settings to adjust behavior, and an algorithm which they run to produce outputs. They also know how to communicate with other blocks. The three types of blocks are the resource block, transducer block, and function block.

Resource Block

A resource block specifies the general characteristics of the resource (or block). This includes the device type and revision, manufacturer ID, serial number, and resource state. Each device has only one resource block. The resource block also contains the state of all of the other blocks in the device.

The resource block must be in automatic mode for any other blocks in the device to execute. The resource block is a good place to start troubleshooting if the device is not behaving as desired. It has diagnostic parameters that help you determine the cause of problems.

Transducer Blocks

Transducer blocks read from physical sensors into function blocks. Transducer blocks decouple the function blocks from the hardware details of a given device, allowing generic indication of function block input and output.

The transducer block knows the details of I/O devices and how to actually read the sensor or change the actuator. The transducer block performs the digitizing, filtering, and scaling conversions needed to provide the sensor value to the function blocks and/or makes the change in the output as dictated by the function block. Generally, there will be one transducer block per device channel. In some devices, multiplexors allow multiple channels to be associated with one transducer block.

Manufacturers can define their own transducer blocks. For some devices, the functionality of the transducer block is included in the function block. You will see no separate transducer blocks for such devices.

Note There are many parameters that can be changed to modify the I/O functionality.

Function Blocks

Function blocks provide the control and I/O behavior.

Usually, a device has a set of functions it can perform. These functions are represented as function blocks within the device. A function block can be thought of as a processing unit. Function blocks are used as building blocks in defining the monitoring and control application.

The Foundation Fieldbus specification *Function Block Application Process* defines a standard set of function blocks, including 10 for basic control and 19 for advanced control. Table 2-1 shows the 10 function blocks for the most basic control and I/O functions.

Table 2-1. Ten Standard Foundation Fieldbus-Defined Function Blocks

Function Block Name	Symbol
Analog Input	AI
Analog Output	AO
Bias/Gain	BG
Control Selector	CS
Discrete Input	DI
Discrete Output	DO
Manual Loader	ML
Proportional/Derivative	PD
Proportional/Integral/Derivative	PID
Ratio	RA

Not all devices contain all 10 standard function blocks. Additionally, manufacturers can also define their own function blocks. A device description file provided with each device tells the configuration software about added function blocks. Thus, manufacturer-defined function blocks are as easy to use as the standard function blocks.

Different function blocks do different things. There can be many function blocks present in a device at one time. Function blocks are stored in the device memory. Some devices come with specific function blocks pre-loaded into memory. They cannot be deleted, nor can new function blocks be added. Other devices allow function blocks to be instantiated (created) and deleted as necessary.

Note Function blocks have a wide variety of parameters that can be changed to control their operation. Refer to the *Function Block Parameters* section for more information.

You can connect the input and output of individual function blocks to specify communication of data between blocks.

In general, the function blocks will be a layer of abstraction above the physical I/O channels. Function blocks "talk" with transducer blocks, which deal with the details of I/O.

AI (Analog Input)

The AI block reads data from a single analog input channel. This block performs simple filtering and scaling of the raw data to engineering units from the input channel and supports limit alarming.

AO (Analog Output)

The AO block writes data to an analog output channel. This block supports cascade initialization to allow upstream control blocks to switch smoothly from manual to automatic mode. It also has a faultstate behavior that allows the block to react if communications fail between itself and the upstream block.

PID (Proportional–Integral–Derivative)

The PID block implements a PID control algorithm. In Fieldbus, a PID block must be connected to an upstream block (such as an AI block) and a downstream block (such as an AO block) before it can be used for control. These software connections are established by using host Fieldbus configuration software, such as the NI-FBUS Configurator.

DI (Discrete Input)

The DI block reads data from discrete input channels. This block performs simple filtering and processing of the raw data from the input channel and supports limit alarming.

DO (Discrete Output)

The DO block writes to a discrete output channel. This block supports cascade initialization to allow upstream control blocks to determine the current state of the process before assuming control. It also has a faultstate behavior that allows the block to react if communications fail between itself and the upstream block.

Function Block Parameters

You can change the behavior of a block by changing the settings of its parameters.

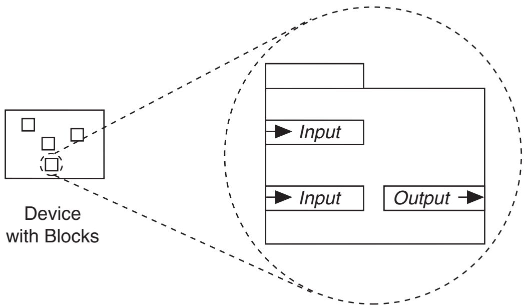
Function block parameters are classified as follows:

- *Input* parameters receive data from another block.
- *Output* parameters send data to another block.
- *Contained* parameters do not receive or send data; they are contained within the block.

Some parameters contain multiple settings called fields. For example, a common output parameter for many function blocks is `OUT`. `OUT`, however, is not just a value. It also contains information about the status of that value—whether it is good or questionable, for example. Thus, the `OUT` parameter actually consists of two fields, `VALUE` and `STATUS`. In this documentation set, the fields of a parameter will be indicated as the *`ParameterName.FieldName`*, for example, `OUT.VALUE`. The `OUT` parameter (including both `VALUE` and `STATUS` fields) is generated by the block at the end of the block execution. This parameter is published on the bus if it is required as the input for another function block in a different device. If the parameter is only required by other function blocks in the same device, it will not be needlessly published on the bus. Refer to the *Scheduled Communication on the Bus* section of Chapter 3, *Fieldbus Communication*, for more information on publishing across the bus.

Figure 2-3 shows an example of a function block and its input and output parameters that are available to other blocks.

Figure 2-3. Function Block Parameters



Note There are different input and output parameters for different uses, such as cascade, remote, and so on.

An output parameter can be linked to an input parameter of another function block. Like an output parameter, an input parameter contains fields VALUE and STATUS. The IN.STATUS is inherited from the linked output parameter.

Contained parameters cannot be linked to an output or input parameter. Instead, they are either set internally or set by an operator. Examples of contained parameters are scaling parameters, alarm limits, I/O options, and error codes.

Function Block Applications

The function block application is like a program that you download to your device for it to execute. How you create a function block application depends on the configuration software you are using. In the NI-FBUS Configurator, the inputs and outputs of the function blocks are wired together in graphical format to create the control strategy. The NI-FBUS Configurator also automatically creates an execution schedule for the function blocks. You then download the entire configuration to the device, including the function block application and schedule. When this is done, the device can begin executing the function blocks according to the schedule.

Function Block Modes

One of the more confusing aspects of Fieldbus for new users is the concept of function block modes—what the modes are and how to get a block into the desired mode.

The most common modes are Automatic (Auto), Cascade (Cas), Manual (Man), and Out Of Service (OOS).

- In Automatic mode, the block is using a local setpoint value in the normal block algorithm to determine the output value.
- In Cascade mode, the block is receiving its setpoint value from another function block for use in the normal block algorithm to determine the output value.
- In Out Of Service mode, the block is not running at all. Normally, this mode is used during block configuration. Also, some devices require that the function block be in Out Of Service mode when changing certain parameters.
- In Manual mode, the block output is not being calculated by the normal block algorithm. The operator writes the output of the block directly.

The Function Block MODE_BLK Parameter

The MODE_BLK parameter for a function block is the parameter that contains information on the modes of the block. It has four fields: TARGET, ACTUAL, PERMITTED, and NORMAL. PERMITTED and NORMAL are defined by the device manufacturer.

- PERMITTED contains a list of all allowable modes for that block.
- NORMAL is the mode the device manufacturer expects the block to be in during normal use.
- ACTUAL is the current operating mode of the function block on the device.
- TARGET is a field that is writable by the user. Writing this field tells the device to change to the specified mode. The device will attempt to change the mode. If it is successful, the ACTUAL field changes to reflect the TARGET field.

Note If the block is not currently scheduled, it will always be in Out Of Service mode. In this case, writing the `TARGET` field will not change the `ACTUAL` field. After it attempts to change the block from Out Of Service to the specified mode and fails, the `TARGET` field will return to Out Of Service.

Objects

In addition to the blocks described, the user layer of a Foundation Fieldbus device contains other types of objects. Objects are defined by the Foundation Fieldbus specification as part of the function block application structure. Some of them are presented here so that you are aware of the features they offer. If you are using a configuration software package, such as the NI-FBUS Configurator, you will not need to deal with them directly.

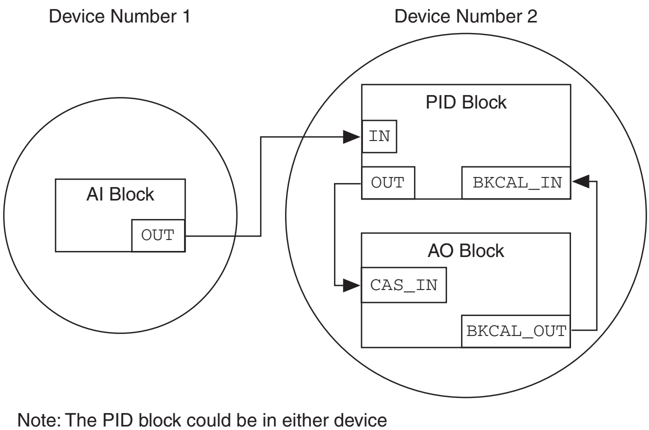
- Linkage object—Linkage objects define the connections between the outputs of one block and inputs of another, whether the blocks are in the same device or different devices.

Note If you are using the NI-FBUS Configurator, the linkage object is created implicitly when you wire the output of one function block to the input of another. The PID and AO are examples of function blocks that often receive their setpoints from an upstream block.

In Figure 2-4, there are three linkages:

- The AI block `OUT` parameter to the PID block `IN` parameter.
- The PID block `OUT` parameter to the AO block `CAS_IN` parameter.
- The AO block `BKCAL_OUT` parameter to the PID block `BKCAL_IN` parameter.

Figure 2-4. Linkages



Parameters are sent across the bus when a linkage connects an output parameter from a function block in one device to an input parameter of a function block in another device. A block parameter in the device that is writing to the bus is referred to as a *published* *parameter*. A block parameter in the device that is receiving data from the bus is referred to as a *subscribed parameter*. In Figure 2-4, the `OUT` parameter of the AI block is a published parameter, and the `IN` parameter of the PID block is a subscribed parameter.

- View object—View objects allow efficient communication of common groups of parameters. This provides easy access to the parameters for HMI packages such as Lookout or LabVIEW DSC. View objects ease tasks such as loop tuning and configuration changes. There are four standard views defined by the Foundation Fieldbus function block specification for each type of block. The specification includes which parameters are contained in each view. The view objects are especially useful if you are using the NI-FBUS Communications Manager API to write your programs.
  – `VIEW_1` contains the main dynamic parameters.
  – `VIEW_2` contains the main static parameters relevant to the process.
  – `VIEW_3` contains the parameters from `VIEW_1` and additional dynamic parameters.
  – `VIEW_4` contains other static parameters, including configuration and maintenance parameters.
- Alert object—Alert objects allow a device to report alarms and events over the bus. Alert objects are fully configurable by the user.
- Trend object—Trend objects accumulate values of function block parameters for access over the network and publish historical data for HMI trending and storage. They include the parameter value, status, and a timestamp. Multiple parameters in the same block can be trended. Other devices or hosts can make use of the information accumulated by the trend object.

3 Fieldbus Communication

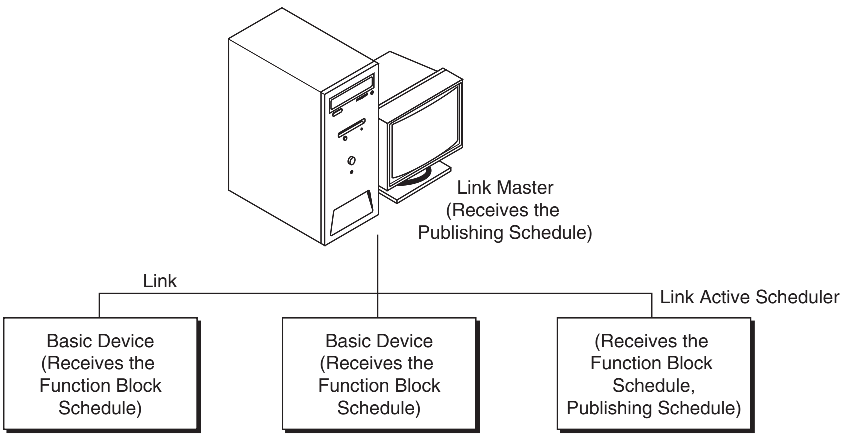
Scheduled Communication on the Bus

A device that sends data to other devices is called a publisher. A device that needs data from a publisher is called a subscriber. The Link Active Scheduler uses the publisher/subscriber information to tell the publisher devices when to transmit their data over the bus. These scheduled data transfers are done in a deterministic manner (meaning that data is transferred between a given set of devices at the same time during each loop iteration), following the schedule developed in your configuration software and downloaded to the link masters.

The schedule can be divided into two parts: a function block schedule that determines when a block executes, and a publishing schedule that determines when data parameters are published over the Fieldbus. The function block schedule is downloaded to the particular device that contains each function block, and the publishing schedule is downloaded to a device or devices that have link master capability. As discussed earlier, the link master currently executing the publishing schedule and thus controlling the process is the Link Active Scheduler.

Figure 3-1 shows the relationship of the function block schedule and the publishing schedule to the device, link, link master, and Link Active Scheduler. Notice that the PC does not receive a function block schedule because it has no function blocks.

Figure 3-1. Publishing and Function Block Schedules Relationships



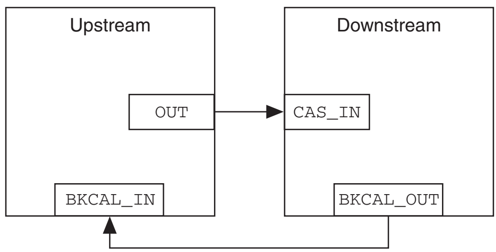
Cascading Blocks

Setting up a control strategy requires communication between function blocks. For example, in a PID loop you would need an analog input block, a PID control block, and an analog output block. The data from the analog input block must be fed to the PID control block. Additionally, data must pass between the PID control block and the analog output block. Most often, these connections would be made in cascade mode.

Parameter Connections for Cascade Mode

A cascade connection exists between two blocks: an upstream controlling block, and a downstream controlled block. In a PID loop, the upstream block is the PID block, and the downstream block is the AO block. In the case of cascaded PID blocks, the upstream PID feeds a setpoint into a second PID that is acting as the downstream block. In both cases, the parameter connections are the same. The output (`OUT`) parameter of the upstream block is connected to the cascade input (`CAS_IN`) parameter of the downstream block. This connection controls the setpoint of the downstream block. To allow the upstream block to determine the current setpoint of the downstream block, you must also connect the backward calculation output (`BKCAL_OUT`) parameter of the downstream block with the backward calculation input (`BKCAL_IN`) of the upstream block. The connections are shown in Figure 3-2.

Figure 3-2. Cascade Connections



4

Wiring and Connections

Foundation Fieldbus wiring is similar to that used in traditional 4-20 mA systems. Fieldbus uses existing 4-20 mA (twisted-pair) wiring to carry multiple signals and power. It also uses existing terminal blocks. Like traditional 4-20 mA systems, Fieldbus provides for intrinsic safety requirements that allow you to use Fieldbus in hazardous areas.

Note Not all devices support bus-powered operation. These devices will need their own power connections in addition to the two Fieldbus connections.

The Fieldbus system provides for power supply redundancy and enables loop-powered instruments. Unlike traditional 4-20 mA systems, Fieldbus allows you to connect field devices in parallel.

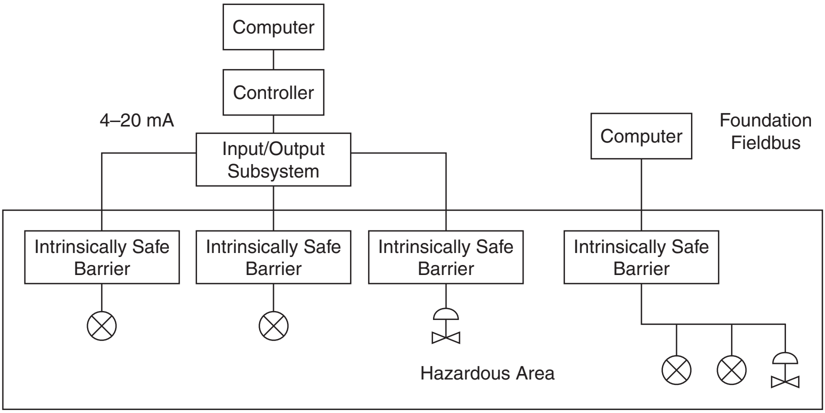
Table 4-1 describes the physical characteristics of an H1 Foundation Fieldbus network.

Table 4-1. Physical Overview of H1 Foundation Fieldbus

Data Rate	31.25 Kbytes/s
Topology	chickenfoot/crowsfoot, daisychain/spur, star
Bus Powering Possible?	Yes, for devices that support it
Intrinsically Safe (IS) Operation Possible?	Yes, for devices that support it
Maximum Possible Number of Devices Per Link Without Repeaters	32
Maximum Possible Cable Length Per Link Without Repeaters	1900 m
Maximum Possible Spur Length	120 m

Figure 4-1 compares the wiring and hardware requirements of Foundation Fieldbus with those of traditional 4-20 mA systems. Notice that the traditional system requires one intrinsic safety barrier and a set of wires for each device, for a total of three barriers and three sets of wires. The Foundation Fieldbus implementation requires only one intrinsic safety barrier and one set of wires for multiple devices.

Figure 4-1. Intrinsic Safety in a Foundation Fieldbus System



Refer to *Wiring and Installation 31.25 kbit/s, Voltage Mode, Wire Medium* at the Fieldbus Foundation website (`www.fieldbus.org`) or refer to *Wiring Design and Installation Guide* at the Relcom, Inc. website (`www.relcominc.com/fieldbus`) for more information on the Fieldbus wiring specification.

Termination

You must terminate your wiring in compliance with the Foundation Fieldbus specification. Foundation Fieldbus requires a resistor and a capacitor in parallel. You must place termination appropriately on the network, which will depend on your network topology.

Refer to *Wiring and Installation 31.25 kbit/s, Voltage Mode, Wire Medium* at the Fieldbus Foundation website (`www.fieldbus.org`) or refer to *Wiring Design and Installation Guide* at the Relcom, Inc. website (`www.relcominc.com/fieldbus`) for more information on the Fieldbus termination specification.

Link Configuration

There are several factors you should consider when deciding how many devices to use on your H1 link.

- Link Topology—A Foundation Fieldbus link can include one or more links joined by repeaters. Physically, one port on a National Instruments interface board connects to one link. Each individual link can be up to 1,900 m and can have up to 32 devices connected. Up to five individual links can be joined by four repeaters to make one large aggregate link. This gives a maximum limit of about 155 devices per interface. The connection to one interface port on the interface board is viewed as one link by the NI-FBUS Configurator, regardless of how many links might actually be joined by repeaters (important for software licensing of the NI-FBUS Configurator). Refer to the *Choosing Configuration Software* section of Chapter 5, *National Instruments Fieldbus Products*, for information on ordering a link upgrade. National Instruments offers boards with one or two interfaces (ports).

Note Maximum Fieldbus link configurations (1,900 m and 32 devices per link without repeaters) cannot be achieved under many circumstances. Refer to the *Link Configuration* section for information on setting up your Fieldbus link.

- Bandwidth—A link can carry about 30 scheduled messages per second. So, you could have three devices, each sending ten messages per second, or 120 devices (connected by repeaters), each sending one message every four seconds.
- Power—Using bus powering reduces the number of devices that can be on one link. Additionally, the power supply on each link must be sufficiently large. Some devices are separately powered and thus do not pull power from the link's power supply.
- Wire Resistance—Existing wire in older plants often has high resistance that can reduce the maximum link length.
- Barriers/Isolators—You should limit the power available in hazardous areas.

Table 4-2 illustrates the effect of intrinsically safe and bus-powered operation on the typical number of devices you can place on one link.

Table 4-2. Typical Number of Devices per Link

Bus Setup	Typical Number of Devices (one link without repeaters)
Not Intrinsically Safe, Not Bus Powered	2-32
Intrinsically Safe, Bus Powered	2-6 (of which 1-4 are in the IS area)
Not Intrinsically Safe, Bus Powered	2-12

5

National Instruments Fieldbus Products

National Instruments provides both software and hardware for Foundation Fieldbus.

Hardware Products

National Instruments provides interface boards for the PCI bus (PCI-FBUS), PCMCIA (PCMCIA-FBUS), and USB (USB-8486). Each National Instruments board comes with the NI-FBUS Communications Manager software, which the board uses to communicate over the bus.

The FBUS-HSE/H1 Linking Device connects an HSE network to two H1 segments.

Software Products

The NI-FBUS Communications Manager software is used by the interface boards to communicate over the bus. Additionally, it provides a high-level API advanced users can use to interface with the National Instruments Foundation Fieldbus communication stack and hardware.

Most users use the NI-FBUS Configurator. In addition to providing the functionality of the NI-FBUS Communications Manager in a graphical format, it includes additional functionality to allow you to configure a Fieldbus network. It can automatically generate the schedule for the network and configure field devices and hosts to transmit and receive alarms and trends.

The NI-FBUS Monitor helps you monitor and debug Fieldbus data traffic. It symbolically decodes data packets from the Fieldbus, monitors the live list, and performs statistical analysis of packets. You can use the NI-FBUS Monitor to debug the development of device and host applications.

You can use Foundation Fieldbus products with several National Instruments HMI software packages, including Lookout and LabVIEW DSC. Communication with Lookout is handled directly by NIFB.EXE (the NI-FBUS process). For LabVIEW DSC and other OPC client programs, the Fieldbus OPC server (included in NI-FBUS software) is used for communication.

Choosing Configuration Software

Foundation Fieldbus links must be configured. Only with configuration software can you do things like set device addresses, clear devices, and download (necessary if you are setting up a system for the first time). Your configuration software must match the interface board your computer is using to connect to the bus. National Instruments offers the NI-FBUS Configurator with a National Instruments Foundation Fieldbus interface board to let you configure your Fieldbus links.

Interface boards and configuration software are sold separately because multiple interface boards can reside in the same computer. The NI-FBUS Configurator, by default, is licensed for use on one machine, with up to four Fieldbus links (segments). If you have more than four links (ports on the boards), you need to purchase a link upgrade for each link in excess of four. To do so, download the order form from `ni.com/fieldbus`.

Note The connection to one interface port on the interface board is viewed as one link by the NI-FBUS Configurator, regardless of how many links might actually be joined by repeaters (important for software licensing of the NI-FBUS Configurator).

If you only want to use the board to read and write values (not configure) and you are a programmer, you can make calls to the API in the NI-FBUS Communications Manager and would not need the NI-FBUS Configurator. This would be possible only if you have already used third-party hardware and configuration software to configure the link.

Choosing an HMI Package

National Instruments offers a choice in industrial automation software. Both Lookout and LabVIEW DSC provide flexible environments in which you can add innovative features to your application; however, in some application areas, one or the other might be the best solution for your application.

Common Features of Both LabVIEW DSC and Lookout

Both LabVIEW DSC and Lookout provide the following functionality:

- OLE for Process Control (OPC) server and client capability.
- Driver support for Programmable Logic Controller (PLCs) connected independently of the Fieldbus.
- Access and control of your process through a Web browser.
- Alarming and trending.
- High-speed data logging.
- Continuous, batch, and discrete process control.

Typical LabVIEW DSC Applications

A typical LabVIEW DSC application has the following features:

- Ideal for users familiar with LabVIEW who want to quickly develop an industrial automation application.
- Ideal for applications that involve motion control, high-speed data acquisition, or image acquisition.
- Ideal for applications that involve complex analysis. LabVIEW includes a vast set of powerful virtual instrument (VI) libraries that can be used for sophisticated calculations, analysis, or control.

Typical Lookout Applications

A typical Lookout application has the following features:

- An easy-to-use, no-programming interface. This is ideal for a large percentage of industrial automation applications where monitoring and supervisory control is required, but advanced analysis is not.
- Online configuration to speed up development and debugging without taking down the process.
- Networking makes the development of client/server and multi-server applications straightforward.
- Built-in network security.
- Internal database makes it easy to share historical data, including alarm and event data, over the network.
- Native drivers are included for the most popular I/O devices and remote telemetry units (RTUs). Configuration of these drivers is built-in.

Using Both LabVIEW DSC and Lookout in an Application

If you are interested in features from both packages, then you can use both LabVIEW DSC and Lookout in your application. A typical scenario for this would be when you want an easily-configurable interface to a remote terminal unit with the capability of online configuration, but would also like to have some extensive analysis or interface with image acquisition or motion control. You could use Lookout for the remote terminal unit connectivity and LabVIEW DSC for the analysis. Lookout and LabVIEW DSC are both OPC-server and OPC-client capable. You could use the OPC connectivity to exchange data between LabVIEW DSC and Lookout.

A

Technical Support and Professional Services

Log in to your National Instruments `ni.com` User Profile to get personalized access to your services. Visit the following sections of `ni.com` for technical support and professional services:

- **Support**—Technical support at `ni.com/support` includes the following resources:
  - **Self-Help Technical Resources**—For answers and solutions, visit `ni.com/support` for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at `ni.com/forums`. NI Applications Engineers make sure every question submitted online receives an answer.
  - **Standard Service Program Membership**—This program entitles members to direct access to NI Applications Engineers via phone and email for one-to-one technical support, as well as exclusive access to self-paced online training modules at `ni.com/self-paced-training`. All customers automatically receive a one-year membership in the Standard Service Program (SSP) with the purchase of most software products and bundles including NI Developer Suite. NI also offers flexible extended contract options that guarantee your SSP benefits are available without interruption for as long as you need them. Visit `ni.com/ssp` for more information.

  For information about other technical support options in your area, visit `ni.com/services`, or contact your local office at `ni.com/contact`.
- **Training and Certification**—Visit `ni.com/training` for training and certification program information. You can also register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit `ni.com/alliance`.
- **Declaration of Conformity (DoC)**—A DoC is our claim of compliance with the Council of the European Communities using the manufacturer’s declaration of conformity. This system affords the user protection for electromagnetic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting `ni.com/certification`.
- **Calibration Certificate**—If your product supports calibration, you can obtain the calibration certificate for your product at `ni.com/calibration`.

You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office websites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

Numbers

4-20 mA system	Traditional control system in which a computer or control unit provides control for a network of devices controlled by 4–20 mA signals.

A

A	Amperes.
A/D	Analog-to-digital converter.
Abstract Syntax Notation 1	A formal syntax description language that defines the exact formatting of FMS messages.
AC	Alternating current.
Actuator	A device that translates electrical signals into mechanical actions.
Acyclic communication	Unscheduled communication on the bus.
Address	Character code that identifies a specific location (or series of locations) in memory.
Administrative Function	An NI-FBUS function that deals with administrative tasks, such as returning descriptors and closing descriptors.
AI	Analog Input.
Alarm	A notification the NI-FBUS Communications Manager software sends when it detects that a block leaves or returns to a particular state.
Alarm condition	A notification that a Fieldbus device sends to another Fieldbus device or interface when it leaves or returns to a particular state.
Alert	An alarm or event.
Alert function	A function that receives or acknowledges an alert.
Alert objects	Objects used for reporting alarms and events.
All dynamic	Information that is changing and may need to be referenced in a detailed display.

Analog	A description of a continuously variable signal or a circuit or device designed to handle such signals.
Analog network	A network that carries signals in analog form as a continuously varying range of electrical voltage or current.
AO	Analog Output.
API	*See* Application Programmer Interface.
Application	Function blocks.
Application clock	An internal clock on the Fieldbus usually set to the local time or Universal Coordinated Time. Application clock synchronization allows the devices to time stamp data throughout the Fieldbus network.
Application Programmer Interface	A message format that an application uses to communicate with another entity that provides services to it.
Argument	A value you pass in a function call. Sometimes referred to as a parameter, but this documentation uses a different meaning for parameter. *See also* Parameter.
Array	Ordered, indexed list of data elements of the same type.
ASCII	American Standard Code for Information Interchange.
ASN.1	Abstract Syntax Notation 1.
Asynchronous	Communication that occurs at times that are not predetermined.
Attenuation	Signal getting smaller as it travels on the cable.
Attribute	Properties of parameters.
AWG	American Wire Gauge.

B

b	Bits.
B	Bytes.
Backbone	Another term for bus, the main wire that connects nodes.
Bandwidth	The range of frequencies present in a signal, or the range of frequencies to which a measuring device can respond.
Bank	The combination of one FieldPoint network module and one or more terminal bases and I/O modules.

Basic device	A device that can communicate on the Fieldbus, but cannot become the LAS.
BG	Bias Gain.
Bipolar signaling device	A device that draws power from the cable for its internal operation and also draws an additional 10 milliamps that it "wastes".
Bit cell	The length of time taken by a single bit. This time is 32 microseconds for H1 Fieldbus.
Bit time	Time necessary to receive one bit.
Bitstring	A data type in the object description.
Block	A logical software unit that makes up one named copy of a block and the associated parameters its block type specifies. The values of the parameters persist from one invocation of the block to the next. It can be a resource block, transducer block, or function block residing within a virtual field device.
Block context	Describes a category of NI-FBUS functions that accept block descriptors.
Block level	The level of an NI-FBUS call that accepts a block descriptor.
Block tag	A character string name that uniquely identifies a block on a Fieldbus network.
Block view objects	Variable list objects used to read multiple block parameters at once.
Boolean	Logical relational system having two values, each the opposite of the other, such as true and false or zero and one.
Bridge	An interface in a Fieldbus network between two different protocols.
Buffer	Temporary storage for acquired or generated data.
Bus	The group of conductors that interconnect individual circuitry in a computer. Typically, a bus is the expansion vehicle to which I/O or other devices are connected. Examples of PC busses are the ISA and PCI buses.
Bus scheduler	*See* Link Active Scheduler.

C

C	Celsius.
Cable	A number of wires and shield in a single sheath.
Capacitance	The ability of a circuit (or device) to store electric energy in an electric field.
Capacitor	An electronic component that stores electrical charge. In a computer, capacitors are used in dynamic RAM cells and power supplies.
CCITT	International Telegraph and Telephone Consultative Committee.
CD	Compel Data.
CD schedule	The CD schedule contains a list of activities that are scheduled to occur an a cyclic basis.
Channel	A pin or wire lead to which you apply or from which you read the analog or digital signal.
Character string name	*See* Tag.
Chickenfoot	A common terminal block at a field junction box to which you can connect field devices.
Circuit	Interconnection of components to provide an electrical path between two or more components.
CISPR	International Special Committee On Radio Interference.
Client	A device that sends a request for communication on the bus.
Client/server VCR type	Used for queued, unscheduled, user-initiated, and one-to-one communication between devices on the Fieldbus.
Communication layer	*See* Communication stack.
Communication service	Provides a standardized way for user applications (such as function blocks) to communicate over the Fieldbus.
Communication stack	Performs the services required to interface the user application to the physical layer.
Configuration device	Device that generates all of the information needed to set up the Fieldbus.
Configuration object	*See* Linkage.

Connection Management	The service the NI-FBUS Communications Manager provides by handling Virtual Communication Relationships.
Contained parameter	A parameter that does not receive or send data and is contained within a function block.
Context management service	FMS service used to establish and release Virtual Communication Relationships with, and determine the status of, a virtual field device.
Control loop	A set of connections between blocks used to perform a control algorithm.
Control strategy	*See* Function Block Application.
Controller	An intelligent device (usually involving a CPU) that is capable of controlling other devices.
ControlNet	A 5 Mbit/sec communications protocol based on Producer/Consumer technology.
Core Function	The basic functions that the NI-FBUS Communications Manager software performs, such as reading and writing block parameters.
CPU	Central processing unit.
Crowsfoot	A common terminal block at a field junction box to which you can connect field devices.
CS	Control Selector.
Current	The flow of electrons through a conductor.
Cyclic	Closed-loop control.

D

D/A	Digital-to-analog converter.
Daisy-chain	A wiring method where a number of devices are attached along the homerun cable.
Data Link Layer	The second-lowest layer in the ISO seven-layer model (layer two). The Data Link Layer splits data into frames to send on the physical layer, receives acknowledgment frames, and re-transmits frames if they are not received correctly. It also performs error checking to maintain a sound virtual channel to the next layer.

Data link time synchronization	A time distribution message broadcast on the Fieldbus so that all devices have exactly the same data link time.
DataSocket	National Instruments programming tool for sharing live measurement data between applications separated by a network or the Internet.
dB	Decibel.
DC	Direct Current.
DC offset	The change in input voltage required to produce a zero output voltage when no signal is applied to an amplifier.
DCS	Distributed Control System.
DD	*See* Device Description.
DDL	*See* Device Description Language.
DDOD	Device Description Object Dictionary. The Device Description binary file.
DDS	*See* Device Description Service.
Descriptor	A number returned to the application by the NI-FBUS Communications Manager, used to specify a target for future NI-FBUS calls.
Deterministic communication	A synonym for scheduled/cyclic communication. Communication in which data is transferred between a given set of devices at the same time during each loop.
Device	A sensor, actuator, or control equipment attached to the Fieldbus.
Device address	A memory address that you use to access a device in a computer system.
Device Description	A machine-readable description of all the blocks and block parameters of a device.
Device Description Language	A formal programming language that defines the parameters of the blocks. It also defines attributes of parameters and blocks like help strings in different languages, ranges of values for parameters, and so on.
Device Description Service	A set of functions that applications use to access Device Descriptions.
Device ID	An identifier for a device that the manufacturer assigns. No two devices can have the same device ID.

Device level	An NI-FBUS call that accepts a physical device descriptor.
Device tag	A name you assign to a Fieldbus device.
DeviceNet	A low-cost industrial network to connect industrial devices such as limit switches, photoelectric cells, valve manifolds, motor starters, drives, and operator displays to PLCs and PCs.
DI	Discrete Input.
Digital	Pertaining to data (signals) in the form of discrete (separate/pulse form) integral values.
Directory	A structure for organizing files into convenient groups. A directory is like an address showing where files are located. A directory can contain files or subdirectories of files.
Distributed control	Process control distributed among several devices connected by network.
DLL	*See* Dynamic Link Library.
DMA	Direct Memory Access.
DO	Discrete Output.
DRAM	Dynamic Random Access Memory. Memory that requires electricity and refreshing to hold data.
Driver	Device driver software installed within the operating system.
Dynamic Link Library	A library of functions and subroutines that links to an application at run time.

E

E/P	Electric-to-pneumatic converter
EMI	Electromagnetic interference.
End delimiter	A bit sequence used to signal the end of a frame.
Ethernet	A widely accepted and recognized industry standard local area network.
Event	An occurrence on a device that causes a Fieldbus entity to send the Fieldbus event message.
Event service	Event services allow the user application to report events and manage event processing.

F

FAS	*See* Fieldbus Access Sublayer.
FB	Function Block.
FBAP	*See* Function Block Application.
FF	Foundation Fieldbus.
Field device	A Fieldbus device connected directly to a Fieldbus.
Field equipment	Sensors, actuators, and controllers.
Fieldbus	An all-digital, two-way communication system that connects control systems to instrumentation. A process control local area network defined by ISA standard S50.02.
Fieldbus Access Sublayer	The layer of the communication stack that provides an interface between the DLL and layer seven of the OSI model. The FAS provides communication services such as client/server, publisher/subscriber, and event distribution.
Fieldbus cable	Shielded, twisted pair cable made specifically for Fieldbus that has characteristics important for good signal transmission and are within the requirements of the Fieldbus standard.
Fieldbus Foundation	An organization that developed a Fieldbus network specifically based upon the work and principles of the ISA/IEC standards committees.
Fieldbus Messaging Specification	The layer of the communication stack that defines a model for applications to interact over the Fieldbus. The services FMS provides allow you to read and write information about the OD, read and write the data variables described in the OD, and perform other activities such as uploading/downloading data and invoking programs inside a device.
Fieldbus Network Address	Location of a board or device on the Fieldbus; the Fieldbus node address.
FIP	Factory Instrumentation Protocol.
FMS	*See* Fieldbus Messaging Specification.
Foundation Fieldbus specification	The communications network specification that the Fieldbus Foundation created.
Frame	A single transmission from a device.

FTP	File Transfer Protocol.
Function block	A named block consisting of one or more input, output, and contained parameters. The block performs some control function as its algorithm. Function blocks are the core components you control a system with. The Fieldbus Foundation defines standard sets of function blocks. There are ten function blocks for the most basic control and I/O functions. Manufacturers can define their own function blocks.
Function Block Application	The block diagram that represents your control strategy.
Function Block Application Editor window	The middle window of the NI-FBUS Configurator where you create your block diagram.
Function block execution schedule	A list of times in the macrocycle when the function block will begin to execute its algorithm.

G

Gateway	Fieldbus device that connects networks that use different communication protocols, such as ControlNet-to-H1.
Ground	An intentional or accidental conducting path between an electrical system or circuit and the earth or some conducting body acting in place of the earth. A ground is often used as the common wiring point or reference in a circuit.

H

H1	The 31.25 kbit/second type of Fieldbus.
H1 device	An H1 device is a Fieldbus device connected directly to an H1 Fieldbus. Typical H1 devices are valves and transmitters.
H2 Fieldbus	Foundation Fieldbus High Speed Ethernet (HSE) backbone network running at 100 Mbit/second.
Hard code	To permanently establish something that should be variable in a program.
HART	Highway Addressable Remote Transducer. A digital communication protocol which operates on top of a conventional 4-20 mA current loop signal from a measurement device, such as a transmitter, or to a valve positioner.
Header file	A C-language source file containing important definitions and function prototypes.

hex	Hexadecimal. A base-16 numbering system which uses 0-9 and A-F.
HMI	Human-Machine Interface. A graphical user interface for the process with supervisory control and data acquisition capability.
Homerun	A twisted-pair cable that connects the control room equipment with a number of devices in the field.
Host device	A computer or controller on a Fieldbus network.
HotPnP	Hot Plug and Play.
HSE	High Speed Ethernet.
Hz	Hertz.

I

I/O	Input/output.
IEC	International Electrotechnical Commission. A technical standards committee which is at the same level as the ISO.
Impedance	Resistance.
in.	Inches.
Incremental DD	Device description provided by device suppliers that references the standard device descriptions. Incremental device descriptions may include supplier-specific features such as calibration and diagnostic procedures for their devices.
Index	An integer that the Fieldbus specification assigns to a Fieldbus object or a device that you can use to refer to the object. A value in the object dictionary used to refer to a single object.
Inductor	Length of conductor used to introduce inductance into a circuit. The conductor is usually wound into a coil to concentrate the magnetic lines of force and maximize the inductance. While any conductor has inductance, in common usage the term inductor usually refers to a coil.
Industrial network	Standardized digital communications network used in industrial automation applications; they often replace vendor-proprietary networks so that devices from different vendors can communicate in control systems.
Input parameter	A block parameter that receives data from another block.

Intrinsic safety	A characteristic of wiring or devices that cannot cause atmospheres to ignite or explode.
IRQ	Interrupt request.
IS	*See* Intrinsic safety.
IS barrier	A device used to keep voltages and currents on wires below the levels that can ignite an atmosphere.
ISA	Industry Standard Architecture.
ISO	International Organization for Standardization. A technical standards organization that creates international technical standards for computers and communications. The ISO is composed of national standards organizations in 89 countries. The American National Standards Institute (ANSI) represents the United States in the ISO.
Isolation	A type of signal conditioning in which you isolate the transducer signals from the computer for safety purposes. This protects you and your computer from large voltage spikes and makes sure the measurements from the devices are not affected by differences in ground potentials.

K

Kb	Kilobytes.
Kbits	Kilobits.
Kernel	The set of programs in an operating system that implements basic system functions.
Kernel mode	The mode in which device drivers run on Windows.
Keycode	Code required from National Instruments to add links to a Fieldbus network when using the NI-FBUS Configurator.
KHz	Kilohertz.
km	Kilometer.

L

LabVIEW	Laboratory Virtual Instrument Engineering Workbench—a program development application based on the programming language G and used commonly for test and measurement purposes.
LabVIEW DSC	The LabVIEW Datalogging and Supervisory Control (DSC) Module builds on the power of LabVIEW for high channel count and distributed applications. It adds easy networking, channel and I/O management, alarm and event management, historical datalogging, real-time trending, and OPC integration to the LabVIEW environment.
LAN	Local Area Network.
LAS	*See* Link Active Scheduler.
LED	Light-emitting diode.
Lift-off voltage	The initial voltage required for a Fieldbus device to start operating.
Link	A Foundation Fieldbus network is made up of devices connected by a serial bus. This serial bus is called a link (also known as a segment).
Link Active Schedule	A schedule of times in the macrocycle when devices must publish their output values on the Fieldbus.
Link Active Scheduler	The Fieldbus device that is currently controlling access to the Fieldbus. A device that is responsible for keeping a link operational. The LAS executes the link schedule, circulates tokens, distributes time, and probes for new devices.
Link context	Describes a category of NI-FBUS calls that accept link descriptors.
Link ID	*See* Link identifier.
Link identifier	A number that specifies a link.
Link master device	A device that is capable of becoming the LAS.
Link object	Link objects define the links between function block inputs and outputs internal to the device and across the Fieldbus network.
Linkage	A connection between function blocks.

Linkage object	An object resident in a device that defines connections between function block input and output across the network. Linkage objects also specify trending connections.
Live list	The list of all devices that are properly responding to the Pass Token.
LM	Link Master.
Local Area Network	A communications network that is limited in physical spatial area for the purpose of easier connection of computers in neighboring buildings.
Lookout	National Instruments Lookout is a full-featured object-based automation software system that delivers unparalleled power and ease of use in demanding industrial measurement and automation applications.
Loop	*See* Control loop.

M

m	Meters.
M/E	Mechanical-to-electric transducer.
mA	milliampere.
Macrocycle	The least common multiple of all the loop times on a given link, or one iteration of a process control loop.
Manchester	A coding method used for sending digital data on the Fieldbus.
Manufacturer's identification	An identifier used to correlate the device type and revision with its device description and device description revision.
MAU	Medium Attachment Unit.
Mbit	Megabit.
Menu	An area accessible from the command bar that displays a subset of the possible command choices. In the NI-FBUS Configurator, refers to menus defined by the manufacturer for a given block.
Method	Methods describe operating procedures to guide a user through a sequence of actions.
ML	Manual Loader.
mm	millimeter.

Mode	Type of communication.
Multicast	To send a message to multiple destinations.
Multi-drop	A transmission circuit with multiple terminals and peripherals. Could also be described as branches off a bus.
Multiplexer	A switching device with multiple inputs that sequentially connects each of its inputs to its outputs, typically at high speeds, in order to measure several signals with a single analog input channel.

N

Network address	The Fieldbus network address of a device.
Network Management	A layer of the Foundation Fieldbus communication stack that contains objects that other layers of the communication stack use, such as Data Link, FAS, and FMS. You can read and write SM and NM objects over the Fieldbus using FMS Read and FMS Write services.
Nifb.exe	The NIFB process that must be running in the background for you to use your PCI-FBUS or PCMCIA-FBUS interface to communicate between the board and the Fieldbus.
NI-FBUS API	The NI-FBUS Communications Manager.
NI-FBUS Communications Manager	Software shipped with National Instruments Fieldbus interfaces that lets you read and write values. It does not include configuration capabilities.
NI-FBUS Configurator	National Instruments Fieldbus configuration software. With it, you can set device addresses, clear devices, change modes, and read and write to the devices.
NI-FBUS Fieldbus Configuration System	*See* NI-FBUS Configurator.
NI-FBUS process	Process that must be running in the background for you to use your PCI-FBUS or PCMCIA-FBUS interface to communicate between the board and the Fieldbus.
NM	*See* Network Management.
NMIB	Network Management Information Base.
Node	Junction or branch point in a circuit.

Non-scheduled/acyclic communication	Communication that occurs at times that are not predetermined.
Non-volatile memory	Memory that does not require electricity to hold data.

O

Object	An element of an object dictionary.
Object attribute	A part of the machine-readable description of a Fieldbus object.
Object description	Describes data that is communicated over the Fieldbus.
Object Dictionary	A structure in a device that describes data that can be communicated on the Fieldbus. The object dictionary is a lookup table that gives information such as data type and units about a value that can be read from or written to a device.
Object value	The actual data value associated with a Fieldbus object.
Octet	A single 8-bit value.
OD	*See* Object Dictionary.
Offline	Not connected to or installed in the computer.
OOS	Out of Service mode.
OPC	OLE for Process Control.
Operator acknowledgment alarm	The verification an operator performs when he or she receives a Fieldbus message.
OSI	Open Systems Interconnect.
OSI Model	Open Systems Interconnect Layered Communication Model. A communications protocol standard that the ISO created. It establishes a seven-layered framework for implementing protocols. In the OSI model, control moves from one layer to the next in the following manner: control starts at the top layer in one station, moves through all protocol layers to the bottom layer, then goes over the channel to the next station and moves back up through all protocol layers.
Output parameter	A block parameter that sends data to another block.

P

Parameter	One of a set of network-visible values that makes up a function block.
PC	Personal Computer.
PCMCIA	Personal Computer Memory Card International Association.
PD	Proportional Derivative.
PDU	Protocol Data Unit.
Peak-to-peak voltage	A measure of signal amplitude; the difference between the highest and lowest excursions of the signal.
Physical device	A single device residing at a unique address on the Fieldbus.
Physical device context	Describes a category of NI-FBUS functions that accept physical device descriptors.
Physical device tag	A user-defined name for a physical device.
Physical Layer	The layer of the communication stack that converts digital Fieldbus messages from the communication stack to actual physical signals on the Fieldbus transmission medium and vice versa.
PI	Program Invocation.
PID	Proportional/Integral/Derivative. A common control function block algorithm that uses proportions, integrals, and derivatives in calculation.
PLC	*See* Programmable Logic Controller.
PN	Probe Node.
Polarity	Term used to describe positive and negative charges.
Poll	To repeatedly inspect a variable or function block to acquire data.
Port	A communications connection on a computer or remote controller.
POST	Power-On Self Test.
Power conditioner	Used to connect a conventional power source to a Fieldbus link.
PR	Probe Response.

Preamble	A bit sequence used to synchronize a signal receiver.
Process value	A common Fieldbus function block parameter representing some value in the process being controlled.
Program	A set of instructions the computer can follow, usually in a binary file format, such as a `.exe` file.
Programmable Logic Controller	A device with multiple inputs and outputs that contains a program you can alter.
PT	Pass Token.
Publisher	A device that has at least one function block with its output value connected to the input of another device.
Publishing schedule	Determines when data members are published over the Fieldbus.
PV	Process Value.

R

RA	Ratio.
Rack-mounted controller	Controller mounted on a shelving unit in a chassis.
RAM	Random-Access Memory.
RC	Ratio Control.
Reflection	An unwanted signal that results from a cable fault or improper termination.
Repeater	Boost the signals to and from the further link.
Resistance	Opposition to current flow and dissipation of energy in the form of heat.
Resistor	Component made of material that opposes flow of current and therefore has some value of resistance.
Resource block	A special block containing parameters that describe the operation of the device and general characteristics of a device, such as manufacturer and device name. Only one resource block per device is allowed.
Roundcard	A hardware interface for developing Foundation Fieldbus-compliant devices.

S

s	Seconds.
Sample type	Specifies how trends are sampled on a device, whether by averaging data or by instantaneous sampling.
Scheduled/cyclic communications	Communication that occurs at the same time during each control cycle.
Segment	*See* Link.
Sensor	A device that responds to a physical stimulus (heat, light, sound, pressure, motion, flow, and so on), and produces a corresponding electrical signal.
Server	Device that receives a message request.
Service	Services allow user applications to send messages to each other across the Fieldbus using a standard set of message formats.
Session	A communication path between an application and the NI-FBUS Communications Manager.
Session context	Describes a category of NI-FBUS functions that accept session descriptors.
Session level	A category of NI-FBUS API calls that accepts a session descriptor.
Shield	Metal grounded cover used to protect a wire, component or piece of equipment from stray magnetic and/or electric fields.
Signal	An extension of the IEEE 488.2 standard that defines a standard programming command set and syntax for device-specific operations.
SMIB	System Management Information Base.
Spur	A secondary route having a junction to the primary route in a network.
Stack	A set of hardware registers or a reserved amount of memory used for calculations or to keep track of internal operations.
Stale	Data that has not been updated for `stale_limit` number of macrocycles, where the stale limit is a parameter of the connection.

Start delimiter	A bit sequence used to signal the start of the data portion of a frame.
Start FMS	A service used to change the state from `Idle` to `Running`, and so on.
Static library	A library of functions/subroutines that you must link to your application as one of the final steps of compilation, as opposed to a Dynamic Link Library, which links to your application at run time.
Stub	*See* Spur.
Subscribed parameter	A block parameter that is receiving data.
Subscriber	A device that has at least one function block with its input value connected to the output of another device.
Surge	Large, unwanted voltage or current on wires. Generally caused by lightning or nearby heavy electrical power use.
Surge suppressor	A device used to discharge surges to ground.
Symbol file	A Fieldbus Foundation or device manufacturer-supplied file that contains the ASCII names for all the objects in a device.
System Management	A layer of the Foundation Fieldbus communication stack that assigns addresses and physical device tags, maintains the function block schedule for the function blocks in that device, and distributes application time. You can also locate a device or a function block tag through SM.
System Management Configuration	Configuration parameters that set up device identification and network time distribution.

T

Tag	A name you can define for a block, virtual field device, or device.
TB	Transducer Block.
TCP/IP	Transmission Control Protocol/Internet Protocol. The communications protocol used on the Internet.
TD	Time Distribution.
Terminator	A device used to absorb the signal at the end of a wire.

Thread	An operating system object that consists of a flow of control within a process. In some operating systems, a single process can have multiple threads, each of which can access the same data space within the process. However, each thread has its own stack and all threads can execute concurrently with one another (either on multiple processors, or by time-sharing a single processor).
Timeout	A period of time after which an error condition is raised if some event has not occurred.
Token passing	The LAS sends a Pass Token (PT) message to all devices in the live list. The device is allowed to transmit unscheduled messages when it receives the PT.
Tokenizer	A program the Fieldbus Foundation provides that creates a binary form of DDL code to ship to an end user with an instrument.
Traditional system	*See* 4-20 mA system.
Transducer block	A block that is an interface to the physical, sensing hardware in the device. It also performs the digitizing, filtering, and scaling conversions needed to present input data to function blocks, and converts output data from function blocks. Transducer blocks decouple the function blocks from the hardware details of a given device, allowing generic indication of function block input and output. Manufacturers can define their own transducer blocks.
Trend	A Fieldbus object that allows a device to sample a process value periodically, then transmit a history of the values on the network.
Trend function	An NI-FBUS call related to trends.
Trunk	*See* Homerun.

U

UCT	Universal Coordinated Time.
Unipolar	A signal range that is always positive (for example, 0 to +10 V).
Unscheduled	Messages sent on the Fieldbus between transmissions of scheduled messages.

Upstream	Fewer network hops away from a backbone or hub. For example, a small ISP that connects to the Internet through a larger ISP that has their own connection to the backbone is downstream from the larger ISP, and the larger ISP is upstream from the smaller ISP.
User Layer	The network layer of the communication stack above layer seven in the OSI. The User Layer defines blocks and objects that represent the functions and data available in a device.

V

V	Volts.
Variable list	A list of variables you can access with a single Fieldbus transaction.
VCR	*See* Virtual Communication Relationship.
VDC	Volts Direct Current.
VFD	*See* Virtual Field Device.
VFD context	Describes a category of NI-FBUS functions that accept virtual field device descriptors.
View objects	Predefined groupings of parameter sets that HMI applications use.
Virtual Communication Relationship	Preconfigured or negotiated connections between virtual field devices on a network.
Virtual Field Device	The virtual field device is a model for remotely viewing data described in the object dictionary. The services provided by the Fieldbus Messaging Specification allow you to read and write information about the object dictionary, read and write the data variables described in the object dictionary, and perform other activities such as uploading/downloading data and invoking programs inside a device. A model for remotely viewing data described in the object dictionary.

W

Waveform	Multiple voltage readings taken at a specific sampling rate.

Index

H

I

K

L

M

N

O

P

R

S

T

U

V

W